PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

PROGRAMME REGULATIONS & CURRICULUM

2025-29

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BACHELOR OF TECHNOLOGY (B.TECH.) IN COMPUTER SCIENCE AND ENGINEERING

www.presidencyuniversity.in #proudpresidencian

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Program Regulations and Curriculum 2025-2029

BACHELOR OF TECHNOLOGY (B.Tech.) in COMPUTER SCIENCE AND ENGINEERING

based on Choice Based Credit System (CBCS) and Outcome Based Education (OBE)

Table of Contents

PART A – PROGRAM REGULATIONS

1. **Vision & Mission of the University and the School / Department**

1.1 Vision of the University

To be a Value-driven Global University, excelling beyond peers and creating professionals of integrity and character, having concern and care for society.

1.2 Mission of the University

- Commit to be an innovative and inclusive institution by seeking excellence in teaching, research and knowledge-transfer.
- Pursue Research and Development and its dissemination to the community, at large.
- Create, sustain and apply learning in an interdisciplinary environment with consideration for ethical, ecological and economic aspects of nation building.
- Provide knowledge-based technological support and services to the industry in its growth and development.
- To impart globally-applicable skill-sets to students through flexible course offerings and support industry's requirement and inculcate a spirit of new-venture creation.

1.3 Vision of Presidency School of Computer Science and Engineering

To be a value based, practice-driven School of Computer Science and Engineering, committed to developing globally-competent Engineers, dedicated to transforming Society.

1.4 Mission of Presidency School of Computer Science and Engineering

- Cultivate a practice-driven environment with a contemporary Learning-pedagogy, integrating theory and practice.
- Attract and nurture world-class faculty to excel in Teaching and Research, in the field of Core Computer Science and Engineering.
- Establish state-of-the-art facilities for effective Teaching and Learning-experiences.
- Promote Interdisciplinary Studies to nurture talent and impart relevant skill-sets for global impact.
- Instil Entrepreneurial and Leadership Skills to address Social, Environmental, and Community-needs.

2. **Preamble to the Program Regulations and Curriculum**

This is the subset of Academic Regulations and it is to be followed as a requirement for the award of B.Tech degree.

The Curriculum is designed to take into the factors listed in the Choice Based Credit System (CBCS) with focus on Industrial Based Project Learning, Industrial Training, and Internship to enable the students to become eligible and fully equipped for employment in industries, choose higher studies or entrepreneurship.

In exercise of the powers conferred by and in discharge of duties assigned under the relevant provision(s) of the Act, Statutes and Academic Regulations, 2025 of the University, the Academic Council hereby makes the following Regulations.

3. **Short Title and Applicability**

a. These Regulations shall be called the Bachelor of Technology Degree Program Regulations and Curriculum 2025-2029.
b. These Regulations are subject to, and pursuant to the Academic Regulations.
c. These Regulations shall be applicable to the ongoing Bachelor of Technology Degree Programs of the 2025-2029 batch, and to all other Bachelor of Technology Degree Programs which may be introduced in future.

d. These Regulations shall supersede all the earlier Bachelor of Technology Degree Program Regulations and Curriculum, along with all the amendments thereto.
e. These Regulations shall come into force from the Academic Year 2025-2026.

4. Definitions

In these Regulations, unless the context otherwise requires:

a. *"Academic Calendar" means the schedule of academic and miscellaneous events as approved by the Vice Chancellor;*
b. *"Academic Council" means the Academic Council of the University;*
c. *"Academic Regulations" means the Academic Regulations, of the University;*
d. *"Academic Term" means a Semester or Summer Term;*
e. *"Act" means the Presidency University Act, 2013;*
f. *"AICTE" means All India Council for Technical Education;*
g. *"Basket" means a group of courses bundled together based on the nature/type of the course;*
h. *"BOE" means the Board of Examinations of the University;*
i. *"BOG" means the Board of Governors of the University;*
j. *"BOM" means the Board of Management of the University;*
k. *"BOS" means the Board of Studies of a particular Department/Program of Study of the University;*
l. *"CGPA" means Cumulative Grade Point Average as defined in the Academic Regulations;*
m. *"Clause" means the duly numbered Clause, with Sub-Clauses included, if any, of these Regulations;*
n. *"COE" means the Controller of Examinations of the University;*
o. *"Course In Charge" means the teacher/faculty member responsible for developing and organising the delivery of the Course;*
p. *"Course Instructor" means the teacher/faculty member responsible for teaching and evaluation of a Course;*
q. *"Course" means a specific subject usually identified by its Course-code and Course-title, with specified credits and syllabus/course-description, a set of references, taught by some teacher(s)/course-instructor(s) to a specific class (group of students) during a specific Academic Term;*
r. *"Curriculum Structure" means the Curriculum governing a specific Degree Program offered by the University, and, includes the set of Baskets of Courses along with minimum credit requirements to be earned under each basket for a degree/degree with specialization/minor/honours in addition to the relevant details of the Courses and Course catalogues (which describes the Course content and other important information about the Course). Any specific requirements for a particular program may be brought into the Curriculum structure of the specific program and relevant approvals should be taken from the BOS and Academic Council at that time.*
s. *"DAC" means the Departmental Academic Committee of a concerned Department/Program of Study of the University;*
t. *"Dean" means the Dean / Director of the concerned School;*
u. *"Degree Program" includes all Degree Programs;*
v. *"Department" means the Department offering the degree Program(s) / Course(s) / School offering the concerned Degree Programs / other Administrative Offices;*
w. *"Discipline" means specialization or branch of B.Tech. Degree Program;*
x. *"HOD" means the Head of the concerned Department;*
y. *"L-T-P-C" means Lecture-Tutorial-Practical-Credit – refers to the teaching – learning periods and the credit associated;*
z. *"MOOC" means Massive Open Online Courses;*

aa. "MOU" means the Memorandum of Understanding;

bb. "NPTEL" means National Program on Technology Enhanced Learning;

cc. "Parent Department" means the department that offers the Degree Program that a student undergoes;

dd. "Program Head" means the administrative head of a particular Degree Program/s;

ee. "Program Regulations" means the Bachelor of Technology Degree Program Regulations and Curriculum, 2024-2028;

ff. "Program" means the Bachelor of Technology (B.Tech.) Degree Program;

gg. "PSCS" means the Presidency School of Computer Science;

hh. "Registrar" means the Registrar of the University;

ii. "School" means a constituent institution of the University established for monitoring, supervising and guiding, teaching, training and research activities in broadly related fields of studies;

jj. "Section" means the duly numbered Section, with Clauses included in that Section, of these Regulations;

kk. "SGPA" means the Semester Grade Point Average as defined in the Academic Regulations, 2021;

ll. "Statutes" means the Statutes of Presidency University;

mm. "Sub-Clause" means the duly numbered Sub-Clause of these Program Regulations;

nn. "Summer Term" means an additional Academic Term conducted during the summer break (typically in June-July) for a duration of about eight (08) calendar weeks, with a minimum of thirty (30) University teaching days;

oo. "SWAYAM" means Study Webs of Active Learning for Young Aspiring Minds.

pp. "UGC" means University Grant Commission;

qq. "University" means Presidency University, Bengaluru; and

rr. "Vice Chancellor" means the Vice Chancellor of the University.

5. Program Description

The Bachelor of Technology Degree Program Regulations and Curriculum 2023-2027 are subject to, and, pursuant to the Academic Regulations. These Program Regulations shall be applicable to the following ongoing Bachelor of Technology (B.Tech.) Degree Programs of 2023-2027 offered by the Presidency School of Computer Science and Engineering (PSCS):

1. Bachelor of Technology in Computer Science and Engineering, abbreviated as B.Tech. Computer Science and Engineering;
2. Bachelor of Technology in Computer Science and Technology (Big Data), abbreviated as B.Tech. Computer Science and Technology (Big Data);
3. Bachelor of Technology in Computer Science and Engineering (Block Chain), abbreviated as B.Tech. Computer Science and Engineering (Block Chain);
4. Bachelor of Technology in Computer Science and Technology (Dev Ops), abbreviated as B.Tech. Computer Science and Technology (Dev Ops);
5. Bachelor of Technology in Computer Science and Engineering (Cyber Security), abbreviated as B.Tech. Computer Science and Engineering (Cyber Security);
6. Bachelor of Technology in Computer Science and Engineering (Internet of Things), abbreviated as B.Tech. Computer Science and Engineering (Internet of Things);
7. Bachelor of Technology in Computer Science and Engineering (Data Science), abbreviated as B.Tech. Computer Science and Engineering (Data Science);
8. Bachelor of Technology in Computer Science and Technology (Artificial Intelligence and Machine Learning), abbreviated as B.Tech. Computer Science and Technology (Artificial Intelligence and Machine Learning);
9. Bachelor of Technology in Information Science and Technology, abbreviated as B.Tech. Information Science and Technology;

10. Bachelor of Technology in Computer Science and Information Technology, abbreviated as B.Tech. Computer Science and Information Technology;
11. Bachelor of Technology in Computer Science and Engineering (Networks), abbreviated as B.Tech. Computer Science and Engineering (Networks);
12. Bachelor of Technology in Computer Engineering (Artificial Intelligence and Machine Learning), abbreviated as B.Tech. Computer Engineering (Artificial Intelligence and Machine Learning);
13. Bachelor of Technology in Information Science and Engineering (Artificial Intelligence and Robotics), abbreviated as B.Tech. Information Science and Engineering (Artificial Intelligence and Robotics); and
14. Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) abbreviated as B.Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning);

5.1. Bachelor of Technology in Computer Science and Engineering, abbreviated as B.Tech. (Computer Science and Engineering)

5.2 These Program Regulations shall be applicable to other similar programs, which may be introduced in future.

5.3 These Regulations may evolve and get amended or modified or changed through appropriate approvals from the Academic Council, from time to time, and shall be binding on all concerned.

6. Minimum and Maximum Duration

6.1 Bachelor of Technology Degree Program is a Four-Year, Full-Time Semester based program. The minimum duration of the B.Tech. Program is four (04) years and each year comprises of two academic Semesters (Odd and Even Semesters) and hence the duration of the B.Tech. program is eight (08) Semesters.

6.2 A student who for whatever reason is not able to complete the Program within the normal period or the minimum duration (number of years) prescribed for the Program, may be allowed a period of two years beyond the normal period to complete the mandatory minimum credits requirement as prescribed by the concerned Program Regulations and Curriculum. In general, the permissible maximum duration (number of years) for completion of Program is 'N' + 2 years, where 'N' stands for the normal or minimum duration (number of years) for completion of the concerned Program as prescribed by the concerned Program Regulations and Curriculum.

6.3 The time taken by the student to improve Grades/CGPA, and in case of temporary withdrawal/re-joining (Refer to Clause 16.1 in Academic Regulations), shall be counted in the permissible maximum duration for completion of a Program.

6.4 In exceptional circumstances, such as temporary withdrawal for medical exigencies where there is a prolonged hospitalization and/or treatment, as certified through hospital/medical records, women students requiring extended maternity break (certified by registered medical practitioner), and, outstanding sportspersons representing the University/State/India requiring extended time to participate in National/International sports events, a further extension of one (01) year may be granted on the approval of the Academic Council.

6.5 The enrolment of the student who fails to complete the mandatory requirements for the award of the concerned Degree (refer Section 19.0 of Academic Regulations) in the prescribed maximum duration (Sub-Clauses 18.1 and 18.2 of Academic Regulations), shall stand terminated and no Degree shall be awarded.

7. Programme Educational Objectives (PEO)

After four years of successful completion of the program, the graduates shall be able to:

PEO1. Demonstrate as a Computer Engineering Professional with innovative skills and moral and ethical values.

PEO2. Become a Teaching and Research Professional in the area of Computer science and engineering through lifelong learning.

PEO3. Emerge as a Consultancy team member in the Computer Science and Engineering Industry.

PEO4. Evolve as an entrepreneur in the computer science and other related areas of specialization.

8. Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

8.1. Programme Outcomes (PO)

On successful completion of the Program, the students shall be able to:

PO1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem Analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. **Life-Long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

8.2. Program Specific Outcomes (PSOs):

On successful completion of the Program, the students shall be able to:

PSO1: Problem Analysis: Identify, formulate, research literature, and analyze complex engineering problems related to Software Engineering principles and practices, Programming and Computing technologies reaching substantiated conclusions using first principle

PSO2: Design/development of Solutions: Design solutions for complex engineering problems related to Software Engineering principles and practices, Programming and Computing technologies and design system components or processes that meet the specified needs

PSO3: Modern Tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities related to Software Engineering principles and practices, Programme.

9. Admission Criteria (as per the concerned Statutory Body)

The University admissions shall be open to all persons irrespective of caste, class, creed, gender or nation. All admissions shall be made on the basis of merit in the qualifying examinations; provided that forty percent of the admissions in all Programs of the University shall be reserved for the students of Karnataka State and admissions shall be made through a Common Entrance Examination conducted by the State Government or its agency and seats shall be allotted as per the merit and reservation policy of the State Government from time to time**.** The admission criteria to the B.Tech. Program is listed in the following Sub-Clauses:

8.1 An applicant who has successfully completed Pre-University course or Senior Secondary School course (+2) or equivalent such as (11+1), 'A' level in Senior School Leaving Certificate Course from a recognized university of India or outside or from Senior Secondary Board or equivalent, constituted or recognized by the Union or by the State Government of that Country for the purpose of issue of qualifying certificate on successful completion of the course, may apply for and be admitted into the Program.

8.2 Provided further, the applicant must have taken Physics and Mathematics as compulsory subjects in the Pre-University / Higher Secondary / (10+2) / (11+1) examination, along with either Chemistry / Biology / Electronics / Computer Science / Biotechnology subject, and, the applicant must have obtained a minimum of 45% of the total marks (40% in case of candidates belonging to the Reserved Category as classified by the Government of Karnataka) in these subjects taken together.

8.3 The applicant must have appeared for Joint Entrance Examinations (JEE) Main / JEE (Advanced) / Karnataka CET / COMED-K, or any other State-level Engineering Entrance Examinations.

8.4 Reservation for the SC / ST and other backward classes shall be made in accordance with the directives issued by the Government of Karnataka from time to time.

8.5 Admissions are offered to Foreign Nationals and Indians living abroad in accordance with the rules applicable for such admission, issued from time to time, by the Government of India.

8.6 Candidates must fulfil the medical standards required for admission as prescribed by the University.

8.7 If, at any time after admission, it is found that a candidate had not in fact fulfilled all the requirements stipulated in the offer of admission, in any form whatsoever, including possible misinformation and any other falsification, the Registrar shall report the matter to the Board of Management (BOM), recommending revoking the admission of the candidate.

8.8 The decision of the BOM regarding the admissions is final and binding.

10. Lateral Entry / Transfer Students requirements

10.1. Lateral Entry

The University admits students directly to the second year (3rd Semester) of the B.Tech. Degree program as per the provisions and/or regulations of the Government of Karnataka pertaining to the “Lateral Entry” scheme announced by the Government from time to time. Further, the general conditions and rules governing the provision of Lateral Entry to the B.Tech. Program of the University are listed in the following Sub-Clauses:

10.1.1. Admission to 2nd year (3rd Semester) of the B.Tech. Degree program shall be open to the candidates who are holders of a 3-year Diploma in Engineering (or equivalent qualification as recognized by the University), who have secured not less than forty-five percentage (45%) marks in the final year examination (5^{th} and 6^{th} Semesters of the Diploma Program) in the appropriate branch of Engineering. Provided that, in case of SC / ST and OBC candidates from Karnataka the minimum marks for eligibility shall be forty percent (40%).

10.1.2. Provided further that, candidates seeking Lateral Entry may be required to complete specified bridge Courses as prescribed by the University. Such bridge Courses, if any, shall not be included in the CGPA computations.

10.1.3. All the existing Regulations and Policies of the University shall be binding on all the students admitted to the Program through the provision of Lateral Entry.

10.1.4. The Course requirements prescribed for the 1^{st} Year of the B.Tech. Program shall be waived for the student(s) admitted through Lateral Entry and the duration of the B.Tech. Program for such students is three (03) years, commencing from the 3^{rd} Semester (commencement of the 2^{nd} Year) of the B.Tech. Program and culminating with the 8^{th} Semester (end of the 4^{th} Year) of the B.Tech. Program.

10.1.5. Provided that, if a Lateral Entry student misses any mandatory program specific courses that are typically offered in the 1^{st} year (1^{st} or 2^{nd} semesters), then those courses must be cleared by the students as soon as possible, preferably during the Summer Term.

10.1.6. The existing Program Regulations of the concerned Program to which the student is admitted through the provision of Lateral Entry shall be binding on the student with effect from the 3^{rd} Semester of the Program. i.e., the Program Structure and Curriculum from the 3^{rd} to 8^{th} Semesters of the Program concerned shall be binding on the student admitted through Lateral Entry. Further, any revisions / amendments made to the Program Regulations thereafter, shall be binding on all the students of the concerned Program.

10.1.7. All the Courses (and the corresponding number of Credits) prescribed for the 1^{st} Year of the concerned B.Tech. Program shall be waived for the student(s) admitted to the concerned B.Tech Program through Lateral Entry. Further, the *Minimum Credit Requirements* for the award of the B.Tech. Degree in the concerned Program shall be prescribed / calculated as follows:

The ***Minimum Credit Requirements*** for the award of the Bachelor of Technology (B.Tech.) Degree prescribed by the concerned Bachelor of Technology Degree Program Regulations and Curriculum, 2025-2029, minus the number of Credits prescribed / accepted by the Equivalence Committee for the 1^{st} Year (1^{st} and 2^{nd} Semesters) of the B.Tech. Program.

For instance, if the *Minimum Credit Requirements* for the award of the Bachelor of Technology (B.Tech.) Degree as prescribed by the Regulations for B.Tech. (Computer Science and Engineering) is "N" Credits, and, if the total credits prescribed in the 1^{st} Year (total credits of the 1^{st} and 2^{nd} Semesters) of the Program concerned is "M" Credits, then the *Minimum Credit Requirements* for the award of the B.Tech. in Computer Science and Engineering for a student who joins the Program through the provision of the Lateral Entry, shall be "N – M" Credits.

10.1.8. Further, no other waiver except the Courses prescribed for the 1st year of the B.Tech. Program of the University shall be permissible for students joining the B.Tech. Program through the provision of Lateral Entry.

10.2. Transfer of student(s) from another recognized University to the 2nd year (3rd Semester) of the B.Tech. Program of the University

A student who has completed the 1^{st} Year (i.e., passed in all the Courses / Subjects prescribed for the 1^{st} Year) of the

B.Tech./B.E./B.S., Four-Year Degree Program from another recognized University, may be permitted to transfer to the 2nd Year (3rd Semester) of the B.Tech. Program of the Presidency University as per the rules and guidelines prescribed in the following Sub-Clauses:

10.2.1. The concerned student fulfils the criteria specified in Sub-Clauses 10.1.1, 10.1.2 and 10.1.3.

10.2.2. The student shall submit the Application for Transfer along with a non-refundable Application Fee (as prescribed by the University from time to time) to the Presidency University no later than July 10 of the concerned year for admission to the 2nd Year (3rd Semester) B.Tech. Program commencing on August 1 on the year concerned.

10.2.3. The student shall submit copies of the respective Marks Cards / Grade Sheets / Certificates along with the Application for Transfer.

10.2.4. The transfer may be provided on the condition that the Courses and Credits completed by the concerned student in the 1st Year of the B.Tech./ B.E. / B.S. Four Degree Program from the concerned University, are declared equivalent and acceptable by the Equivalence Committee constituted by the Vice Chancellor for this purpose. Further, the Equivalence Committee may also prescribe the Courses and Credits the concerned students shall have to mandatorily complete, if admitted to the 2nd Year of the B.Tech. Program of the University.

10.2.5. The Branch / Discipline allotted to the student concerned shall be the decision of the University and binding on the student.

11. Change of Branch / Discipline / Specialization

A student admitted to a particular Branch of the B.Tech. Program will normally continue studying in that Branch till the completion of the program. However, the University reserves the right to provide the option for a change of Branch, or not to provide the option for a change of Branch, at the end of 1st Year of the B.Tech. Program to eligible students in accordance with the following rules and guidelines: framed by the University from time to time.

11.1. Normally, only those students, who have passed all the Courses prescribed for the 1st Year of the B.Tech. Program and obtained a CGPA of not less than 6.50 at the end of the 2nd Semester, shall be eligible for consideration for a change of Branch.

11.2. Change of Branch, if provided, shall be made effective from the commencement of the 3rd Semester of the B.Tech. Program. There shall be no provision for change of Branch thereafter under any circumstances whatsoever.

11.3. The student provided with the change of Branch shall fully adhere to and comply with the Program Regulations of the concerned Branch of the B.Tech. Program, the Fee Policy pertaining to that Branch of the B.Tech. Program, and, all other rules pertaining to the changed Branch existing at the time.

11.4. Change of Branch once made shall be final and binding on the student. No student shall be permitted, under any circumstances, to refuse the change of Branch offered.

11.5. The eligible student may be allowed a change in Branch, strictly in order of inter se merit, subject to the conditions given below:

11.5.1 The actual number of students in the 3rd Semester in any particular Branch to which the transfer is to be made, should not exceed the intake fixed by the University for the concerned Branch;

11.5.2 The actual number of students in any Branch from which transfer is being sought does not fall below 75% of the total intake fixed by the University for the concerned Branch.

The process of change of Branch shall be completed within the first five days of Registration for the 3rd Semester of the B.Tech. Program.

12. Specific Regulations regarding Assessment and Evaluation (including the Assessment Details of NTCC Courses, Weightages of Continuous Assessment and End Term Examination for various Course Categories)

12.1 The academic performance evaluation of a student in a Course shall be according to the University Letter Grading System based on the class performance distribution in the Course.

12.2 Academic performance evaluation of every registered student in every Course registered by the student is carried out through various components of Assessments spread across the Semester. The nature of components of Continuous Assessments and the weightage given to each component of Continuous Assessments (refer Clause 8.8 of Academic regulations) shall be clearly defined in the Course Plan for every Course, and approved by the DAC.

12.3 Format of the End-Term examination shall be specified in the Course Plan.

12.4 Grading is the process of rewarding the students for their overall performance in each Course. The University follows the system of Relative Grading with statistical approach to classify the students based on the relative performance of the students registered in the concerned Course except in the following cases:

- Non-Teaching Credit Courses (NTCC)
- Courses with a class strength less than 30

Absolute grading method may be adopted, where necessary with prior approval of concerned DAC.

Grading shall be done at the end of the Academic Term by considering the aggregate performance of the student in all components of Assessments prescribed for the Course. Letter Grades (Clause 8.10 of Academic regulations) shall be awarded to a student based on her/his overall performance relative to the class performance distribution in the concerned Course. These Letter Grades not only indicate a qualitative assessment of the student's performance but also carry a quantitative (numeric) equivalent called the Grade Point

12.5 **Assessment Components and Weightage**

	Credit Structure [L-T-P-C]	Percentage / Marks	CA		Mid-Term		End-Tterm		Project	Total	Exam Conducted by
			Theory	Practical	Theory	Practical	Theory	Practical			
1	3-0-0-3	Percentage	25%	-	25%	-	50%	-	-	100 %	Mid-Term & End Term by CoE
		Marks	50	-	50	-	100	-	-	200	
2	2-0-2-3	Percentage	12.50 %	12.50%	12.50 %	12.50 %	25%	25%	-	100 %	Mid-Term & End Term by CoE * Except for full stack courses
		Marks	25	25	25	25	50	50	-	200	
3	1-0-4-3	Percentage	-	25%	10%	40%	5%	20%	-	100 %	Mid-Term & End Term by School
		Marks	-	25	10	40	5	20	-	100	
4	2-0-4-4	Percentage	12.50 %	12.50%	10%	15%	20%	30%	-	100 %	*Mid-Term & End Term by CoE
		Marks	25	25	20	30	40	60	-	200	
5	0-0-4-2	Percentage	-	50%	-	-	-	-	50 %	100 %	Project evaluated by IC at School level
		Marks	-	50	-	-	-	-	50	100	
6	0-0-2-1	Percentage	-	100%	-	-	-	-	-	100 %	Only CA at School Level
		Marks	-	100	-	-	-	-	-	100	
7	3-0-2-4	Percentage	12.50 %	12.50%	15%	10%	30%	20%	-	100 %	Mid-Term & End Term by CoE
		Marks	25	25	30	20	60	40	-	200	
8	2-0-0-2	Percentage	25%	-	25%	-	50%	-	-	100 %	Mid-Term & End Term by CoE
		Marks	50	-	50	-	100	-	-	200	

* CSE3150-Front End Full stack development

* CSE3151-Java Full Stack Development

* CSE3152-.Net Full Stack development

The exact weightages of Evaluation Components shall be clearly specified in the concerned PRC and respective Course Plan.

Normally, for Practice/Skill based Courses, without a defined credit structure (L–T–P) [NTCC], but with assigned Credits (as defined in Clause 5.2 of the Academic Regulations), the method of evaluation shall be based only on Continuous Assessments. The various components of Continuous Assessments, the distribution of weightage among such components, and the method of evaluation/assessment, shall be as decided and indicated in the Course Plan/PRC. The same shall be approved by the respective DAC.

12.6 **Minimum Performance Criteria:**

12.6.1 Theory only Course and Lab/Practice Embedded Theory Course

12.6.1.1 A student shall satisfy the following minimum performance criteria to be eligible to earn the credits towards the concerned Course:

12.6.1.2 A student must obtain a minimum of 30% of the total marks/weightage assigned to the End Term Examinations in the concerned Course.

12.6.1.3 The student must obtain a minimum of 40% of the AGGREGATE of the marks/weightage of the components of Continuous Assessments, Mid Term Examinations and End Term Examinations in the concerned Course.

12.6.2 Lab/Practice only Course and Project Based Courses

The student must obtain a minimum of 40% of the AGGREGATE of the marks/weightage of all assessment components in the concerned Course.

12.6.3 A student who fails to meet the minimum performance criteria listed above in a Course shall be declared as "Fail" and given "F" Grade in the concerned Course. For theory Courses, the student shall have to re-appear in the "Make-Up Examinations" as scheduled by the University in any subsequent semester, or, re-appear in the End Term Examinations of the same Course when it is scheduled at the end of the following Semester or Summer Term, if offered. The marks obtained in the Continuous Assessments (other than the End Term Examination) shall be carried forward and be included in computing the final grade, if the student secures the minimum requirements (as per Clause 8.9.1, 8.9.2 of Academic regulations) in the "Make-Up Examinations" of the concerned Course. Further, the student has an option to re-register for the Course and clear the same in the summer term/ subsequent semester if he/she wishes to do so, provided the Course is offered.

13 Additional clarifications - Rules and Guidelines for Transfer of Credits from MOOC, etc. – Note: These are covered in Academic Regulations

The University allows students to acquire credits from other Indian or foreign institutions and/or Massive Open Online Course (MOOC) platforms, subject to prior approval. These credits may be transferred and counted toward fulfilling the minimum credit requirements for the award of a degree. The process of transfer of credits is governed by the following rules and guidelines:

13.1 The transfer of credits shall be examined and recommended by the Equivalence Committee (Refer Annexure B of Academic regulations) and approved by the Dean - Academics.

13.2 Students may earn credits from other Indian or foreign Universities/Institutions with which the University has an MOU, and that MOU shall have specific provisions, rules and guidelines for transfer of credits. These transferred credits shall be counted towards the minimum credit requirements for the award of the degree.

13.3 Students may earn credits by registering for Online Courses offered by Study Web of Active Learning by Young and Aspiring Minds (SWAYAM) and National Program on Technology Enhanced Learning (NPTEL), or other such recognized Bodies/ Universities/Institutions as approved by the concerned BOS and Academic Council from time to time. The concerned School/Parent Department shall publish/include the approved list of Courses and the rules and guidelines governing such transfer of credits of the concerned Program from time to time. The Rules and Guidelines for the transfer of credits specifically from the Online Courses conducted by SWAYAM/ NPTEL/ other approved MOOCs are as stated in the following Sub-Clauses:

13.3.1 A student may complete SWAYAM/NPTEL/other approved MOOCs as mentioned in Clause 17.3 (as per Academic regulations) and transfer equivalent credits to partially or fully complete the mandatory credit requirements of Discipline Elective Courses and/or the mandatory credit requirements of Open Elective Courses as prescribed in the concerned Curriculum Structure. However, it is the sole responsibility of the student to complete the mandatory credit requirements of the Discipline Elective Courses and the Open Elective Courses as prescribed by the Curriculum Structure of the concerned Program.

13.3.2 SWAYAM/NPTEL/ other approved MOOCs as mentioned in Clause 17.3 (as per Academic regulations) shall be approved by the concerned Board of Studies and placed (as Annexures) in the concerned PRC.

13.3.3 Parent Departments may release a list of SWAYAM/NPTEL/other approved MOOCs for Pre-Registration as per schedule in the Academic Calendar or through University Notification to this effect.

13.3.4 Students may Pre-Register for the SWAYAM/NPTEL/other approved MOOCs in the respective Departments and register for the same Courses as per the schedule announced by respective Online Course Offering body/institute/ university.

13.3.5 A student shall request for transfer of credits only from such approved Courses as mentioned in Sub Clause 17.3.2 above.

13.3.6 SWAYAM/NPTEL/other approved MOOCs Courses are considered for transfer of credits only if the concerned student has successfully completed the SWAYAM/NPTEL/other approved MOOCs and obtained a certificate of successful/satisfactory completion.

13.3.7 A student who has successfully completed the approved SWAYAM/NPTEL/ other approved MOOCs and wants to avail the provision of transfer of equivalent credits, must submit the original Certificate of Completion, or such similar authorized documents to the HOD concerned, with a written request for the transfer of the equivalent credits. On verification of the Certificates/Documents and approval by the HOD concerned, the Course(s) and equivalent Credits shall forwarded to the COE for processing of results of the concerned Academic Term.

13.3.8 The credit equivalence of the SWAYAM/NPTEL/other approved MOOCs are based on Course durations and/or as recommended by the Course offering body/institute/university. The Credit Equivalence mapped to SWAYAM/ NPTEL approved Courses based on Course durations for transfer of credits is summarised in Table shown below. The Grade will be calculated from the marks received by the Absolute Grading Table 8.11 in the Academic regulations.

Table 2: Durations and Credit Equivalence for Transfer of Credits from SWAYAM-NPTEL/ other approved MOOC Courses

Sl. No.	Course Duration	Credit Equivalence
1	4 Weeks	1 Credit
2	8 Weeks	2 Credits
3	12 Weeks	3 Credits

13.6.1 The maximum permissible number of credits that a student may request for credit transfer from MOOCs shall not exceed 20% of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum for the award of the concerned Degree.

13.3.9 The University shall not reimburse any fees/expense; a student may incur for the SWAYAM/NPTEL/other approved MOOCs.

13.4 The maximum number of credits that can be transferred by a student shall be limited to forty percent (40%) of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum for the award of the concerned Degree. However, the grades obtained in the Courses transferred from other Institutions/MOOCs, as mentioned in this Section (13.0), shall not be included in the calculation of the CGPA.

13.5 **Mandatory Non-Credit Course Completion Requirements:** All mandatory non-credit courses shall be satisfactorily completed by the student as part of the degree requirements. These courses will be evaluated and awarded letter grades based on the following criteria:

S (Satisfactorily Completed): Awarded when the student successfully completes all prescribed course requirements.

NC (Not Completed): Awarded when the student fails to meet the prescribed course requirements.

A student receiving an **NC** grade must reappear for and complete the course in accordance with the guidelines prescribed by the University.

In the case of non-taught and non-credited mandatory courses—where students are advised to undertake learning through MOOC platforms—there shall be a clearly defined Course Catalogue and a corresponding Course Plan. The Course Plan shall outline the assessment components, which will form the basis for evaluation.

PART B – PROGRAM STRUCTURE

14. Structure / Component with Credit Requirements Course Baskets & Minimum Basket wise Credit Requirements

The B.Tech. (Computer Science and Engineering) Program Structure (2025-2029) totaling 160 credits. Table 3.0 summarizes the type of baskets, number of courses under each basket and the associated credits that are mandatorily required for the completion of the Degree.

Table 3.0: B.Tech. (Computer Science & Engineering) 2025-2029: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets

Sl. No.	Baskets	Credit Contribution
1	Humanities and Social Sciences including Management Courses (HSMC)	10
2	Basic Science Courses (BSC)	24
3	Engineering Science Courses (ESC)	22
4	Professional Core Courses (PCC)	64
5	Professional Elective Courses (PEC)	18
6	Project Work (PRW)	16
7	Open Elective Courses (OEC)	6
8	Mandatory Courses (MAC)*	0
	Total Credits	**160 (Minimum)**

* Please refer to Table 3.8, (where the number '8' corresponds to the serial number of the Mandatory course basket.)

In the entire Program, the practical and skill-based course component contribute to an extent of approximately 61% out of the total credits of 160 for B.Tech. (Computer Science and Engineering) program of four years' duration.

15. Minimum Total Credit Requirements of Award of Degree

As per the AICTE guidelines, a minimum of 160 credits is required for the award of a B.Tech. degree.

16. Other Specific Requirements for Award of Degree, if any, as prescribed by the Statutory Bodies,

16.1 The award of the Degree shall be recommended by the Board of Examinations and approved by the Academic Council and Board of Management of the University.

16.2 A student shall be declared to be eligible for the award of the concerned Degree if she/he:

a. Fulfilled the Minimum Credit Requirements and the Minimum Credits requirements under various baskets;

b. Secure a minimum CGPA of 4.50 in the concerned Program at the end of the Semester/Academic Term in which she/he completes all the requirements for the award of the Degree as specified in Sub-Clause 19.2.1 of Academic Regulations;

c. No dues to the University, Departments, Hostels, Library, and any other such Centers/ Departments of the University; and

d. No disciplinary action is pending against her/him.

PART C – CURRICULUM STRUCTURE

17. Curriculum Structure – Basket Wise Course List (not Semester Wise)
List of Courses Tabled – aligned to the Program Structure
(Course Code, Course Name, Credit Structure (LTPC), Contact Hours, Course Basket, Type of Skills etc., as applicable).

Type of Skill
F - Foundation S - Skill Development EM – Employability EN – Entrepreneurship

Course Caters to
GS - Gender Sensitization ES - Environment and sustainability HP - Human values and Professional Ethics

Baskets
HSMC - Humanities and Social Sciences (*including Management courses*) BSC - Basic Science Courses ESC - Engineering Science Courses
PCC - Program Core Course
PEC - Professional Elective Courses
OEC - Open Elective Courses
EEC - Employment Enhancement Courses
MAC - Mandatory Course

Table 3.1 :
List of Humanities and Social Sciences including Management Courses (HSMC)

Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Pre-requisite
1	ENG1900	English for Technical Communication	2	0	0	2	2	S	Nil
2	DES1146	Introduction to Design Thinking	1	0	0	1	1	F	Nil
3	ENG2501	Advanced English	2	0	0	2	2	S	Nil
4	FIN1002	Essentials of Finance	3	0	0	3	3	S	Nil
5	APT4005	Aptitude for Employability	0	0	2	1	2	AT	Nil
6	PPS3018	Preparedness for Interview	0	0	2	1	2	AT	Nil
		Total	**8**	**0**	**4**	**10**	**12**		

Table 3.2 : List of Basic Science Courses (BSC)

Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Pre-requisite
1	MAT2301	Calculus and Differential Equations	3	1	0	4	4	F	Nil
2	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F	Nil
3	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F	Nil
4	MAT2402	Probability and Statistics	3	1	0	4	4	F	Nil
5	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S	Nil
6	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S	Nil
7	MAT2303	Linear Algebra and Vector Calculus	3	1	0	4	4	EM	Nil
8	MAT2404	Discrete Mathematics	3	1	0	4	4	F	Nil
		Total	**18**	**4**	**4**	**24**	**26**		

Table 3.3 : List of Engineering Science Courses (ESC)									
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	MEC1006	Engineering Graphics	2	0	0	2	2	S	Nil
2	CSE1500	Computational Thinking using Python	2	0	2	3	4	S	Nil
3	ECE2022	Digital Design	2	0	0	2	2	F/S	Nil
4	ECE2052	Digital Design Lab	0	0	2	1	2	F/S	Nil
5	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S	Nil
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S	Nil
7	EEE1250	Basics of Electrical and Electronics Engineering Lab	0	0	2	1	2	F/S	Nil
8	ECE1511	Design Workshop	1	0	2	2	3	S/EM	Nil
9	CSE2264	Essentials of AI	3	0	0	3	3	S/EM	Nil
10	CSE2265	Essentials of AI Lab	0	0	2	1	2	S/EM	Nil
11	CSE2274	Competitive Programming and Problem Solving	0	0	4	2	4	S/EM	Nil
		Total	**15**	**0**	**14**	**22**	**29**		

Table 3.4 : List of Professional Core Courses (PCC)									
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	CSE2200	Problem Solving using C	2	0	0	2	2	S	Nil
2	CSE2201	Problem Solving using C Lab	0	0	4	2	4	S	Nil
3	CSE2251	Data Communication and Computer Networks	3	0	0	3	3	S	Nil
4	CSE2252	Data Communication and Computer Networks Lab	0	0	2	1	2	S	Nil
5	CSE2253	Data Structures	3	0	0	3	3	S	Nil
6	CSE2254	Data Structures Lab	0	0	2	1	2	S	Nil
7	CSE2255	Object Oriented Programming Using Java	3	0	0	3	3	S/EM	Nil
8	CSE2256	Object Oriented Programming Using Java Lab	0	0	2	1	2	S/EM	Nil
9	CSE2257	Computer Organization and Architecture	3	0	0	3	3	S	Nil
10	CSE2258	Web Technologies	3	0	0	3	3	S/EM	Nil
11	CSE2259	Web Technologies Lab	0	0	2	1	2	S/EM	Nil
12	CSE2260	Database Management Systems	3	0	0	3	3	S	Nil
13	CSE2261	Database Management Systems Lab	0	0	2	1	2	S	Nil
14	CSE2500	Data Analytics	3	0	0	3	2	S/EM	MAT2402
15	CSE2501	Data Analytics Lab	0	0	2	1	2	S/EM	MAT2402
16	CSE2262	Analysis of Algorithms	3	1	0	4	4	S	Nil
17	CSE2263	Analysis of Algorithms Lab	0	0	2	1	2	S	Nil
18	CSE2266	Theory of Computation	3	0	0	3	3	S	Nil
19	CSE2502	Cryptography and Network Security	3	0	0	3	3	S	CSE2251
20	CSE2267	Machine Learning Techniques	3	0	0	3	3	S/EM	Nil

21	CSE2268	Machine Learning Techniques Lab	0	0	2	1	2	S/EM	Nil
22	CSE2269	Operating Systems	3	0	0	3	3	S	Nil
23	CSE2270	Operating Systems Lab	0	0	2	1	2	S/EM	Nil
24	CSE2503	Scalable Application Development using Java	3	0	0	3	3	S/EM	CSE2255
25	CSE2504	Scalable Application Development using Java Lab	0	0	2	1	2	S/EM	CSE2256
26	CSE2271	Software Design and Development	3	0	0	3	3	S	Nil
27	CSE2272	Cloud Computing	2	0	0	2	2	S/EM	Nil
28	CSE2273	Cloud Computing Lab	0	0	2	1	2	S/EM	Nil
29	CSE2505	Mobile Application Development	2	0	0	2	2	S/EM	CSE2255
30	CSE2506	Mobile Application Development Lab	0	0	4	2	4	S/EM	CSE2256
		Total	**48**	**1**	**30**	**64**	**78**		

18. List of Elective Courses under various Specializations / Stream Basket

Table 3.5 : Professional Electives Courses/Specialization Tracks – Minimum of 12 credits is to be earned by the student in a particular track and overall 18 credits.

Track -1 Artificial Intelligence and Machine Learning

Sl. No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Prerequisite	Type of course
1	CSE3500	Intelligent Systems with Machine Learning	2	0	2	3	3	S	CSE2267	PEC
2	CSE3501	Advanced Deep Learning Techniques	2	0	2	3	3	S/ EM	CSE2267	PEC
3	CSE3502	Computational Optimization for Intelligent Systems.	2	0	2	3	3	S	CSE2267	PEC
4	CSE3503	Reinforcement Learning for AI Systems	2	0	2	3	3	S/EM	CSE2267	PEC
5	CSE3504	Computational Linguistics Natural Language Processing	2	0	2	3	3	S	CSE2267	PEC
6	CSE3505	Synergistic Neural Fuzzy Computing	2	0	2	3	3	S	CSE2267	PEC

Track -2 BioInformatics & Data Science

Sl. No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Prerequisite	Type of course
1	CSE3506	Introduction to Bioinformatics	2	0	2	3	3	S/ EM	Nil	PEC
2	CSE3507	Algorithms in Computational Biology	2	0	2	3	3	S/EM	CSE3506	PEC
3	CSE3508	Statistical Methods for Bioinformatics	2	0	2	3	3	S/EM	CSE3506	PEC
4	CSE3509	Emerging Technologies in Big Data	2	0	2	3	3	S	CSE2500	PEC
5	CSE3510	Statistical Techniques for Data Science	2	0	2	3	3	S	MAT2402	PEC
6	CSE3511	Predictive Analytics and Applications	2	0	2	3	3	S/EM	MAT2402	PEC
7	CSE3512	Data Mining	2	0	2	3	3	S	MAT2402	PEC
8	CSE3513	No SQL Data Management	2	0	2	3	3	S	CSE2500	PEC
9	CSE3514	Applied Data Intelligence	2	0	2	3	3	S/EM	CSE2264	PEC

Track 3 - Cloud & Security										
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Prerequisite**	**Type of course**
1	CSE3515	Cloud Data Engineering	2	0	2	3	3	S	CSE2272	PEC
2	CSE3516	Federated Learning	2	0	2	3	3	S	CSE2272	PEC
3	CSE3517	Edge Computing	2	0	2	3	3	S/EM	CSE2272	PEC
4	CSE3518	Network Security and Firewall Management	2	0	2	3	3	S/EM	CSE2502	PEC
5	CSE3519	Information Security and Management	2	0	2	3	3	S	CSE2502	PEC
6	CSE3520	Network Intrusion Detection and Prevention	2	0	2	3	3	S	CSE2502	PEC
7	CSE3521	Principles and Practices of Web Security	2	0	2	3	3	S/EM	CSE2502	PEC
8	CSE3522	Penetration Testing and Risk Assessment	2	0	2	3	3	S/EM	CSE2502	PEC
Track 4 - Fintech and Blockchain										
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Prerequisite**	**Type of course**
1	CSE3523	Introduction to Fintech	2	0	2	3	3	S/EM	FIN1001	PEC
2	CSE3524	Banking Technology	2	0	2	3	3	S/EM	NIl	PEC
3	CSE3525	Blockchain Technology	2	0	2	3	3	S/EM	Nil	PEC
4	CSE3526	Embedded and Decentralized Finance	2	0	2	3	3	S/EM	FIN1001	PEC
5	CSE3527	Financial and Capital Markets	2	0	2	3	3	S	FIN1001	PEC
6	CSE3528	Blockchain Development and Programming	2	0	2	3	3	S	FIN1001	PEC
7	CSE3529	Statistics and Data Analysis for Finance	2	0	2	3	3	S	FIN1001	PEC
8	CSE3530	Financial Regulations and Compliances	2	0	2	3	3	S	FIN1001	PEC
Track -5 Programming										
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Prerequisite**	**Type of course**
1	CSE3531	Go Programming	2	0	2	3	3	S/ EM	CSE2200	PEC
2	CSE3532	Advanced Database Management Systems	2	0	2	3	3	S	CSE2210	PEC
3	CSE3533	Programming in C# and .NET	2	0	2	3	3	S	CSE2210	PEC
4	CSE3534	Rust Programming	2	0	2	3	3	S/ EM	CSE2200	PEC
Track - 6 Special Track										
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Prerequisite**	**Type of course**
1	CAI3427	Language Models for Text Mining	2	0	2	3	3	S/ EM	CSE2264	PEC

2	CAI3428	Practical Deep Learning with TensorFlow	2	0	2	3	3	S/ EM	CSE2264	PEC
3	CAI3429	Deep Learning Techniques for Computer Vision	2	0	2	3	3	S/ EM	MAT2402	PEC
4	CSE3426	Front End Full Stack Development *	2	0	2	3	3	S/ EM	CSE2258	PEC
5	CSE3427	Java Full Stack Development *	2	0	2	3	3	S/ EM	CSE2258	PEC
6	CSE3428	.Net Full Stack Development *	2	0	2	3	3	S/ EM	CSE2258	PEC
Track -5 Mandatory Non-Credited Course (Offered for Lateral Entry students in higher semester wherever applicable in MOOC mode)**										
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Credit Effort**	**Type of Skill**	**Prerequisite**	
1	LAW7601	Indian Constitution **	0	0	0	0		F	Nil	MAC
2	CHE7601	Environmental Studies **	0	0	0	0		F	Nil	MAC
3	CIV7601	Universal Human Values and Ethics **	0	0	0	0		F	Nil	MAC
*Mandatory for Students selected for Tech Mahindra and Capgemini										
** Offered for Lateral Entry students in higher semester wherever applicable in MOOC mode										
+Mandatory for Students Selected for Samsung Innovation Campus										

19. Practical / Skill based Courses – Internships / Thesis / Dissertation / Capstone Project Work / Portfolio / Mini project

Table 3.6 : List of course in Project Work basket (PRW)									
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	CSE7000	Internship	0	0	0	2	0	S/EM	--
2	CSE7100	Mini Project	0	0	0	4	0	S/EM	--
3	CSE7300	Capstone Project	0	0	0	10	0	S/EM	--
		Total	**0**	**0**	**0**	**16**	**0**		

Practical / Skill based Courses like internship, project work, capstone project, research project / dissertation, and such similar courses, where the pedagogy does not lend itself to a typical L-T-P-C Structure as defined in Clause 5.1 of the Academic Regulations, are simply assigned the number of Credits based on the quantum of work / effort required to fulfill the learning objectives and outcomes prescribed for the concerned Courses. Such courses are referred to as Non-Teaching Credit Courses (NTCC). These Courses are designed to provide students with hands-on experience and skills essential for their professional development. These courses aim to equip students with abilities in problem identification, root cause analysis, problem-solving, innovation, and design thinking through industry exposure and project-based learning. The expected outcomes are first level proficiency in problem solving and design thinking skills to better equip B.Tech. graduates for their professional careers. The method of evaluation and grading for the Practical / Skill based Courses shall be prescribed and approved by the concerned Departmental Academic Committee (refer Annexure A of the Academic Regulations). The same shall be prescribed in the Course Handout.

18.1 Internship

A student may undergo an internship for a period of 6-8 weeks in an industry / company or academic / research institution during the Semester Break between 4th and 5th Semesters or 6th and 7th Semesters, subject to the following conditions:

18.1.1 The Internship shall be in conducted in accordance with the Internship Policy prescribed by the University from time to time.

18.1.2 The selection criteria (minimum CGPA, pass in all Courses as on date, and any other qualifying criteria) as applicable / stipulated by the concerned Industry / Company or academic / research institution for award of the Internship to a student;

18.1.3 The number of Internships available for the concerned Academic Term. Further, the available number of internships shall be awarded to the students by the University on the basis of merit using the CGPA secured by the student. Provided further, the student fulfils the criteria, as applicable, specified by the Industry / Company or academic / research institution providing the Internship, as stated in Sub-Clause 18.1.2 above.

18.1.4 A student may opt for Internship in an Industry / Company or academic / research institution of her / his choice, subject to the condition that the concerned student takes the responsibility to arrange the Internship on her / his own. Provided further, that the Industry / Company or academic / research institution offering such Internship confirms to the University that the Internship shall be conducted in accordance with the Program Regulations and Internship Policy of the University.

18.1.5 A student selected for an Internship in an industry / company or academic / research institution shall adhere to all the rules and guidelines prescribed in the Internship Policy of the University.

18.2 Project Work

A student may opt to do a Project Work for a period of 6-8 weeks in an Industry / Company or academic / research institution or the University Department(s) as an equivalence of Internship during the Semester Break between 4^{th} and 5^{th} Semesters or 6^{th} and 7^{th} Semesters or during the 5^{th} / 6^{th} / 7^{th} Semester as applicable, subject to the following conditions:

18.2.1 The Project Work shall be approved by the concerned HOD and be carried out under the guidance of a faculty member.

18.2.2 The student may do the project work in an Industry / Company or academic / research institution of her / his choice subject to the above-mentioned condition (Sub-Clause 2.6.2.1). Provided further, that the Industry / Company or academic / research institution offering such project work confirms to the University that the project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.3 Capstone Project

A student may undergo a Capstone Project for a period of 10-12 weeks in an industry / company or academic / research institution in the 7th / 8th Semester as applicable, subject to the following conditions:

18.3.1 The Capstone Project shall be in conducted in accordance with the Capstone Project Policy prescribed by the University from time to time.

18.3.2 The selection criteria (minimum CGPA, pass in all Courses as on date, and any other qualifying criteria) as applicable / stipulated by the concerned Industry / Company or academic / research institution for award of the Capstone Project to a student;

18.3.3 The number of Capstone Project available for the concerned Academic Term. Further, the available number of Capstone Project shall be awarded to the students by the University on the basis of merit using the CGPA secured by the student. Provided further, the student fulfils the criteria, as applicable, specified by the Industry / Company or academic / research institution providing the Capstone Project, as stated in Sub-Clause 18.3.2 above.

18.3.4 A student may opt for Capstone Project in an Industry / Company or academic / research institution of her / his choice, subject to the condition that the concerned student takes the responsibility to arrange the I Capstone Project on her / his own. Provided further, that the Industry / Company or academic / research institution offering such Capstone Project confirms to the University that the Capstone Project shall be conducted in accordance with the Program Regulations and Capstone project Policy of the University.

18.3.5 A student selected for a Capstone Project in an industry / company or academic / research institution shall adhere to all the rules and guidelines prescribed in the Capstone Project Policy of the University.

18.4 Research Project / Dissertation

A student may opt to do a Research Project / Dissertation for a period of 12-14 weeks in an Industry / Company or academic / research institution or the University Department(s) as an equivalence of Capstone Project, subject to the following conditions:

18.4.1 The Research Project / Dissertation shall be approved by the concerned HOD and be carried out under the guidance of a faculty member.

The student may do the Research Project / Dissertation in an Industry / Company or academic / research institution of her / his choice subject to the above mentioned condition (Sub-Clause 18.4.1). Provided further, that the Industry / Company or academic / research institution offering such Research Project / Dissertation confirms to the University that the Research Project / Dissertation work will be conducted in accordance with the Program Regulations and requirements of the University.

20. List of Open Electives to be offered by the School / Department (Separately for ODD and EVEN Semesters.

Type of Skill
F - Foundation
S - Skill Development
EM – Employability
EN – Entrepreneurship

Course Caters to
GS - Gender Sensitization
ES - Environment and sustainability
HP - Human values and Professional Ethics

Table 3.7: Open Elective Courses Baskets: Minimum Credits to be earned from this Basket is 06											
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Type of Skill/ Focus**	**Course Caters to**	**Pre-requisites/ Co-requisites**	**Anti-requisites**	**Future Courses that need this as a Prerequisite**
Chemistry Basket											
1	CHE1003	Fundamentals of Sensors	3	0	0	3	S	ES	-	-	-
2	CHE1004	Smart materials for IOT	3	0	0	3	S	ES	-	-	-
3	CHE1005	Computational Chemistry	2	0	0	2	S	ES	-	-	-
4	CHE1006	Introduction to Nano technology	3	0	0	3	S	ES	-	-	-
5	CHE1007	Biodegradable electronics	2	0	0	2	S	ES	-	-	-
6	CHE1008	Energy and Sustainability	2	0	0	2	S	ES	-	-	-
7	CHE1009	3D printing with Polymers	2	0	0	2	S	ES	-	-	-
8	CHE1010	Bioinformatics and Healthcare IT	2	0	0	2	S	ES	-	-	-
9	CHE1011	Chemical and Petrochemical catalysts	3	0	0	3	S	ES	-	-	-
10	CHE1012	Introduction to Composite materials	2	0	0	2	S	ES	-	-	-
11	CHE1013	Chemistry for Engineers	3	0	0	3	S	ES	-	-	-
12	CHE1014	Surface and Coatings technology	3	0	0	3	S	ES	-	-	-
13	CHE1015	Waste to Fuels	2	0	0	2	S	ES	-	-	-
14	CHE1016	Forensic Science	3	0	0	3	S	ES	-	-	-
Civil Engineering Basket											
1	CIV1001	Disaster mitigation and management	3	0	0	3	S	-	-	-	-
2	CIV1002	Environment Science and Disaster Management	3	0	0	3	FC	-	-	-	-
3	CIV2001	Sustainability Concepts in Engineering	3	0	0	3	S	-	-	-	-
4	CIV2002	Occupational Health and Safety	3	0	0	3	S	-	-	-	-
5	CIV2003	Sustainable Materials and Green Buildings	3	0	0	3	EM	-	-	-	-
6	CIV2004	Integrated Project Management	3	0	0	3	EN	-	-	-	-
7	CIV2005	Environmental Impact Assessment	3	0	0	3	EN	-	-	-	-
8	CIV2006	Infrastructure Systems for Smart Cities	3	0	0	3	EN	-	-	-	-
9	CIV2044	Geospatial Applications for Engineers	2	0	2	3	EM	-	-	-	-
10	CIV2045	Environmental Meteorology	3	0	0	3	S	-	-	-	-
11	CIV3046	Project Problem Based Learning	3	0	0	3	S	-	-	-	-

12	CIV3059	Sustainability for Professional Practice	3	0	0	3	EN	-	-	-	-
Commerce Basket												
1	COM2001	Introduction to Human Resource Management	2	0	0	2	F	HP/GS	-	-	-
2	COM2002	Finance for Non Finance	2	0	0	2	S	-	-	-	-
3	COM2003	Contemporary Management	2	0	0	2	F	-	-	-	-
4	COM2004	Introduction to Banking	2	0	0	2	F	-	-	-	-
5	COM2005	Introduction to Insurance	2	0	0	2	F	-	-	-	-
6	COM2006	Fundamentals of Management	2	0	0	2	F	-	-	-	-
7	COM2007	Basics of Accounting	3	0	0	3	F	-	-	-	-
Computer Science Basket (not to be offered for Computer Science and Engineering students)												
1	CSE2002	Programming in Java	2	0	2	3	S/EM	-	-	-	-
2	CSE2003	Social Network Analytics	3	0	0	3	S	GS	-	-	-
3	CSE2004	Python Application Programming	2	0	2	3	S/ EM	-	-	-	-
4	CSE2005	Web design fundamentals	2	0	2	3	S/ EM/EN	-	-	-	-
Design Basket												
1	DES1001	Sketching and Painting	0	0	2	1	S	-	-	-	-
2	DES1002	Innovation and Creativity	2	0	0	2	F	-	-	-	-
3	DES1121	Introduction to UX design	1	0	2	2	S	-	-	-	-
4	DES1122	Introduction to Jewellery Making	1	0	2	2	S	-	-	-	-
5	DES1124	Spatial Stories	1	0	2	2	S	-	-	-	-
6	DES1125	Polymer Clay	1	0	2	2	S	-	-	-	-
7	DES2001	Design Thinking	3	0	0	3	S	-	-	-	-
8	DES1003	Servicability of Fashion Products	1	0	2	2	F	ES	-	-	-
9	DES1004	Choices in Virtual Fashion	1	0	2	2	F	ES, GS, HP	-	-	-
10	DES1005	Fashion Lifestyle and Product Diversity	1	0	2	2	F	ES, GS, HP	-	-	-
11	DES1006	Colour in Everyday Life	1	0	2	2	F	ES	-	-	-
12	DES2080	Art of Design Language	3	0	0	3	S	-	-	-	-
13	DES2081	Brand Building in Design	3	0	0	3	S	-	-	-	-
14	DES2085	Web Design Techniques	3	0	0	3	S	-	-	-	-
15	DES2089	3D Modeling for Professionals	1	0	4	3	S	-	-	-	-
16	DES2090	Creative Thinking for Professionals	3	0	0	3	S	-	-	-	-
17	DES2091	Idea Formulation	3	0	0	3	S	-	-	-	-
Electrical and Electronics Basket												
1	EEE1002	IoT based Smart Building Technology	3	0	0	3	S	-	-	-	-
2	EEE1003	Basic Circuit Analysis	3	0	0	3	S	-	-	-	-
3	EEE1004	Fundamentals of Industrial Automation	3	0	0	3	S	-	-	-	-
4	EEE1005	Electric Vehicles & Battery Technology	3	0	0	3	S	-	-	-	-
5	EEE1006	Smart Sensors for Engineering Applications	3	0	0	3	S	-	-	-	-
Electronics and Communication Basket												
1	ECE1003	Fundamentals of Electronics	3	0	0	3	F	-	-	-	-
2	ECE1004	Microprocessor based systems	3	0	0	3	F	-	-	-	-
3	ECE3089	Artificial Neural Networks	3	0	0	3	S	-	-	-	-
4	ECE3097	Smart Electronics in Agriculture	3	0	0	3	F/EM	-	-	-	-
5	ECE3098	Environment Monitoring Systems	3	0	0	3	F/EM	-	-	-	-
6	ECE3102	Consumer Electronics	3	0	0	3	F/EM	-	-	-	-
7	ECE3103	Product Design of Electronic Equipment	3	0	0	3	S/F/ EM / EN	-	-	-	-
8	ECE3106	Introduction to Data Analytics	3	0	0	3	F/EM	-	-	-	-
9	ECE3107	Machine Vision for Robotics	3	0	0	3	F/EM	-	-	-	-

English Basket											
1	ENG1008	Indian Literature	2	0	0	2	-	GS/ HP	-	-	-
2	ENG1009	Reading Advertisement	3	0	0	3	S	-	-	-	-
3	ENG1010	Verbal Aptitude for Placement	2	0	2	3	S	-	-	-	-
4	ENG1011	English for Career Development	3	0	0	3	S	-	-	-	-
5	ENG1012	Gender and Society in India	2	0	0	2	-	GS/ HP	-	-	-
6	ENG1013	Indian English Drama	3	0	0	3	-	-	-	-	-
7	ENG1014	Logic and Art of Negotiation	2	0	2	3	-	-	-	-	-
8	ENG1015	Professional Communication Skills for Engineers	1	0	0	1	-	-	-	-	-
DSA Basket											
1	DSA2001	Spirituality for Health	2	0	0	2	F	HP	-	-	-
2	DSA2002	Yoga for Health	2	0	0	2	S	HP	-	-	-
3	DSA2003	Stress Management and Well Being	2	0	0	2	F	-	-	-	-
Kannada Basket											
1	KAN1001	Kali Kannada	1	0	0	1	S	-	-	-	-
2	KAN1003	Kannada Kaipidi	3	0	0	3	S	-	-	-	-
3	KAN2001	Thili Kannada	1	0	0	1	S	-	-	-	-
4	KAN2003	Pradharshana Kale	1	0	2	2	S	-	-	-	-
5	KAN2004	Sahithya Vimarshe	2	0	0	2	S	-	-	-	-
6	KAN2005	Anuvadha Kala Sahithya	3	0	0	3	S	-	-	-	-
7	KAN2006	Vichara Manthana	3	0	0	3	S	-	-	-	-
8	KAN2007	Katha Sahithya Sampada	3	0	0	3	S	-	-	-	-
9	KAN2008	Ranga Pradarshana Kala	3	0	0	3	S	-	-	-	-
Foreign Language Basket											
1	FRL1004	Introduction of French Language	2	0	0	2	S	S	-	-	-
2	FRL1005	Fundamentals of French	2	0	0	2	S	S	-	-	-
3	FRL1009	Mandarin Chinese for Beginners	3	0	0	3	S	S	-	-	-
Law Basket											
1	LAW1001	Introduction to Sociology	2	0	0	2	F	HP		-	-
2	LAW2001	Indian Heritage and Culture	2	0	0	2	F	HP/GS		-	-
3	LAW2002	Introdcution to Law of Succession	2	0	0	2	F	HP/GS		-	-
4	LAW2003	Introduction to Company Law	2	0	0	2	F	HP		-	-
5	LAW2004	Introduction to Contracts	2	0	0	2	F	HP	-	-	-
6	LAW2005	Introduction to Copy Rights Law	2	0	0	2	F	HP	-	-	-
7	LAW2006	Introduction to Criminal Law	2	0	0	2	F	HP	-	-	-
8	LAW2007	Introduction to Insurance Law	2	0	0	2	F	HP	-	-	-
9	LAW2008	Introduction to Labour Law	2	0	0	2	F	HP	-	-	-
10	LAW2009	Introduction to Law of Marriages	2	0	0	2	F	HP/GS	-	-	-
11	LAW2010	Introduction to Patent Law	2	0	0	2	F	HP	-	-	-
12	LAW2011	Introduction to Personal Income Tax	2	0	0	2	F	HP	-	-	-
13	LAW2012	Introduction to Real Estate Law	2	0	0	2	F	HP	-	-	-
14	LAW2013	Introduction to Trademark Law	2	0	0	2	F	HP	-	-	-
15	LAW2014	Introduction to Competition Law	3	0	0	3	F	HP	-	-	-
16	LAW2015	Cyber Law	3	0	0	3	F	HP	-	-	-
17	LAW2016	Law on Sexual Harrassment	2	0	0	2	F	HP/GS	-	-	-
18	LAW2017	Media Laws and Ethics	2	0	0	2	F	HP/GS	-	-	-
Mathematics Basket											
1	MAT2008	Mathematical Reasoning	3	0	0	3	S	-	-	-	-
2	MAT2014	Advanced Business Mathematics	3	0	0	3	S	-	-	-	-
3	MAT2041	Functions of Complex Variables	3	0	0	3	S	-	-	-	-
4	MAT2042	Probability and Random Processes	3	0	0	3	S	-	-	-	-
5	MAT2043	Elements of Number Theory	3	0	0	3	S	-	-	-	-
6	MAT2044	Mathematical Modelling and Applications	3	0	0	3	S	-	-	-	-
Mechanical Basket											
1	MEC1001	Fundamentals of Automobile Engineering	3	0	0	3	F	-	-	-	-
2	MEC1002	Introduction to Matlab and Simulink	3	0	0	3	S/EM	-	-	-	-
3	MEC1003	Engineering Drawing	1	0	4	3	S	-	-	-	-

4	MEC2001	Renewable Energy Systems	3	0	0	3	F	ES	-	-	-
5	MEC2002	Operations Research & Management	3	0	0	3	F	-	-	-	-
6	MEC2003	Supply Chain Management	3	0	0	3	S/ EM/ EN	-	-	-	-
7	MEC2004	Six Sigma for Professionals	3	0	0	3	S/EM	-	-	MEC2008	-
8	MEC2005	Fundamentals of Aerospace Engineering	3	0	0	3	F	-	-	-	-
9	MEC2006	Safety Engineering	3	0	0	3	S/EM	ES	-	-	-
10	MEC2007	Additive Manufacturing	3	0	0	3	F/EM	-	-	-	-
11	MEC3069	Engineering Optimisation	3	0	0	3	S/EM	-	-	-	-
12	MEC3070	Electronics Waste Management	3	0	0	3	F/S	ES	-	-	-
13	MEC3071	Hybrid Electric Vehicle Design	3	0	0	3	S/EM	ES	-	-	-
14	MEC3072	Thermal Management of Electronic Appliances	3	0	0	3	S/EM	-	-	-	-
15	MEC3200	Sustainable Technologies and Practices	3	0	0	3	S/EM	-	-	-	-
16	MEC3201	Industry 4.0	3	0	0	3	S/EM	-	-	-	-
Petroleum Basket											
1	PET1011	Energy Industry Dynamics	3	0	0	3	FC	ES	-	NIL	-
2	PET1012	Energy Sustainability Practices	3	0	0	3	FC	ES	-	NIL	-
Physics Basket											
1	PHY1003	Mechanics and Physics of Materials	3	0	0	3	FC / SD				
2	PHY1004	Astronomy	3	0	0	3	FC				
3	PHY1005	Game Physics	2	0	2	3	FC / SD				
4	PHY1006	Statistical Mechanics	2	0	0	2	FC				
5	PHY1007	Physics of Nanomaterials	3	0	0	3	FC				
6	PHY1008	Adventures in nanoworld	2	0	0	2	FC				
7	PHY2001	Medical Physics	2	0	0	2	FC	ES			
8	PHY2002	Sensor Physics	1	0	2	2	FC / SD				
9	PHY2003	Computational Physics	1	0	2	2	FC				
10	PHY2004	Laser Physics	3	0	0	3	FC	ES			
11	PHY2005	Science and Technology of Energy	3	0	0	3	FC	ES			
12	PHY2009	Essentials of Physics	2	0	0	2	FC				
Management Basket- I											
1	MGT2007	Digital Entrepreneurship	3	0	0	3	S/EM/ EN	-	-	-	-
2	MGT2015	Engineering Economics	3	0	0	3	S	-	-	-	-
3	MGT2023	People Management	3	0	0	3	S/EM/ EN	HP	-	-	-
4	ECO1001	Introduction to Economics	3	0	0	3	S/EM/ EN	HP			
Management Basket- II											
1	MGT1001	Introduction to Psychology	3	0	0	3	F	HP	-	-	-
2	MGT1002	Business Intelligence	3	0	0	3	EN	-	-	-	-
3	MGT1003	NGO Management	3	0	0	3	S	-	-	-	-
4	MGT1004	Essentials of Leadership	3	0	0	3	EM/ EN	GS/ HP	-	-	-
5	MGT1005	Cross Cultural Communication	3	0	0	3	S/EM/ EN	HP	-	-	-
6	MGT2001	Business Analytics	3	0	0	3	S/ EM/EN	-	-	-	-
7	MGT2002	Organizational Behaviour	3	0	0	3	F	HP	-	-	-
8	MGT2003	Competitive Intelligence	3	0	0	3	S	-	-	-	-
9	MGT2004	Development of Enterprises	3	0	0	3	S/EM/ EN	-	-	-	-
10	MGT2005	Economics and Cost Estimation	3	0	0	3	S/EM	-	-	-	-

11	MGT2006	Decision Making Under Uncertainty	3	0	0	3	S	-	-	-	-
12	MGT2008	Econometrics for Managers	3	0	0	3	S	-	-	-	-
13	MGT2009	Management Consulting	3	0	0	3	S/EM/ EN	-	-	-	-
14	MGT2010	Managing People and Performance	3	0	0	3	S/EM/ EN	HP/GS	-	-	-
15	MGT2011	Personal Finance	3	0	0	3	F	-	-	-	-
16	MGT2012	E Business for Management	3	0	0	3	S/EM	-	-	-	-
17	MGT2013	Project Management	3	0	0	3	EN / EM	GS/HP/ ES	-	-	-
18	MGT2014	Project Finance	3	0	0	3	EN / EM	HP	-	-	-
19	MGT2016	Business of Entertainment	3	0	0	3	EM/ EN	-	-	-	-
20	MGT2017	Principles of Management	3	0	0	3	S/EM/ EN	-	-	-	-
21	MGT2018	Professional and Business Ethics	3	0	0	3	S/EM/ EN	HP	-	-	-
22	MGT2019	Sales Techniques	3	0	0	3	S/EM/ EN	HP	-	-	-
23	MGT2020	Marketing for Engineers	3	0	0	3	S/EM/ EN	HP	-	-	-
24	MGT2021	Finance for Engineers	3	0	0	3	S/EM/ EN	HP	-	-	-
25	MGT2022	Customer Relationship Management	3	0	0	3	S/EM/ EN	HP	-	-	-
Media Studies Basket												
1	BAJ3050	Corporate Filmmaking and Film Business	0	0	4	2	EM	HP	-	-	-
2	BAJ3051	Digital Photography	2	0	2	3	EM	HP	-	-	-
3	BAJ3055	Introduction to News Anchoring and News Management	0	0	2	1	EM	-	-	-	-

Table 3.8 : List of Mandatory Courses (MAC)						
S.No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	CHE7601	Environmental Studies	-	-	-	0
2	LAW7601	Indian Constitution	-	-	-	0
3	CIV7601	Universal Human Values and Ethics	-	-	-	0
					Total No. of Credits	**0**

21. List of MOOC Courses for Computer Science and Engineering Program of 12 weeks

21.1 Presidency University students are given the opportunity to study abroad in International Universities through a selection process coordinated by the Office of International Affairs (OIA). Such selected students need to complete their credits for the semester that they are abroad in the following way:

21.1.1.1. The student needs to study and complete School Core and Program Core Courses in offline mode only.

21.1.1.2. Massive Open Online Course (MOOC) courses maybe given for Open Elective and Discipline Elective Courses. These courses need to be approved by the concerned BOS and Academic Council from time to time.

21.1.1.3. SWAYAM/other approved MOOCs shall be approved by the concerned Board of Studies and placed in the concerned PRC.

21.1.1.4. Student shall register for these courses in the ERP of Presidency University.

21.1.1.5. For these MOOC courses faculty coordinators are identified. These faculty should have undergone similar MOOC courses and therefore should be familiar with the mode of class conduction, types of assessments and evaluation procedures.

21.1.1.6. Study materials shall be provided to the students as video lectures shared by the MOOCs Coordinator(s), or the students may access the approved MOOCs Portal directly. The mode of class conduction is determined by the MOOCs coordinator(s) as detailed in the Course Catalogue and Course Plan.

21.1.1.7. The question paper shall be prepared by the MOOCs coordinator(s).
21.1.1.8. Students write the exams in online mode. These exams are scheduled and conducted by the School.
21.1.1.9. Results are evaluated by School and given to the Office of the Controller of Examinations (CoE).
21.1.1.10. The details of the duration, credits and evaluation are given below:

Sl#	Duration	Credits	Evaluation
1.	12 weeks	3	Continuous Assessment –50 Marks Mid Term –50 Marks End Term-100 Marks
2.	8 weeks	2	Mid Term-50 Marks End Term-100 Marks
3	4 weeks	1	End Term-100 Marks

21.2 List of MOOC – B.Tech. Computer Science and Engineering Program.							
Table 3.9 : MOOC Professional Elective Courses for B.Tech. Computer Science and Engineering Program							
4 weeks (01 credit) / 8 weeks (02 credits) / 12 weeks (03 credits)							
Sl.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**
1	CSE3111	Artificial Intelligence: Search Methods for Problem Solving	3	0	0	3	3
2	CSE3112	Privacy and Security in Online social media	3	0	0	3	3
3	CSE3113	Computational Complexity	3	0	0	3	3
4	CSE3114	Deep Learning for Computer Vision	3	0	0	3	3
5	CSE3115	Learning Analytics Tools	3	0	0	3	3
6	CSE502	Technical Skills in JAVA	0	0	6	3	6
7	CSE503	Technical Skills in Python	0	0	6	3	6
8	CSE504	Comprehensive Technical Skills	0	0	1	5	1
9	CSE505	The Joy of Computing Using Python	3	0	0	3	3
10	CSE3119	Coding Skills in Python	3	0	0	3	3
11	CSE3121	Parallel Computer Architecture	3	0	0	3	3
12	CSE3124	Games and Information	3	0	0	3	3
13	CSE3140	Introduction to Industry 4.0 and Industrial Internet of Things	3	0	0	3	3
14	CSE3142	Affective Computing	3	0	0	3	3
15	CSE3196	Foundations of Cyber Physical Systems	3	0	0	3	3
16	CSE3197	Getting Started with Competitive Programming	3	0	0	3	3
17	CSE3198	GPU Architectures and Programming	3	0	0	3	3
18	CSE3199	Artificial Intelligence: Knowledge Representation and Reasoning	3	0	0	3	3
19	CSE3200	Programming in Modern C++	3	0	0	3	3
20	CSE3201	Circuit Complexity Theory	3	0	0	3	3
21	CSE3202	Basics of Computational Complexity	3	0	0	3	3
22	CSE3212	Introduction to Computer and Network Performance Analysis using Queuing	1	0	0	1	1
23	CSE3213	C Programming and Assembly Language	1	0	0	1	1
24	CSE3214	Python for Data Science	1	0	0	1	1
25	CSE3215	Software Conceptual Design	1	0	0	1	1
26	CSE3117	Industrial Digital Transformation	3	0	0	3	3
27	CSE3118	Blockchain for Decision Makers	3	0	0	3	3
28	CSE3349	Technology for Lawyers	3	0	0	3	3

29	CSE3430	Deep Learning for Natural Language Processing	3	0	0	3	3
30	CSE3431	Machine Learning for Engineering and Science Applications	3	0	0	3	3
31	CSE3432	Algorithms in Computational Biology and Sequence Analysis	3	0	0	3	3
32	CSE3433	Introduction to Large Language Models (LLMs)	3	0	0	3	3
33	CSE3434	Quantum Algorithms and Cryptography	3	0	0	3	3

21.3 MOOC - Open Elective Courses for B. Tech. (Computer Science and Engineering)

Table 3.10: MOOC Open Elective Courses

Open Elective Courses Duration is 4 weeks (01 credit)/ 8 weeks (02 credits)/ 12 weeks (03 credits)

Sl. No.	Course code	Course Name	L	T	P	C
1	BBA2022	Supply Chain digitization	3	0	0	3
2	BBA2021	E Business	3	0	0	3
3	BBB2016	Business Analytics for Management Decisions	3	0	0	3
4	BBB2015	Artificial Intelligence for Investments	3	0	0	3
5	MEC3001 *	Design and Development of Product	1	0	0	1
6	ENG3004 **	Perspectives of Neurolinguistics	1	0	0	1
7	PPS4009 ***	Working in Contemporary Teams	1	0	0	1
8	MGT3001	Data Analysis and Decision Making	3	0	0	3

Note :

** MEC3001 is offered to the students who had 1 credit shortage because of implementation of CBCS system during their 1st year.*

*** ENG3004 is offered to the students who had 2 credits shortage along with the MEC3001 because of implementation of CBCS System during their 1st year.*

**** PPS4009 is offered to only international students in place of Interview Preparedness course of their batch mates.*

22. Recommended Semester Wise Course Structure / Flow including the Programme / Discipline Elective Paths / Options

Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Pre-requisite	Basket
Semester I - PHYSICS Cycle (CSE-Allied (29 Sec))			**15**	**1**	**8**	**19**	**24**			
1	MAT2301	Calculus and Differential Equations	3	1	0	4	4	F	Nil	BSC
2	ENG1900	English for Technical Communication	2	0	0	2	2	S	Nil	HSMC
3	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F	Nil	BSC
4	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F	Nil	BSC
5	MEC1006	Engineering Graphics	2	0	0	2	2	S	Nil	ESC
6	ECE2022	Digital Design	2	0	0	2	2	F/S	Nil	ESC
7	ECE2052	Digital Design Lab	0	0	2	1	2	F/S	Nil	ESC
8	DES1146	Introduction to Design Thinking	1	0	0	1	1	F	Nil	HSMC
9	CSE1500	Computational Thinking using Python	2	0	2	3	4	S	Nil	ESC

10	PPS1025	Industry Readiness Program – I	0	0	2	0	2	S	Nil	MAC
Semester I - CHEMISTRY Cycle (CSE (21 Sec + Engg 10 Sec))			**16**	**1**	**8**	**19**	**25**			
1	MAT2301	Calculus and Differential Equations	3	1	0	4	4	F	Nil	BSC
2	ENG1900	English for Technical Communication	2	0	0	2	2	S	Nil	HSMC
3	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S	Nil	BSC
4	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S	Nil	BSC
5	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S	Nil	ESC
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S	Nil	ESC
7	EEE1250	Basics of Electrical and Electronics Engineering Lab	0	0	2	1	2	F/S	Nil	ESC
8	CIV7601	Universal Human Values and Ethics	0	0	0	0	1	F	Nil	MAC
9	CSE1500	Computational Thinking using Python	2	0	2	3	4	S	Nil	ESC
10	PPS1025	Industry Readiness Program – I	0	0	2	0	2	S	Nil	MAC
Semester II - CHEMISTRY Cycle (CSE-Allied (29 Sec))			**16**	**1**	**10**	**22**	**27**			
1	MAT2402	Probability and Statistics	3	1	0	4	4	F	Nil	BSC
2	ENG2501	Advanced English	2	0	0	2	2	S	Nil	HSMC
3	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S	Nil	BSC
4	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S	Nil	BSC
5	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S	Nil	ESC
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S	Nil	ESC
7	EEE1250	Basics of Electrical and Electronics Engineering Lab	0	0	2	1	2	F/S	Nil	ESC
8	LAW7601	Indian Constitution	0	0	0	0	0	F	Nil	MAC
9	CSE2000	Problem Solving using C	2	0	0	2	2	S	Nil	PCC
10	CSE2001	Problem Solving using C Lab	0	0	4	2	4	S	Nil	PCC
11	PPS1026	Industry Readiness Program – II	0	0	2	0	2	S	Nil	MAC
12	ECE1511	Design Workshop	1	0	2	2	3	S/EM	Nil	ESC
13	CHE7601	Environmental Studies	0	0	0	0	0	F	Nil	MAC
Semester II - PHYSICS Cycle			**15**	**1**	**10**	**22**	**26**			

(CSE (21 Sec + Engg 10 Sec))										
1	MAT2402	Probability and Statistics	3	1	0	4	4	F	Nil	BSC
2	ENG2501	Advanced English	2	0	0	2	2	S	Nil	HSMC
3	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F	Nil	BSC
4	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F	Nil	BSC
5	MEC1006	Engineering Graphics	2	0	0	2	2	S	Nil	ESC
6	ECE2022	Digital Design	2	0	0	2	2	F/S	Nil	ESC
7	ECE2052	Digital Design Lab	0	0	2	1	2	F/S	Nil	ESC
8	DES1146	Introduction to Design Thinking	1	0	0	1	1	F	Nil	HSMC
9	CSE2000	Problem Solving using C	2	0	0	2	2	S	Nil	PCC
10	CSE2001	Problem Solving using C Lab	0	0	4	2	4	S	Nil	PCC
11	PPS1026	Industry Readiness Program – II	0	0	2	0	2	S	Nil	MAC
12	ECE1511	Design Workshop	1	0	2	2	3	S/EM	Nil	ESC
13	CHE7601	Environmental Studies	0	0	0	0	0	F	Nil	MAC
Semester 3			**18**	**1**	**8**	**22**	**27**			
1	MAT2303	Linear Algebra and Vector Calculus	3	1	0	4	4	EM	Nil	BSC
2	CSE2251	Data Communication and Computer Networks	3	0	0	3	3	S	Nil	PCC
3	CSE2252	Data Communication and Computer Networks Lab	0	0	2	1	2	S	Nil	PCC
4	CSE2253	Data Structures	3	0	0	3	3	S	Nil	PCC
5	CSE2254	Data Structures Lab	0	0	2	1	2	S	Nil	PCC
6	CSE2255	Object Oriented Programming Using Java	3	0	0	3	3	S/EM	Nil	PCC
7	CSE2256	Object Oriented Programming Using Java Lab	0	0	2	1	2	S/EM	Nil	PCC
8	CSE2257	Computer Organization and Architecture	3	0	0	3	3	S	Nil	PCC
9	FIN1002	Essentials of Finance	3	0	0	3	3	S	Nil	HSMC
10	CIV7601	Universal Human Values and Ethics	0	0	0	0	0	S	Nil	MAC
11	APT4002	Introduction to Aptitude	0	0	2	0	2	AT	Nil	MAC
Semester 4			**18**	**2**	**12**	**25**	**31**			
1	MAT2404	Discrete Mathematics	3	1	0	4	4	F	Nil	BSC
2	CSE2258	Web Technologies	3	0	0	3	3	S/EM	Nil	PCC
3	CSE2259	Web Technologies Lab	0	0	2	1	2	S/EM	Nil	PCC

4	CSE2260	Database Management Systems	3	0	0	3	3	S	Nil	PCC
5	CSE2261	Database Management Systems Lab	0	0	2	1	2	S	Nil	PCC
6	CSE2500	Data Analytics	3	0	0	3	2	S/EM	MAT2402	PCC
7	CSE2501	Data Analytics Lab	0	0	2	1	2	S/EM	MAT2402	PCC
8	CSE2262	Analysis of Algorithms	3	1	0	4	4	S	Nil	PCC
9	CSE2263	Analysis of Algorithms Lab	0	0	2	1	2	S	Nil	PCC
10	CSE2264	Essentials of AI	3	0	0	3	3	S/EM	Nil	ESC
11	CSE2265	Essentials of AI Lab	0	0	2	1	2	S/EM	Nil	ESC
12	APT4004	Aptitude Training - Intermediate	0	0	2	0	2	AT	Nil	MAC
Semester 5			**18**	**0**	**8**	**24**	**26**			
1	CSE2266	Theory of Computation	3	0	0	3	3	S	Nil	PCC
2	CSE2502	Cryptography and Network Security	3	0	0	3	3	S	CSE2251	PCC
3	CSE2267	Machine Learning techniques	3	0	0	3	3	S/EM	Nil	PCC
4	CSE2268	Machine Learming Techniques Lab	0	0	2	1	2	S/EM	Nil	PCC
5	CSE2269	Operating Systems	3	0	0	3	3	S	Nil	PCC
6	CSE2270	Operating Systems Lab	0	0	2	1	2	S/EM	Nil	PCC
7	CSE2503	Scalable Application Development using Java	3	0	0	3	3	S/EM	CSE2255	PCC
8	CSE2504	Scalable Application Development using Java Lab	0	0	2	1	2	S/EM	CSE2256	PCC
9	CSEXXXX	Professional Elective – I	3	0	0	3	3	S/EM	Nil	PEC
10	CSE7000	Internship	0	0	0	2	0	S/EM	Nil	PRW
11	APT4006	Logical and Critical Thinking	0	0	2	1	2	AT	Nil	MAC
Semester 6			**16**	**0**	**12**	**22**	**28**			
1	CSE2271	Software Design and Development	3	0	0	3	3	S	Nil	CSE2271
2	CSE2272	Cloud Computing	2	0	0	2	2	S/EM	Nil	CSE2272
3	CSE2273	Cloud Computing Lab	0	0	2	1	2	S/EM	Nil	CSE2273
4	CSE2505	Mobile Application Development	2	0	0	2	2	S/EM	CSE2255	CSE2505
5	CSE2506	Mobile Application Development Lab	0	0	4	2	4	S/EM	CSE2256	CSE2506
6	CSE2274	Competitive Programming and Problem Solving	0	0	4	2	4	S/EM	Nil	CSE2274
7	CSEXXXX	Professional Elective – II	3	0	0	3	3	S	Nil	CSEXXXX

8	CSEXXXX	Professional Elective – III	3	0	0	3	3	S	Nil	CSEXXXX
9	XXXXXXX	Open Elective – I	3	0	0	3	3	S	Nil	XXXXXXX
10	APT4005	Aptitude for Employability	0	0	2	1	2	AT	Nil	APT4005
Semester 7			**12**	**0**	**2**	**17**	**14**			
1	CSEXXXX	Professional Elective – IV	3	0	0	3	3	S	Nil	PEC
2	CSEXXXX	Professional Elective – V	3	0	0	3	3	S	Nil	PEC
3	CSEXXXX	Professional Elective – VI	3	0	0	3	3	S	Nil	PEC
4	XXXXXXX	Open Elective – II	3	0	0	3	3	S	Nil	OEC
5	CSE7100	Mini Project	0	0	0	4	0	S	Nil	PRW
6	PPS3018	Preparedness for Interview	0	0	2	1	2	AT	Nil	HSMC
Semester 8			**0**	**0**	**0**	**10**	**0**			
1	CSE7300	Capstone Project	0	0	0	10	0	S/EM	Nil	PRW
		Total	**114**	**5**	**56**	**160**	**174**			

23. Course Catalogue

Course Catalogue of all Courses Listed including the Courses Offered by other School / Department and Discipline / Programme Electives – Course Code, Course Name, Prerequisite, Anti-requisite, Course Description, Course Outcome, Course Content (with Blooms Level, CO, No. of Contact Hours), Reference Resources.

urse Code: AT2301	**Course Title: Calculus and Differential Equations** **Type of Course : BSC**	**T- P- C**	**3**	**1**	**0**	**4**
rsion No.						
urse Pre-requisites						
ti-requisites	L					
urse Description	Calculus and differential equations are used ubiquitously throughout mathematics, statistics and operations research. In this course, students can be able to build upon the foundations of calculus established to greatly enhance their repertoire of theory and practice in these areas. The application of calculus and differential equations in the description and modelling of real-world problems will also be considered. This unit will extend the problem-solving skills, range of knowledge and use of techniques in differential and integral calculus. The course focuses on the concepts of Calculus and Differential Equations with reference to specific engineering problems. The course is of both conceptual and analytical type in nature.					
urse Objective	The goal of the course Calculus and Differential Equations is to facilitate the students with a concrete foundation of differential calculus and to solve the first and higher-order ordinary differential equations enabling them to acquire the knowledge of these mathematical tools.					
urse Out Comes	On successful completion of the course the students shall be able to: 1. Apply the knowledge of calculus to solve problems related to polar curves and its applications in determining the bentness of a curve. 2. Apply the principles of integral calculus to evaluate integrals. 3. Learn the notion of partial differentiation to calculate rate of change of multivariate functions and solve problems related to composite functions and Jacobian. 4. Solve first-order linear/nonlinear ordinary differential equations analytically using standard methods.					
urse Content:						

dule 1	**ferential Calculus**		**(10 Classes)**

Polar Coordinates, polar curves, angle between radius vector and the tangent, angle between two curves, pedal equations, curvature and radius of curvature.
Limit, continuity and partial derivatives, directional derivatives, total derivative; Tangent plane and normal line; Maxima, minima and saddle points; Method of Lagrange multipliers; Gradient, curl and divergence.

dule 2	**egral Calculus**	**Assignment**	**(10 Classes)**

Evolutes and involutes; Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate surface areas and volumes of revolutions. Rolle's theorem, Mean value theorems, Taylor's and Maclaurin theorems with remainders; Indeterminate forms and L'Hospital's rule; Maxima and minima.

dule 3	**ltivariable Calculus**		**(10 lectures)**

Multiple Integration: Double integrals (Cartesian), change of order of integration in double integrals, Change of variables (Cartesian to polar), Applications: areas and volumes, Center of mass and Gravity (constant and variable densities); Triple integrals (Cartesian), orthogonal curvilinear coordinates, Simple applications involving cubes, sphere and rectangular parallelepipeds; Scalar line integrals, vector line integrals, scalar surface integrals, vector surface integrals, Theorems of Green, Gauss and Stokes.

dule 4	**ferential Equations**	**Assignment**	**(15 lectures)**

Definition, types of Differential Equations, Applications, Variable Separable, Homogeneous, Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.
Linear differential equations of second and higher order with constant coefficients - Non-Homogeneous term of the type $Q(x) = e^{ax}$, Sin ax, Cosax, $e^{ax}v(x)$, $x^n v(x)$ - Method of variation of parameters.

rgeted Application & Tools that can be used:

fferential calculus is used extensively in science and engineering. It can solve problems related to motion, velocity, acceleration, angles of incline or curve on a surface, etc.

fferential Equations are used to model the behavior of electromagnetic fields, including in the design of antennas, microwave ovens, and other devices. Biology: PDEs are used to model biological processes, such as the spread of diseases and the development of biological tissues.

ols Used: Python.

signment:

Select any one simple differential equation pertaining to the respective branch of engineering, identify the dependent and independent variable – Obtain the solution and compare the solution sets by varying the values of the dependent variable.

xt Book

1. Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition
2. B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

ferences:

1. Victor Henner, Tatyana Belozerova, Mickhail Khenner, Ordinary and Partial Differential Equations, CRC Press, Edition, 2013.
2. Walter Ledermann, Multiple integrals, Springer, 1st edition
3. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
4. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_103205
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_106839
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_61605
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_134719
5. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
6. https://www.scu.edu.au/study-at-scu/units/math1005/2022/

pics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of calculus and differential equation with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem solving**.** This is attained through the assessment component mentioned in course handout.

<table>
<tr><td>urse Code:
Y2501</td><td colspan="3">Course Title: Optoelectronics and Quantum Physics
pe of Course: BSC</td><td>-T-P-C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>rsion No.</td><td colspan="8"></td></tr>
<tr><td>urse Pre-requisites</td><td colspan="8">L</td></tr>
<tr><td>ti-requisites</td><td colspan="8">L</td></tr>
<tr><td>urse Description</td><td colspan="8">The purpose of this course is to enable the students to understand the fundamentals, working and applications of optoelectronic devices and to develop the basic abilities to appreciate the applications of advanced microscopy and quantum computers. The course develops the critical thinking and analytical skills.</td></tr>
<tr><td>urse Out Comes</td><td colspan="8">On successful completion of the course the students shall be able to:

CO1: To understand the concepts of electrical conducting properties of metal, semiconductor and superconductivity.

CO2: To understand the principles of quantum mechanics.

CO3: Discuss the quantum concepts used in quantum computers.

CO4: Explain the applications of lasers and optical fibers in various technological fields.</td></tr>
<tr><td>urse Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of “Applied Physics for Computer Science Cluster “and to attain the basic knowledge related to quantum mechanics and computation.</td></tr>
<tr><td>urse Content:</td><td colspan="8"></td></tr>
<tr><td>dule 1</td><td>ctrical Conductivity of Solids and Semiconducting Devices</td><td>signment</td><td colspan="2">ta collection on efficiency of solar cells.</td><td colspan="4">Sessions</td></tr>
<tr><td colspan="9">Topics: Classification of materials based on bandgap, Fermi energy and Fermi level, Fermi level in semiconductors, Law of mass action, Electrical conductivity of a semiconductor, Hall effect, Superconductivity, p-n junctions, Zener diode, Solar cells, I-V characteristics, and LEDs</td></tr>
<tr><td>dule 2</td><td>antum Mechanics</td><td>signment</td><td colspan="2"></td><td colspan="4">Sessions</td></tr>
<tr><td colspan="9">Topics: Introduction, de-Broglie hypothesis, Heisenberg’s uncertainty principle- statement and physical significance. Wave function-properties and physical significance. Schrodindger’s time independent wave equation, Probability density and normalization of wave function. Wave Function in Ket Notation: Matrix form of wave function, Identity operator, Determination of I|0> and I|1>, Pauli Matrices and its operations on 0 and 1 states, Mention of Conjugate and Transpose, Unitary Matrix U, Examples: 2x2 Matrices and their multiplication (Inner Product), Probability, Orthogonality</td></tr>
<tr><td>dule 3</td><td>antum Computing</td><td>m paper</td><td colspan="2">minar on quantum computers.</td><td colspan="4">Sessions</td></tr>
<tr><td colspan="9">Topics: Introduction to quantum computing, Moore’s law & its end, Differences between classical and quantum computing, Concept of Qubit and its properties, . Representation of qubit by Bloch sphere, Quantum Gates: Single Qubit Gates: Quantum Not Gate, Pauli Z Gate, Hadamard Gate, Phase Gate (or S Gate), T Gate. Multiple Qubit Gates: Controlled gate - CNOT Gate, (Discussion for 4 different input states). Representation of Swap gate, Controlled - Z gate, Toffoli gate. Problems.</td></tr>
<tr><td>dule 4</td><td>sers And Optical Fibers</td><td>m paper</td><td colspan="2">se study on medical applications of Lasers.</td><td colspan="4">Sessions</td></tr>
<tr><td colspan="9">Topics: Interactions of radiations with matter, expression for energy density of a system under thermal equilibrium in terms of Einstein’s coefficients, conditions for LASER action using Einstein’s coefficients, Characteristics of laser, conditions and requisites of laser, Principle of optical fibers, Numerical aperture and acceptance angle (Qualitative), Attenuation, Applications: Point to point communication with block diagram, application of optical fibers in endoscopy.</td></tr>
<tr><td colspan="9">rgeted Application & Tools that can be used:
1. Areas of application are optoelectronics industry, Solar panel technologies, quantum computing software, electronic devices using transistors and diodes, memory devices, endoscopy, SQUIDS in MRI, Advanced material characterizations using SEM and STM.</td></tr>
</table>

2. Origin, excel and Mat lab soft wares for programming and data analysis.

oject work/Assignment: Mention the Type of Project /Assignment proposed for this course

Assessment Type

- Midterm exam
- Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screen shot accessing digital resource.)
- Quiz
- End Term Exam
- Self-Learning

Prepare a comprehensive report on non-conventional energy resources in Karnataka and their pros and cons.

Write a report on importance of quantum entanglement in supercomputers.

xt Book

1. Engineering Physics by Avadhanalu, Revised edition, S. Chand Publications,2024.
2. Quantum Computation and Quantum Information, Michael A. Nielsen & Isaac L. Chuang, Cambridge Universities Press, 2010 Edition

References:

1. Elementary Solid state Physics: Principles and Applications by M.A. Omar, 1st Edition, Pearson Publications, 2002.
2. Principles of Quantum Mechanics by R Shankar, 2nd edition, springer Publications, 2011.
3. Optoelectronics: An Introduction by John Wilson and John Hawkes, 3rd edition, Pearson Publications, 2017.
4. Engineering Physics by Gaur and Gupta, Dhanpat Rai Publications, 2012.
5. Introduction to Quantum Mechanics, David J Griffiths, Cambridge University Press, 2019

E-Resourses:

1. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=553045&site=ehost-live
2. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=833068&site=ehost-live
3. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=323988&site=ehost-live
4. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1530910&site=ehost-live
5. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=486032&site=ehost-live

Topics relevant to "SKILL DEVELOPMENT": Fundamentals of materials, Lasers and optical fibers. for Skill Development through Participative Learning Techniques. This is attained through the Assignment/ Presentation as mentioned in the assessment component in course handout.

urse Code: IY2504	Course Title: Optoelectronics and Quantum Physics Lab pe of Course: BSC	T-P-C				
rsion No.						
urse Pre-requisites	L					
ti-requisites	L					
urse Description	The laboratory provides an opportunity to validate the concepts taught and enhances the ability to use the concepts for technological applications. The laboratory tasks aim to develop following skills: **An attitude of enquiry, confidence and ability to tackle new problems, ability to interpret events and results, observe and measure physical phenomena, select suitable equipment, instrument and materials, locate faults in systems.**					
urse Out Comes	On successful completion of the course the students shall be able to: CO1: To understand electrical and optical properties of materials CO2: Interpret the results of various experiments to verify the concepts used in optoelectronics and advanced devices.					
urse Objective	The objective of the course is to familiarize the learners with the concepts of " Applied Physics for Computer Science Cluster "and attain **Skill Development** through **Experiential Learning** techniques					

t of Laboratory Tasks:

Experiment No. 1: Experimental errors and uncertainty using excel
Level 1: Calculation of accuracy and precision of a given data
vel 2: propagation of errors in addition, subtraction, multiplication and division.

Experiment N0 2: To determine the wavelength of semiconductor diode Laser and to estimate the particle size of lycopodium powder using diffraction.
vel 1: Determination of Wavelength of Laser
vel 2: Finding the particle size of lycopodium powder.

periment No. 3: To determine the proportionality of Hall Voltage, magnetic flux density and the polarity of Charge carrier.
vel 1: To determine the proportionality of Hall Voltage and magnetic flux density
vel 2: To determine the polarity of Charge carrier.

Experiment No. 4: To study the I-V characteristics of a given zener diode in forward and reverse bias conditions.
vel 1: To study I –V characteristics of the given Zener diode in reverse bias and to determine break down voltage.
vel 2: To study I –V characteristics of the given Zener diode in forward bias and to determine knee voltage and forward resistance.

Experiment No. 5: To study input and output characteristics of a given Transistor.
vel 1: To determine the input resistance of a given transistor.
vel 2: To determine current transfer characteristics and transistor parameters of a given transistor.

Experiment No. 6: Determination of Fermi energy and Fermi temperature of a given metal and bimetallic wire.
Level 1: Determination of Fermi energy and Fermi temperature of given metal wire.
Level 2: Determination of Fermi energy and Fermi temperature of given bimetallic wire.

periment No. 7: To study the I-V characteristics and I-R characteristics of a solar cell as a function of the irradiance.
vel 1: To study the I-V characteristics
vel 2: I-R characteristics of a solar cell as a function of the irradiance.

periment No. 8: Calculate the numerical aperture and study the losses that occur in optical fiber cable. .
vel 1: Calculate the numerical aperture.
vel 2: study the losses that occur in optical fiber cable.

periment No. 9: Plotting I-V characteristics in forward and reverse bias for LEDs
l Determination of knee voltage.
vel 1: Plotting I-V characteristics in forward and reverse bias for LEDs
vel 2: Determination of knee voltage.

periment No. 10: Determination of Stefan's constant and verification of Stefan-Boltzmann Law.
vel 1: Determination of Stefan's constant
vel 2: Verification of Stefan-Boltzmann Law.

periment No. 11: Determination of dielectric constant of given materials.
vel 1: Determination of Stefan's constant
vel 2: compare the obtain results with other materials

periment No. 12: determine the wavelength of monochromatic light, such as sodium light, using Newton's rings.
vel 1: Determination of wavelength
vel 2: determine the radius of curvature of the Plano-convex lens.

rgeted Application & Tools that can be used:

3. Areas of application are optoelectronics industry, Solar panel technologies, quantum computing software, electronic devices using transistors and diodes, memory devices, endoscopy, SQUIDS in MRI, Advanced material characterizations using SEM and STM.
4. Origin, excel and Mat lab soft wares for programming and data analysis.

oject work/Assignment: Mention the Type of Project /Assignment proposed for this course

Assessment Type

- Midterm exam
- Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screen shot accessing digital resource.)
- Quiz
- End Term Exam
- Self-Learning

Prepare a comprehensive report on non-conventional energy resources in Karnataka and their pros and cons.
Write a report on importance of quantum entanglement in supercomputers.

Course Code: **MEC1006**	**Course Title: Engineering Graphics** **Type of Course: School Core & Theory Only**	**L- T-P-C**	2	0	0	2
Version No.	1.2					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	The course is designed with the objective of giving an overview of engineering graphics. It is introductory in nature and acquaints the students with the techniques used to create engineering drawings. The course emphasizes on projection of points, lines, planes and solids and isometric projections.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Engineering Graphics" and attain SKILL DEVELOPMENT through Problem solving methodologies.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Demonstrate competency of Engineering Graphics as per BIS conventionsand standards. 2. Comprehend the theory of projection for drawing projections of Points, Lines andPlanes under different conditions. 3. Prepare multiview orthographic projections of Solids by visualizing them in different positions. 4. Prepare pictorial drawings using the principles of isometric projections to visualizeobjects in three dimensions.					

Course Content:				
Module 1	Introduction to Drawing	Assignment	Standard technical drawing	**02 Sessions**
Topics: Introduction, drawing instruments and their uses, relevant BIS conventions and standards, Lettering, Lineconventions, dimensioning, Selection of drawing sheet size and scale.				
Module 2	Orthographic projections of Points, Straight Linesand Plane Surfaces	Assignment	Projection methods Analysis	**10 Sessions**
Topics: Introduction, Definitions – Elements of projection and methods of projection, Planes of projection, reference line and conventions adopted. First angle and third angle projections. Projection of Points inall 4 quadrants. Projections of Straight Lines (located in first quadrant/first angle projection only): True and apparent lengths, true and apparent Inclinations to reference planes. (No application problems). Projection of Plane surfaces (First angle projection): Regular plane surfaces – triangle, square, rectangle, pentagon, hexagon and circle – in different positions inclined to both the planes using change of position method only.				
Module 3	Orthographic Projections of Solids	Assignment	Multi-view drawing Analysis	**10 Sessions**
Topics: **Introduction, Projection of right regular prisms, pyramids, cone, hexahedron and tetrahedron in different positions (Problems resting on HP only and First angle projection).**				

Module 4	Isometric Projections of Solids (Using isometric scale only)	Assignment	Spatial Visualization	**8 Sessions**

Topics:
Introduction, Isometric scale, Isometric projections of right regular prisms, cylinders, pyramids, cones and their frustums, spheres and hemispheres, hexahedron (cube), and combination of 2 solids, conversion of orthographic view to isometric projection of simple objects. [8 Hours: Application Level]

Text Book:
1.N. D. Bhatt, "Engineering Drawing: Plane and Solid Geometry," Charotar Publishing House Pvt. Ltd.

References:
1. K.R. Gopalakrishna, "Engineering Graphics", Subhash Publishers, Bangalore.
2. D. M. Kulkarni, A. P. Rastogi, A. K. Sarkar, "Engineering Graphics with AutoCAD," Prentice Hall.
3. D. A. Jolhe, "Engineering Drawing with Introduction to AutoCAD," Tata McGraw Hill.

Web resources:
https://nptel.ac.in/courses/112103019

Topics relevant to "SKILL DEVELOPMENT": Projection in first and third angle for **SKILL DEVELOPMENT** through **Problem Solving methodologies**. This is attained through the assessment component mentioned in the course handout.

ENG1900	English for Technical Communication	L- T- P- C	2	0	0	2
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	NIL					
Course Description	This course enhances the technical communication skills of BTech students, focusing on clarity, precision, and conciseness in academic and professional settings. Students will learn to differentiate between general and technical communication, analyze technical content, develop structured writing skills, and deliver effective presentations. Through interactive activities such as TED Talk analyses, report writing, and presentation practice, the course provides hands-on experience for real-world applications. By the end, students will be equipped to communicate complex technical information effectively in various professional contexts.					
Course Outcomes	On successful completion of the course the students shall be able to: 1. Differentiate between general and technical communication. 2. Explain key reading comprehension techniques to enhance understanding of technical texts. 3. Write clear, concise, and well-structured technical reports and documents. 4. Deliver technical presentations and implement peer feedback for continuous improvement. 5. Explain ethical practices in digital communication for professional use.					

Course Content: Theory

Module 1	Technical communication	Quiz	Listening	9 Hours

Introduction to Communication
Technical vs. General Communication
Characteristics of technical communication
Importance of clarity, precision, and objectivity

Activity:

- Watching TED Talks/videos to identify differences in technical and general vocabulary

Module 2	Technical Reading	Assignment	Reading	12 Hours

Reading Comprehension
Note making & Notetaking
Content Analysis

Activity:

- Reading technical articles and answering comprehension questions
- Note making techniques

Module 3	Technical Writing	Assignment	Writing	12hours

Paragraph Writing
Structure of a paragraph (topic sentence, supporting details, coherence)
Report Writing
Structure of technical and project reports (Introduction, Methods, Results, Discussion)

Activity:

- Writing a structured paragraph on a technical topic
- Writing project reports

Module 4	Professional Presentation	Presentation	Speaking	12Hours

Introduction to Presentation Skills
Preparing a Presentation

- Structuring content (Introduction, Body, Conclusion)
- Designing effective slides (Text. visual aids, readability, and impact)

Delivering a Presentation

- Engagement techniques, Storytelling, narration, pitching ideas handling Q&A
- Conviction, commitment, generating interest through enthusiasm

Demonstration & Practice

- Giving presentations on topics based on their academic interest
- Evaluating and providing peer feedback

Activity:

- Analyze a real-world engineering issue and present solutions using a structured approach.

Targeted Application & Tools that can be used: YouTube, Instagram, Quill Bot, Grammarly, & Padlet.

References:

Text books:

1. Gupta, R.C. *Technical Communication.* 2nd ed., Cambridge University Press, 2021.
2. Lannon, John M., and Laura J. Gurak. *Technical Communication.* 15th ed., Pearson, 2022.

Reference Books:

1. Gerson, Sharon J., and Steven M. Gerson. Technical Communication: Process and Product. 9th ed., Pearson, 2020.
2. Lannon, John M., and Laura J. Gurak. Technical Communication. 15th ed., Pearson, 2022.
3. Markel, Mike, and Stuart A. Selber. Technical Communication. 13th ed., Bedford/St. Martin's, 2020.

Web Resources:

1. https://owl.purdue.edu/owl/subject_specific_writing/technical_writing.
2. https://journals.ieeeauthorcenter.ieee.org/.
3. https://www.stc.org/.
4. https://ocw.mit.edu/.https://www.ted.com/talks.

Topics Relevant to "employability": Teamwork and Collaboration, Critical Thinking and Problem-Solving

Topics Relevant to "Human Values and Professional Ethics": Critical reasoning, Inclusivity and Fairness

urse Code: E1500	urse Title: **Computational Thinking using Python** pe of Course: ESC	T-P-C	2	)	2	3

rsion No.	
urse Pre-requisites	
ti-requisites	L
urse Description	The course efficiently introduces fundamental ideas including conditionals, loops, functions, lists, strings, and tuples through some inspiring examples. It then discusses dynamic programming like handling exceptions and file usage. In terms of data structures, the course covers Python dictionaries, classes, and objects for constructing user-defined datatypes like linear and binary search.
urse Object	The objective of the course is to familiarize the learners with the concepts of **Computational Thinking using Python** and attain **Skill Development** through **Participative Learning** techniques.
urse Out Comes	On successful completion of the course the students shall be able to: 1) Describe algorithmic solutions for basic computing issues.. (Understand) 2) Explain data types and operators. (Understand) 3) Demonstrate control structures and Functions. (Apply) 4) Apply the data structures for the given data. (Apply) 5) Demonstrate the file operations. (Apply)

urse Content:

dule 1	**Computational Thinking And Problem Solving**	signment		gramming	**6 Sessions**

Topics:

Fundamentals of Computing– Identification of Computational Problems Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi

dule 2	**tatypes, Expressions, Statements**	signment		gramming	**essions**

Topics:

Python interpreter and interactive mode,debugging; values and types: int, float, boolean, string , and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

dule 3	**ntrol flow, Functions, Strings**	signment		gramming	**essions**

Topics:

Conditionals:Boolean values and operators, conditional (if), alternative (if else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

dule 4	**sts, Tuples, Dictionaries**	signment		gramming	**essions**

pics:

ts: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing- list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.

dule 5	**es**	signment		gramming	**essions**

es and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter's age validation, Marks range validation (0-100).

oject work/Assignment:

1. **Assignment 1 on (Module 1 and Module 2)**
2. **Assignment 2 on (Module 3 and Module 4 & 5)**

Text Book

1. Paul Deitel and Harvey Deitel, "Python for Programmers", Pearson Education, 1st Edition, 2021
2. Eric Matthes, Python Crash Course,: A Hands-On, Project-Based Introduction to Programming, 3rd Edition, 2023

References

1. Allen B. Downey, "Think Python: How to Think like a Computer Scientist", 2nd Edition, O'Reilly Publishers, 2016.
2. Karl Beecher, "Computational Thinking: A Beginner's Guide to Problem Solving and Programming", 1st Edition, BCS Learning & Development Limited, 2017.

Web Resources

W1. https://onlinecourses.nptel.ac.in/noc20_cs70/preview

Topics relevant to development of "Employability": Data structures using python.

Topics relevant to "PROFESSIONAL ETHICS": Naming and coding convention for simple programs using python.

urse Code: E2022	urse Title: Digital Design pe of Course: ESC	L- T-P- C	2	0	0	2
rsion No.						
urse Pre-requisites	[1] Elements of Electronics/Electrical Engineering, 2] Basic concepts of number representation, Boolean Algebra					
i-requisites	-					
urse Description	The purpose of this course is to enable the students to appreciate the fundamentals of digital logic circuits and Boolean algebra focusing on both combinational and sequential logic circuits. The course emphasizes on minimization techniques for making canonical and low-cost digital circuit implementations. This course deals with analysis and design of digital electronic circuits. The course also creates a foundation for future courses which includes Computer Architecture, Microprocessors, Microcontrollers, and Embedded Systems etc. The course enhances the Design, Implementation and Programming abilities through laboratory tasks. The associated laboratory provides an opportunity to verify the theoretical knowledge.					
urse Objective	The objective of the course is to familiarize the learners with the concepts of Digital Design and attain the SKILL DEVELOPMENT through EXPERIENTIAL LEARNING.					
urse Outcomes	On successful completion of this course the students shall be able to: i. Describe the concepts of number systems, Boolean algebra and logic gates. ii. Apply minimization techniques to simplify Boolean expressions. iii. Demonstrate the Combinational circuits for a given logic iv. Demonstrate the Sequential and programmable logic circuits					
urse Content:						

dule 1	ndamentals of Number systems- Boolean algebra and digital logic	plication Assignment	a Analysis task	06 classes

pics:

Review of Number systems and logic gates, Number base conversions, Overview of Boolean functions and simplifications, two, three, four variable K-Maps- Don't care conditions- Both SOP and POS- Universal Gates (NAND & NOR) Implementations. Introduction to HDL.

dule 2	Boolean function simplification	plication Assignment	a Analysis task	08 Classes

pics:

Introduction to Combinational circuits, Analysis, Design procedure, Binary Adder and Subtractor, Magnitude comparator, Parity generator and checker, Multiplexers-Demultiplexers, Decoders, Encoders and Priority Encoders, HDL Models of combinational circuits.

dule 3	Combinational Logic circuits:	plication Assignment	gramming Task & Data Analysis task	08 Classes

pics:

Introduction to sequential circuits, Storage elements: latches and flip flops, Characteristic tables and equations, excitation table, Analysis of clocked sequential circuits, Mealy & Moore Models of finite state machines - Registers & Counters. HDL Models of Sequential circuits.

geted Application & Tools that can be used:

ital electronics is the foundation of all modern electronic devices such as cellular phones, MP3 players, laptop computers, digital cameras, high definition televisions, Home Automation, Communication in systems in industries Professionally Used Software: HDL/VHDL/Verilog HDL/ OOPS

Text Book(s):

1. Mano, M. Morris and Ciletti Michael D., *"Digital Design"*, Pearson Education, 6^{th} edition
2. Thomas L. Floyd "DIGITAL LOGIC DESIGN" , Pearson Education, fourth edition.

Reference(s):

Reference Book(s):

R1. Jain, R. P., *"Modern Digital Electronics",* McGraw Hill Education (India), 4^{th} Edition

R2. Roth, Charles H., Jr and Kinney Larry L., *"Fundamentals of logic Design"*, Cengage Learning, 7^{th}

Edition

Online Resources (e-books, notes, ppts, video lectures etc.): Book Free Download (studymaterialz.in)

1. eBook1: Mano, M. Morris and Ciletti Michael D., *“Digital Design”*, Pearson Education.
2. {[PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download
}
3. eBook2:Floyd “DIGITAL LOGIC DESIGN” fourth edition- ePub, eBook- [PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org.
4. NPTEL Course- NPTEL :: Electrical Engineering - NOC:Digital Electronic Circuits
5. Digital Logic Design PPT Slide 1 (iare.ac.in)
6. Lab Tutorial: Multisim Tutorial for Digital Circuits - Bing video

CircuitVerse - Digital Circuit Simulator online

Learn Logisim ➠ Beginners Tutorial | Easy Explanation! - Bing video

Digital Design 5: LOGISIM Tutorial & Demo

7. https://presiuniv.knimbus.com/user#/home

E-content:

1. Z. Xin-Li and W. Hong-Ying, "The Application of Digital Electronics in Networking Communication," 2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2016, pp. 684-687, doi: 10.1109/ICMTMA.2016.168.
2. An encoding technique for design and optimization of combinational logic circuit DipayanBhadra;Tanvir Ahmed Tarique;Sultan Uddin Ahmed;Md. Shahjahan;KazuyukiMurase2010 13th International Conference on Computer and Information Technology (ICCIT)
3. A. Matrosova and V. Provkin, "Applying Incompletely Specified Boolean Functions for Patch Circuit Generation," 2021 IEEE East-West Design & Test Symposium (EWDTS), 2021, pp. 1-4, doi: 10.1109/EWDTS52692.2021.9581029.
4. A. Matrosova, V. Provkin and E. Nikolaeva, "Masking Internal Node Faults and Trojan Circuits in Logical Circuits," *2019 IEEE East-West Design & Test Symposium (EWDTS)*, 2019, pp. 1-4, doi: 10.1109/EWDTS.2019.8884434.

Topics relevant to “SKILL DEVELOPMENT”: Adders, Multiplexers, Decoders / Encoders; Flip-Flops, Counters and Registers for Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

<table>
<tr><td>Course Code:
DES1146</td><td colspan="2">Course Title: Introduction to Design Thinking
Type of Course: Theory</td><td>L-T-P- C</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>Version No.</td><td colspan="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">The course aims to introduce students to the fundamental principles and processes of Design Thinking and will learn to apply Design Thinking methodologies to real-world challenges. The course emphasizes empathy, creativity, and collaboration, equipping students with essential skills for successful engineering practice.</td></tr>
<tr><td>Course Objective</td><td colspan="7">This course is designed to develop and familiarize the learners with the concepts of creating thinking and attain Entrepreneurship by using Participative Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course the students shall be able to:
1) Understand the concept and importance of Design Thinking.
2) Differentiate between traditional problem-solving and Design Thinking.
3) Identify the core stages of the Design Thinking process.</td></tr>
<tr><td>Course Content:</td><td colspan="7">All assignments and projects must be developed using the reference materials available from the PU e-resource database – JSTOR, EBSCO, Library OPAC, NPTEL Videos, etc.</td></tr>
<tr><td>Module 1</td><td>Introduction to Design Thinking</td><td>Visual journal, book of essays, context-specific assignment/project</td><td colspan="3">Visual output generation, by Visual Journal and narrative development.</td><td colspan="2">3 hours</td></tr>
<tr><td colspan="8">Topic
1) Definition and Introduction to Design Thinking
2) Understand the Design Thinking Process</td></tr>
<tr><td>Module 2</td><td>Design Thinking in Action</td><td>Visual journal, book of essays, context-specific assignment/project</td><td colspan="3">Visual output generation, by visual journal and narrative development.</td><td colspan="2">12 hours</td></tr>
<tr><td colspan="8">Topics:
1) Introduction to the steps of Design Thinking Process
2) Understand use cases of Design thinking
3) Design Thinking and Research Tools pertaining to Consumer Tech. , Home Tech. , Personal Tech. , Auto Tech. or Extended Reality.</td></tr>
<tr><td colspan="8">Targeted Application & Tools that can be used:
1) Design ideation tools like Miro , SCAMPER etc.
2) Research Tools for Human Centric Design using forecasting tools like WGSN
3) Feedback tools like Google Forms , etc.
4) Expert Lectures</td></tr>
<tr><td colspan="8">1. Text Book
2. Thinking Design by S Balaram. New Delhi [India]: Sage Publications Pvt. Ltd. 2010. eBook., Database: eBook Collection (EBSCOhost)
3. https://puniversity.informaticsglobal.com:2284/ehost/detail/detail?vid=6&sid=18ab1f43-1f92-4d02-ae2e-a9c06dc06d8c%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=354920&db=nlebk</td></tr>
</table>

References 1. Design Thinking by Clarke, Rachel Ivy. Series: Library Futures, Vol. 4. Chicago: ALA Neal-Schuman. 2020. eBook., Database: eBook Collection (EBSCOhost) 2. https://puniversity.informaticsglobal.com:2282/ehost/detail/detail?vid=4&sid=c80a7d79-eda4-4b7e-a0d6-afafe437962b%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=2433506&db=nlebk 3. The Pocket Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions by Bruce Hanington; Bella Martin. Minneapolis: Rockport Publishers. 2017. eBook., Database: eBook Collection (EBSCOhost) https://puniversity.informaticsglobal.com:2282/ehost/detail/detail?vid=11&sid=f086b8c2-260e-4caa-8c48-d732c21a7724%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=1638693&db=nlebk 4. What Is Design Thinking and Why Is It Important? By Rim Razzouk and Valerie Shute - Review of Educational Research, Vol. 82, No. 3 (September 2012), pp. 330-348 (19 pages), Published by: American Educational Research Association https://puniversity.informaticsglobal.com:2054/stable/23260048?Search=yes&resultItemClick=true&searchText=design+thinking&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3Ddesign%2Bthinking%26so%3Drel&ab_segments=0%2FSYC-6168%2Ftest&refreqid=fastly-default%3Acb1be24976e25734cb5fc13a8af6fdfb&seq=1#metadata_info_tab_contents 5. Abductive Thinking and Sensemaking: The Drivers of Design Synthesis by John Kolko, Design Issues, Vol. 26, No. 1 (Winter, 2010), pp. 15-28 (14 pages), Published by: The MIT Press https://puniversity.informaticsglobal.com:2054/stable/20627839?Search=yes&resultItemClick=true&searchText=design+thinking&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3Ddesign%2Bthinking%26so%3Drel&ab_segments=0%2FSYC-6168%2Ftest&refreqid=fastly-default%3A0b89336ea274d63c010536b01316d7bb&seq=1#metadata_info_tab_contents 6. Designerly Ways of Knowing: Design Discipline versus Design Science by Nigel Cross, Design Issues, Vol. 17, No. 3 (Summer, 2001), pp. 49-55 (7 pages), Published by: The MIT Press https://puniversity.informaticsglobal.com:2054/stable/1511801?Search=yes&resultItemClick=true&searchText=design+thinking&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3Ddesign%2Bthinking%26so%3Drel&ab_segments=0%2FSYC-6168%2Ftest&refreqid=fastly-default%3A0d5b607b163f60876ca973ed90e22b1c&seq=1#metadata_info_tab_contents

Course Code: **PPS 1025**	**Course Title: Industry Readiness Program – I (Audited Course)** **Type of Course: HSMC**	**L- T - P- C**	0	0	2	0
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					

Course Description	This course is designed to enable students to set SMART goals, form Professional & personal ethics for success and learn various email writing techniques. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.
Course Objective	The objective of the course is to **familiarize the learners with the concepts of "Employability for Young Professionals" and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.**
Course Out Comes	**On successful completion of this course the students shall be able to:** CO 1 **Define** their career goals CO 2 **Practice ethical habits** for better career success CO3 **Demonstrate** effective email writing techniques
Course Content	

Module 1	**Goal Setting & Grooming**	Classroom activities	**10 Hours**
Topics: SMART Goals, formal grooming through self-introduction activity **Activity:** Real world scenarios			
Module 2	**Habit Formation**	Role plays	**10 Hours**
Topics: Professional and Personal ethics for success and activity-based practice **Activity:** Students to present 2 min video on building professional ethics			
Module 3	**Email Etiquettes**	Individual and group presentation	**10 Hours**
Topics: Types of prompts to generate effective or desired results for email etiquettes **Activity:** Individual student presenting various search prompts			
Faculty: L&D			
Targeted Application & Tools that can be used: 1. TED Talks 2. You Tube Links 3. Activities			
Assignment proposed for this course Assignment 1: SMART Goal Assignment 2: AI tools for prompt search			
Continuous Individual Assessment Module 1: Presentation Module 2: Activity based assessment Module 3: Class assessment			
The topics related to skill development: Students acquire knowledge on SMART goals, implement grooming standards, practice ethical behavior in class and campus, acquire hands-on experience to use AI tools to get search prompts for desired email etiquettes.			

Course Code: MAT2402	Course Title: **Probability and Statistics** Type of Course: BSC	L-T- P- C	3	0	0	3
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	The course introduces the concepts of probability theory and statistical analysis, covering how to collect, organize, interpret, and draw inferences from data using mathematical models to understand randomness and uncertainty, with applications across various fields like science, engineering, economics, and social sciences.					
Course Objective	The objective of the course is to equip students with the foundational knowledge of probability theory and statistical methods, enabling them to collect, analyze, interpret data, and make informed decisions based on the likelihood of events occurring in various situations, often applied across different fields like science, engineering, and business.					
Course Out Comes	On successful completion of the course the students shall be able to: 1. Be able to compute conditional probabilities directly and using Bayes' theorem, and check for independence of events. 2. Be able to set up and work with discrete & continuous random variables; in particular, to understand the Bernoulli, binomial, geometric, Poisson distributions, uniform, normal, and exponential distributions. 3. Identifying different types of data relationships (linear, polynomial, exponential, logarithmic). 4. Be able to use specific significance tests, including z-test, t-test (one- and two-sample), and chi-squared test					

Course Content:

Module 1	**Basic Probability**		**(6 Classes)**
Probability of an Event, multiplication rule, combinations, permutations, Addition Law, Multiplication Law, Conditional Probability, Bayes's Theorem and Problems.			
Module 2	**Random Variables and Bivariate Distributions**	**Assignment**	**(15 Classes)**
Random Variables (discrete and continuous), Probability Mass/Density Functions, Mathematical Expectations, discrete probability distributions - Binomial distribution, Poisson distribution, geometric distribution, Continuous uniform distribution - exponential distribution, normal distribution, gamma distribution. Bivariate distributions and their properties, distribution of sums and quotients, conditional densities, Bayes' rule.			
Module 3	**Curve Fitting & Statistical Methods**		**(13 Classes)**
Curve Fitting (Straight Line ($y = a + bx$), Parabola ($y = a + bx + cx^2$), Exponential Curves ($y = ae^{bx}$, $y = ab^x$ and $y = ax^b$) Measures of Central tendency, Moments, skewness and Kurtosis, Correlation - Karl Pearson's coefficient of correlation and rank correlation (with & Without repetition, Multiple Correlation - Problems. Regression analysis - lines of regression, Multiple regression - Problems.			
Module 4	**Joint Probability Distribution and Sampling Theory**	**Assignment**	**(15 Classes)**
Joint Probability distribution for two discrete random variables, expectation and covariance. Random sampling, sampling distributions, Standard Error, Type I & Type II errors, Testing of Hypothesis, Test of significance - Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations, Test for single mean, difference of means and correlation coefficients, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes.			

Targeted Application & Tools that can be used:

The contents of this course has direct applications in most of the core engineering courses for problem formulations, Problem Solution and system Design.

Tools Used: R software (Open Source)

Assignment:

1. Select any one simple differential equation pertaining to the respective branch of engineering, identify the dependent and independent variable – Obtain the solution and compare the solution sets by varying the values of the dependent variable.

Text Book

1. Ronald .E. Walpole, Raymond. H. Myers, Sharon. L Myers, and Keying E. Ye, "Probability and Statistics for Engineers and Scientists", Pearson Education, Delhi-9th edition, 2012.
2. B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

References:

1. Miller and Freund, Probability and Statistics for Engineers, Pearson Education Ltd.
2. Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition.
3. Douglas C. Montgomery & George Runger, Applied Statistics and Probability for Engineers, , Wiley Publications

E-resources/ Web links:

1. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_10427
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_100198
3. https://nptel.ac.in/courses/109104124
4. https://nptel.ac.in/courses/111106051
5. https://nptel.ac.in/courses/111102137
6. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
7. https://www.scu.edu.au/study-at-scu/units/math1005/2022/
8. Presidency University's Knimbus library URL is: presiuniv.knimbus.com

Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of Vector calculus and Linear Algebra with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem Solving**.** This is attained through the assessment component mentioned in the course handout.

urse Code: IE7601	vironmental Studies pe of Course: MOOC course	T- P- C				
		ntact hours				
urse Pre-requisites	NIL					
ti-requisites	L					
urse Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamental environmental concepts and their relevance to business operations, preparing them to address forthcoming sustainability challenges. It is designed to equip students with the knowledge and skills needed to make decisions that account for environmental consequences, fostering environmentally sensitive and responsible future managers. **This course is designed to cater to Environment and Sustainability**					
urse jective	The objective of the course is **'SKILL DEVELOPMENT'** of the student by using **'PARTICIPATIVE LEARNING'** techniques					
urse Outcomes	On successful completion of this course the students shall be able to: 1. Describe the issues related to natural resources, ecosystems and biodiversity 2. Identify environmental hazards affecting air, water and soil quality 3. Recognize the importance of healthy environment and finding the sustainable methods to protect the environment 4. Convert skills to address immediate environmental concerns through changes in environmental processes, policies, and decisions					
urse Content:						
dule 1	derstanding Environment, Natural Resources, and Sustainability					

pics:

ssification of natural resources, issues related to Population growth and their overutilization, and strategies for their conservation. Water, air, soil, mineral, energy and food source. Effect of human activities on natural resources.

ncept of sustainability- Sustainable Development Goals (SDGs)- targets and indicators, challenges and strategies for SDGs; Sustainable practices in managing resources, including deforestation, water conservation, Desalination – types, energy security, and food security issues, Life Cycle thinking and Circular Economy.

dule 2	osystems, Biodiversity, and Sustainable Practices			

pics:

Ecosystems and ecosystem services: Various natural ecosystems, Major ecosystem types in India and their basic characteristics; forests, wetlands, grasslands, agriculture, coastal and marine; Ecosystem services- classification and their significance.

The importance of biodiversity, Types of biodiversity, Biodiversity and Climate Change, the threats it faces, hotspots, and the methods used for its conservation. Strategies for in situ and ex situ conservation, mega diverse nation.

dule 3	vironmental Pollution, Waste Management, and Sustainable Development			

pics:

Types of pollution- Chemical, - Biological, Biomedical, noise, air, water, soil, thermal, radioactive and marine pollution, and their impacts on society. Urbanization and Urban environmental problems; effects, and mitigation.

Causes of pollution, such as global climate change, ozone layer depletion, the greenhouse effect, and acid rain, with a particular focus on pollution episodes in India. Importance of adopting cleaner technologies; Solid waste management;

Sustainable Materials and Technologies: Biodegradable and compostable materials, Recycled and reclaimed materials (E-waste management), Sustainable manufacturing processes.

odule 4	**cial Issues, Legislation, and Practical Applications**			

pics:

Overview of key environmental legislation and the judiciary's role in environmental protection, including the Water (Prevention and Control of Pollution) Act of 1974, the Environment (Protection) Act of 1986, and the Air (Prevention and Control of Pollution) Act of 1981. Hazardous waste Rule 1989, Biomedical Waste handling 1998, Fly Ash Rule 1999, Municipal Solid Waste Rule 2000, Battery Rules 2001, E- Waste Rules 2011, Plastic waste management Rules 2016, Construction Demolition waste Rules 2016 National Biodiversity Action Plan (NBAP)

Major International Environmental Agreements: Convention on Biological Diversity (CBD), The Biological Diversity (Amendment) Act, 2023, United Nations Framework Convention on Climate Change (UNFCCC); Kyoto Protocol; Paris Agreement.

Major International organisations and initiatives: United Nations Environment Programme (UNEP), United Nations Educational, Scientific and Cultural Organization (UNESCO), Intergovernmental Panel on Climate Change (IPCC).

rgeted Application & Tools that can be used:

plication areas are Energy, Environment and sustainability

ols: Online Tools – NPTEL and Swayam.

oject work/Assignment:

ssessment Type

- Online exams (MCQs) will be conducted by the department of Chemistry

line Link*:

1) Lecure by Dr. Samik Chowdhury, Dr. Sudha Goel, NPTEL course: Environmental Science, https://nptel.ac.in/courses/109105203, 2024.
2) Lecture by Dr. Padmavati, Dr Narendran Thiruthy, NPTEL Course: Biodiversity Protection, Farmers and Breeders Rights, https://nptel.ac.in/courses/129105008, 2024.

)ther source links are available in below Resources link.

xt Book

1. G. Tyler Miller and Scott Spoolman (2020), Living in the Environment, 20th Edition, Cengage Learning, USA
2. Poonia, M.P. Environmental Studies (3rd ed.), Khanna Book Publishing Co.
3. Bharucha, E. Textbook of Environmental Studies (3rd ed.) Orient Blackswan Private Ltd.
4. Dave, D., & Katewa, S. S. Text Book of Environmental Studies. Cengage Learning India Pvt Ltd.
5. Rajagopalan, R. Environmental studies: from crisis to cure (4th ed.). Oxford University Press.
6. Basu, M., & Xavier Savarimuthu, S. J. Fundamentals of environmental studies. Cambridge University Press.
7. Roy, M. G. Sustainable Development: Environment, Energy and Water Resources. Ane Books.
8. Pritwani, K. Sustainability of business in the context of environmental management. CRC Press.
9. Wright, R.T. & Boorse, D.F. Environmental Science: Toward A Sustainable Future (13th ed,). Pearson.

ference Books

1. Varghese, Anita, Oommen, Meera Anna, Paul, Mridula Mary, Nath, Snehlata (Editors) (2022), Conservation through Sustainable Use: Lessons from India. Routledge.
2. William P. Cunningham and Mary Ann Cunningham (2020), Principles of Environmental Science: Inquiry & Applications, 9th Edition, McGraw-Hill Education, USA.
3. Richard A. Marcantonio, Marc Lame (2022). Environmental Management: Concepts and Practical Skills. Cambridge University Press.
4. Manahan, S.E. (2022). Environmental Chemistry (11th ed.). CRC Press. https://doi.org/10.1201/9781003096238
5. Theodore, M. K. and Theodore, Louis (2021) Introduction to Environmental Management, 2nd Edition. CRC Press

Resources:

1. https://nptel.ac.in/courses/109105203

2. https://archive.nptel.ac.in/courses/120/108/120108004/
3. https://nptel.ac.in/courses/127105018
4. https://onlinecourses.nptel.ac.in/noc23_lw06/preview
5. https://onlinecourses.swayam2.ac.in/ini25_bt02/preview
6. https://archive.nptel.ac.in/courses/120/108/120108002/
7. https://onlinecourses.swayam2.ac.in/ini25_bt02/preview
8. https://nptel.ac.in/courses/102104088
9. https://nptel.ac.in/courses/124107165
10. https://nptel.ac.in/courses/109106200
11. https://archive.nptel.ac.in/content/storage2/courses/120108004/module1/lecture1.pdf
12. https://onlinecourses.swayam2.ac.in/nou25_ge19/preview
13. https://onlinecourses.swayam2.ac.in/ini25_hs01/preview
14. http://kcl.digimat.in/nptel/courses/video/105105184/L32.html
15. https://nptel.ac.in/courses/105105169

Topics relevant to Skill Development:

1. An attitude of enquiry.
2. Write reports

e topics related to Environment and Sustainability :

topics in theory component are relevant to Environment and Sustainability.

urse Code: V1200	urse Title: **Foundations of Integrated Engineering** pe of Course: ESC	- T-P- C	2		0	2
rsion No.						
urse Pre-requisites	**NIL**					
ti-requisites	L					
urse Description	This interdisciplinary course introduces first-year engineering students to foundational principles and practices across key engineering domains, emphasizing real-world problem-solving, sustainability, and ethical innovation. Students explore how civil, mechanical, electrical, and IT systems intersect with emerging technologies like IoT, AI, and geomatics to address global challenges. Through case studies, learners gain deeper understanding of smart infrastructure, prototyping mechanical/electronic systems, and securing IT solutions. Topics include bioinformatics for environmental monitoring, GIS-enabled urban planning, renewable energy integration, and cybersecurity fundamentals. The course cultivates a holistic understanding of engineering's role in sustainable development, safety, and ethical decision-making, preparing students to contribute meaningfully to multidisciplinary projects in a technology-driven world.					
urse Objective	The objective of the course is skill development of student by using Participative Learning techniques.					
urse Outcomes	On successful completion of this course the students shall be able to: 1] Recall key principles of Agile, DevOps, and bioinformatics used in interdisciplinary engineering contexts. 2] Explain the role of GIS, LiDAR, and sustainable materials in designing smart infrastructure and disaster management systems. 3] Describe core components of mechanical systems and their real-world applications. 4] Describe the functionality of IoT-enabled wearable devices, embedded systems, and renewable energy integration in smart grids. 5] List foundational IT concepts such as cloud computing architectures, cybersecurity threats, and blockchain applications.					
urse Content:						

dule 1	**undations of Engineering Practice**	signment	se studies	**essions**
al-world problem-solving using data logic and practical applications, Collaboration and Innovation through multi-domain project, Engineering Ethics & Environmental Impact erging Fields: Automation, and Introduction to bioinformatics and its application stainability & Safety: Circular economy principles, carbon footprint analysis.				
dule 2	**il Engineering & Geomatics**	signment	icle Review	**essions**
art Infrastructure & Geomatics: GIS mapping, LiDAR, drone surveys for urban planning, Geospatial data analysis for disaster management. stainable Construction: 3D-printed structures, self-healing concrete, Digital twins for infrastructure monitoring. een Innovations: Net-zero energy buildings, rainwater harvesting systems.				
dule 3	**chanical Engineering in Action**	signment & Quiz	ta Collection	**essions**
vanced Manufacturing: Collaborative robots (cobots), additive manufacturing and 3D printing, Reverse engineering and prototyping. ergy Systems: Solar/wind energy harvesting, piezoelectric applications. mechanics: Prosthetics design, ergonomic product lifecycle.				
dule 4	**ctrical & Electronics Engineering**	signment & Quiz	ta Collection and visualization	**essions**
art Devices & Systems: Embedded systems, Wearable technology, Edge computing and hardware platforms ergy Innovations: EV charging infrastructure, wireless power transfer, Smart grid integration with renewables.				
dule 5	**ndamentals of IT**	signment & Quiz	se studies	**6 Sessions**

re IT Topics: Networking basics, Cloud computing
bersecurity & Data: Encryption, phishing prevention, zero-trust models, Database management.
ierging Tech: Blockchain for supply chains, AI/ML basics, IoT integration with cloud platforms

rgeted Application & Tools that can be used:

plication Areas include Interdisciplinary problem-solving, Smart city planning, disaster management, Robotics prototyping, renewable energy systems, Wearable health tech, smart grids, Secure cloud systems.
ols: 3D Printers, Autocad, Tinkercad, ArcGIS / QGIS, Arduino/Raspberry Pi

xt Book:

1. William Oakes & Les Leone, "Engineering Your Future: An Introduction to Engineering", Oxford University Press, 9th Edition, 2021
2. Barry F. Kavanagh, "Introduction to Geomatics", Pearson, 5th Edition, 2021
3. Ian Gibson, David Rosen, & Brent Stucker, "Additive Manufacturing Technologies", Springer, 3rd Edition, 2021
4. Sudip Misra, "The Internet of Things: Enabling Technologies, Protocols, and Use Cases", Wiley, 2nd Edition, 2022
5. James Kurose & Keith Ross, "Computer Networking: A Top-Down Approach", Pearson, 8th Edition, 2020

ferences

1. Supratim Choudhuri, "Bioinformatics for Beginners: Genes, Genomes, and Molecular Evolution", Academic Press, 1st Edition, 2023,
2. Robert McGinn, "The Ethical Engineer: Contemporary Concepts and Cases", Princeton University Press, 1st Edition, 2020
3. Charles J. Kibert, "Sustainable Construction: Green Building Design and Delivery", Wiley, 5th Edition, 2022
4. Anthony M. Townsend, "Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia", W.W. Norton & Company, 1st Edition, 2020
5. David Buchla, "Renewable Energy Systems: A Smart Energy Systems Approach", Pearson, 2nd Edition, 2023
6. Charles Platt, "Make: Electronics: Learning Through Discovery", Make Community, 3rd Edition, 2021
7. Charles J. Brooks, Christopher Grow, & Philip Craig, "Cybersecurity Essentials", Wiley, 2nd Edition, 2021

eb-resources:

1. Post-parametric Automation in Design and Construction
https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1155197&site=ehost-live
2. Smart Cities : Introducing Digital Innovation to Cities
https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1993146&site=ehost-live
3. Innovation Energy: Trends and Perspectives or Challenges of Energy Innovation
https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2323766&site=ehost-live
4. Additive Manufacturing: Opportunities, Challenges, Implications
https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1134464&site=ehost-live

urse Code: **IE2501**	**urse Title: Chemistry of Smart Materials** **pe of Course: Theory - BSC**	**L-T- P- C**		0		3
rsion No.	)					
urse Pre-requisites						
ti-requisites	**L**					
urse Description	The objective of the course is to introduce the students to concepts and applications of chemistry of smart materials. The course also aims to enhance the knowledge of smart materials associated with memory system, display devices, , sensors, energy devices and environment. It will also cultivate an ability to identify chemistry in each of smart engineered materials and interpret solutions for the challenges connected to memory, display, energy, smart, green and sustainable technologies. It targets to strengthen the fundamental concepts behind chemistry of smart materials and then builds an interface with their industrial applications. **This course is designed to cater to Environment and Sustainability**					
urse bjective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using Participative learning techniques.					
urse Outcomes	On successful completion of this course the students shall be able to: **1.** Relate the knowledge of chemistry to computational approaches to identify materials functionalities and properties **2.** Recognize and interpret solutions for the challenges connected to memory, display, smart, green and sustainable technologies. **3.** Explain the quality parameters of engineering materials associated with environment and sensors. 4. Interpret the knowledge of sustainable chemistry for E- waste management. 5. Analyse the importance of various electrochemical sources in energy systems.					
urse Content:						
odule 1	**mputational Chemistry**	Assignment	ata Collection and analysis	**09 classes**		

pics: Fundamental particles of atom – their mass, charge and location – atomic number and mass number, Stabilizing interactions: Bonded and non-bonded interactions. Chemistry of weak interactions – van der Waals force and hydrogen bonding, Density functional theory. 3D co-ordinate generation for small molecules, geometry optimization by Molview. Chemical Databases: Chemoinformatics, MSDS

lf- learning topics: Scope, cost and efficiency of computational modeling.

odule 2	**aterials for Memory and Display Systems**	Assignment	ata Collection and analysis	**09 Classes**

Topics: Memory Systems : Introduction, classification of electronic memory devices- Transistor, capacitor, charge -transfer and Resistor, types of materials - organic, polymeric and hybrid materials, and applications, manufacturing of semiconductor chips.

Display Systems: photo and electroactive materials , materials for display -Principle, Properties and applications: Liquid crystals for LCD-Liquid crystals display, Basics of LED: OLED- organic light emitting diode and light emitting electrochemical cells.

Self- learning topics: Green computing: Biocomposite based memory devices

odule 3	**nomaterials based Smart Sensors and Devices**	Assignment	ata Collection and analysis	**09 Classes**

pics: **Nanomaterials**- Introduction, classification based on dimensionality, quantum confinement. Size dependent properties, Synthesis, Properties of CNT and Graphene and their application as **Materials for data analysis and packaging** -RFID and IONT.
Sensors: Introduction, types, Principle and applications- electrochemical sensor: nanomaterials for sensing applications - Glucose, VOC sensing.
Self-learning topics: Fullerene, biomolecules in sensing, Strain sensors

odule 4	**Sustainable Materials and Development**	Quiz/Seminar	ata Collection and analysis	**09 Classes**

pics: **E waste:** Introduction, E waste Hazards, E- waste management, Recovery of precious metal- Cu by Hydrometallurgy.
Green Chemistry: Fundamentals and 12 principles with examples, Carbon footprint and sequestration
Sustainable Chemistry: -Introduction to Biomaterials- PLA , polymers in bio-compatible and bio-degradable materials - Poly(3-hydroxybutyrate-co-3-hydroxyvalerate) PHBV, synthesis and applications in drug delivery.
Self-learning topics: circular economy- case studies.

odule 5	**ergy Science**	Quiz/Seminar	ata Collection and analysis	**Classes**

pics: **Battery technology:**Fundamentals of electrochemistry, Introduction to electrochemical storage devices: battery (Lithium-ion battery- $LiMnO_2$, $LiCoO_2$, metal air batteries- LiO_2) and supercapacitors-Introduction, Principle, Types - EDLC, pseudo and asymmetric capacitor.
Photovoltaics: Solar cells - Construction and working principle; types- Inorganic, Organic and quantum dot sensitized (QDSSC's).
Self -learning topics: Battery technology for e-mobility, Green hydrogen

rgeted Application & Tools that can be used:

plication areas are Data storage and analysis, logistics, Biomedicine, Energy, Environment and sustainability

ols: Molview, chemdraw, excel etc

oject work/Assignment:

ssessment Type

- Midterm exam
- Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screenshot accessing the digital resource.)
- Quiz/Student Seminar
- End Term Exam
- Self-learning

xt Book

10. Wiley, "Engineering Chemistry", Wiley.
11. G.A. Ozin and A.C. Arsenault, Nanochemistry: A chemical approach to nanomaterials, Royal Society of Chemistry, 2009

ference Books

1. Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press,
2. ISBN: 978-036-727-510-5.
3. E-waste recycling and management: present scenarios and environmental issues, Khan, Anish, and
4. Abdullah M. Asiri. 2019, Springer, Vol. 33. ISBN: 978-3-030-14186-8.
5. Essentials of computational chemistry: theories and models, Christopher J Cramer, 2013, John Wiley &
6. Sons. ISBN: 978-0-470-09182-1.
7. Energy storage and conversion devices: Supercapacitors, batteries and hydroelectric cells, Anurag Gaur,
8. L. Sharma, Anil Arya. 2021, CRC press, 1st edition, ISBN: 978-1-003-14176-1.
9. Fundamentals of analytical chemistry: An introduction, Douglas A. Skooget etal., 2004 Thomson

Asia pte
10. Ltd., 8th, ISBN: 978-0-495-55828-6
11. Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press,
12. ISBN: 978-036-727-510-5.
13. Electrical and electronic devices, circuits and materials: Technological challenges and solutions. Tripathi,
14. L., Alvi, P. A., & Subramaniam, U, 2021, John Wiley & Sons, ISBN: 978-0367564261.
15. F. Jensen, Introduction to Computational Chemistry, 3rd edition, Wiley, 2017.

E resources

1. https://presiuniv.knimbus.com/user#/searchresult?searchId=computational%20chemistry&_t=1738054970142
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_48504
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_147967
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_130301
5. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_87297
6. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_67006
7. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_137261
8. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_86712

ill Sets

l topics in theory component are relevant to Environment and Sustainability.

Course Code: CSE2200	**Course Title: Program Solving Using C** **Type of Course: Theory - PCC**			**L- T-P-C**	2	0	0	2
Version No.	1.0							
Course Pre-requisites	**NIL**							
Anti-requisites	**NIL**							
Course Description	The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs and applications in C. ACAlso by learning the basic programming constructs they can easily switch over to any other language in future.							
Course Object	The objective of the course is to familiarize the learners with the concepts of Problem SolvingUsing C and attain Employability through Problem Solving Methodologies.							
Course Outcomes	**On successful completion of this course the students shall be able to:** 1. Write algorithms and to draw flowcharts for solving problems 2. Demonstrate knowledge and develop simple applications in C programmingconstructs 3. Develop and implement applications using arrays and strings 4. Decompose a problem into functions and develop modular reusable code 5. Solve applications in C using structures and Union 6. Design applications using Sequential and Random Access File Processing.							
Course Content:								
Module 1	Introduction to C Language	Quiz	Problem Solving	**6 Sessions**				

Topics: Introduction to Programming – Algorithms – Pseudo Code - Flow Chart – Compilation – Execution – Preprocessor Directives (#define, #include, #undef) - Overview of C – Constants, Variables and Data types – Operators and Expressions – Managing Input and Output Operations – Decision Making and Branching - Decision Making and Looping.				
Module 2	Introduction to Arrays and Strings	Quiz	Problem Solving	**6 Sessions**
Topics: **Arrays:** Introduction – One Dimensional Array – Initialization of One Dimensional Arrays – Example Programs – Sorting (Bubble Sort, Selection Sort) – Searching (Linear Search) - Two Dimensional Arrays – Initialization of Two Dimensional Arrays. Example Programs – Matrix operations. **Strings:** Introduction – Declaring and Initializing String Variables – Reading Strings from Terminal – Writing String to Screen – String Handling Functions.				
Module 3	Functions and Pointers	Quiz	Problem Solving	**6 Sessions**
Topics: **Functions:** Introduction – Need for User-defined functions – Elements of User-Defined Functions: declaration, definition and function call–Categories of Functions – Recursion. **Pointers:** Introduction – Declaring Pointer Variables – Initialization of Variables – Pointer Operators – Pointer Arithmetic – Arrays and Pointers – Parameter Passing: Pass by Value, Pass by Reference.				
Module 4	Structures and Union	Quiz	Problem Solving	**6 Sessions**
Topics: **Structures:** Introduction – Defining a Structure – Declaring Structure Variable – Accessing Structure Members – Array of Structures – Arrays within Structures – **Union:** Introduction – Defining and Declaring Union – Difference Between Union and Structure.				
Module 5	File handling	Case Study	Problem Solving	**6 Sessions**
Topics: Files: Defining and Opening a File – Closing a File – Input / Output Operations on File – Random Access Files				
Text Book(s): **1. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.**				
Reference Book(s): 1. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020. 2. ReemaThareja, "Programming in C", Oxford University Press, Second Edition, 2016. 3. Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015 4. Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014. 5. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.				
Web Links and Video Lectures: **1. https://nptel.ac.in/courses/106/105/106105171/** **2. https://archive.nptel.ac.in/courses/106/104/106104128/**				

<table>
<tr><td>Course Code: CSE2201</td><td>Course Title: Program Solving Using C Lab
Type of Course: Lab - PCC</td><td>L- T-P-C</td><td>0</td><td>0</td><td>4</td><td>2</td></tr>
<tr><td>Version No.</td><td colspan="6">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td colspan="6">The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs and applications in C. ACAlso by learning the basic programming constructs they can easily switch over
to any other language in future.</td></tr>
<tr><td>Course Object</td><td colspan="6">The objective of the course is to familiarize the learners with the concepts of Problem SolvingUsing C and attain Employability through Problem Solving Methodologies.</td></tr>
<tr><td>Course Outcomes</td><td colspan="6">On successful completion of this course the students shall be able to:
1. Write algorithms and to draw flowcharts for solving problems
2. Demonstrate knowledge and develop simple applications in C programmingconstructs
3. Develop and implement applications using arrays and strings
4. Decompose a problem into functions and develop modular reusable code
5. Solve applications in C using structures and Union
6. Design applications using Sequential and Random Access File Processing.</td></tr>
<tr><td>Course Content:</td><td colspan="6"></td></tr>
<tr><td colspan="7">List of Practicals:
Lab Sheet 1: 10 Sessions

Program 1: Sum of Two Numbers
Program 2: Find the Greatest of Three Numbers
Program 3: Check Even or Odd using Conditional Operator
Program 4: Print Multiplication Table using Loop
Program 5: Count Digits in a Number using While Loop
Program 6: Demonstration of Preprocessor Directives
Program 7: Simple Calculator using Switch Case

Lab Sheet 2: 10 Sessions
Program 1: Check Whether a Number is Positive, Negative or Zero
Program 2: Find the Sum of First N Natural Numbers
Program 3: Check Whether a Number is Prime or Not
Program 4: Find Factorial of a Number
Program 5: Reverse a Number
Program 6: Simple Number Guessing Game

Lab Sheet 3: 10 Sessions
Program 1: Linear Search in a One-Dimensional Array
Program 2: Bubble Sort on an Integer Array
Program 3: Matrix Addition (2D Arrays)
Program 4: Count Vowels in a String
Program 6: Selection Sort on an Array</td></tr>
</table>

Lab Sheet 4: 10 Sessions
Program 1: Sum of Two Numbers Using User-Defined Function
Program 2: Factorial Using Recursion
Program 3: Swap Two Numbers Using Call by Value (No Swap)
Program 4: Swap Two Numbers Using Call by Reference (With Swap)
Program 5: Pointer Basics - Access and Modify Variable via Pointer
Program 6: Accessing Array Elements Using Pointers

Lab Sheet 5: 10 Sessions
Program 1: Basic Structure Usage
Program 2: Input and Display Array of Structures
Program 3: Array Inside Structure (Student Marks)
Program 4: Structure with Nested Structures (Date of Birth)
Program 5: Union Example and Member Access

Lab Sheet 6: 10 Sessions
Program 1: Write to a File (Text Mode)
Program 2: Read from a File (Text Mode)
Program 3: Append Data to a File
Program 4: Count Characters, Words and Lines in a File
Program 5: Write and Read Structure to/from a Binary File
Program 6: Random Access in File (Update a Record)

Text Book(s):

1. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.

Reference Book(s):

6. **Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020.**
7. **ReemaThareja, "Programming in C", Oxford University Press, Second Edition, 2016.**
8. **Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015**
9. **Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014.**
10. **Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.**

Web Links and Video Lectures:

1. **https://nptel.ac.in/courses/106/105/106105171/**
2. **https://archive.nptel.ac.in/courses/106/104/106104128/**

IG2501	**lvanced English**	**T- P- C**	2	)	)	2
rsion No.	)					
urse Pre-requisites	IG1900 - English for Technical Communication					
ti-requisites	L					
urse Description	This course is designed to equip students to enhance their communication abilities in Listening, Speaking, Reading, and Writing. The curriculum covers interpersonal communication principles, the art of speech writing and delivery (including impromptu speaking), strategic approaches to critical reading, the identification of logical fallacies, and persuasive writing. Furthermore, the course will introduce students to the potential of AI tools and the techniques of prompt engineering to elevate their communication skills in the digital age. Upon course completion, students will be well-prepared to communicate effectively and critically in both academic and professional environments.					
urse Out Come	On successful completion of the course the students shall be able to: 1. Recognize the elements of interpersonal and cross-cultural communication to address communication challenges effectively. 2. Demonstrate the ability to deliver structured and impromptu speeches using effective speaking techniques. 3. Interpret textual and visual materials using critical reading strategies to evaluate arguments, logic, and persuasion. 4. Produce persuasive and analytical essays using effective argumentation techniques and structured writing strategies.					

Course Content:

odule 1	**undations of Effective Communication**	**se Studies/ Role play**	**oss-Cultural Competency**	**12 Classes**

pics:

- Fundamentals of Interpersonal Communication
- Verbal, Non-verbal, and Paraverbal communication.
- Cultural dimensions theory (Hofstede's Cultural Dimensions).
- Active Listening Techniques
- Common Errors in Communication

Activities:

- Instagram/YouTube Vocabulary Activity
- Charades with a Twist/Tone and Emotion Experiment/Mixed Messages Challenge/Role Reversal Conversations/Observation Exercise

odule 2	**astering Speech Delivery**	**M**	**blic Speaking Confidence**	**12 Classes**

pics:

- Introduction to Prompt Engineering
- Speech Preparation and Organization
- Techniques for Effective Impromptu Speaking
- Practice Speech Delivery

Activities:

- Speech Writing
- Impromptu Speech

odule 3	**itical Reading and Logical Analysis**	**orksheet**	**itical Thinking and Analysis**	**12 Classes**

Topics:

- Critical Reading Strategies: Contextualizing, Figurative Language, Evaluating Logic of an Argument, Recognizing Emotional Manipulation, Analysing Visuals
- Recognizing Logical Fallacies: Slippery Slope, False Dilemma, Post Hoc, Hasty Generalization, Ad Hominem, Straw Man, Bandwagon, No True Scotsman, Red Herring, Appeal to Authority, Sunk Cost, Appeal to ignorance

Activities:

- Critical Reading Worksheet/Identifying Bias in News Articles

odule 4	**riting Effective Arguments**	**signment**	**ear and Coherent Writing**	**Classes**

pics**:**

- Understanding Critical Writing
- Building Arguments (Pathos, Ethos, Logos)
- Techniques for Persuasion

Activities:

- Causes or Effects/Appeal Mash-Up/Debates on Controversial Topics
- Opinion Writing

rgeted Application & Tools that can be used: Quizziz, Chatgpt, Gemini, Youtube, Instagram, Quillbot, Grammarly, Padlet

ferences

1. Adler, R. B., Rodman, G., & DuPré, A. (2019). *Understanding human communication (14th ed.).* Oxford University Press.
2. Moore, B. N., & Parker, R. (2020). *Critical thinking* (13th ed.). McGraw-Hill Education.
3. Hamilton, C. (2020). *Communicating for success* (2nd ed.). Routledge.
4. Ting-Toomey, S., & Dorjee, T. (2018). Intercultural competence: A model for teaching and assessing cross-cultural communication. *Journal of Intercultural Communication*, *47*(2), 213–229. https://doi.org/10.1016/j.jicc.2018.03.004
5. https://www.ted.com/

Topics Relevant to "employability": Teamwork and Collaboration, Critical Thinking and Problem-Solving

Topics Relevant to "Human Values and Professional Ethics": Critical reasoning, Inclusivity and Fairness

urse Code: **E1200**	**urse Title: Basics of Electrical and Electronics Engineering.** **pe of Course: Theory - ESC**	**L-T-P-C**	3	0	0	3
sion No.						
urse Pre-requisites	**NIL**					
ti-requisites						
urse Description	This is a fundamental Course which is designed to know the use of basics of electrical and electronics engineering principles occurs in various fields of Engineering. The course emphasises on the characteristics and applications of electrical and electronic devices. The course also emphasizes on the working, analysis and design of electrical circuits using both active & passive components. Additionally, this course creates a foundation for the future courses such as Electrical machines, power system, power electronics Linear Integrated Circuits, Analog Communication and Digital Communication etc.					
urse Objective	The objective of the course is to familiarize the learners with the concepts of Basics of Electrical and Electronics Engineering and attain **Skill Development** through **Participative Learning** techniques.					
urse Outcomes	**On successful completion of this course the students shall be able to:** 1. **Apply** basic laws of Electrical Engineering to compute voltage, currents and other parameters in the circuits. 2. **Discus**s various fundamental parameters appearing in the characteristics of semiconductor devices and their applications. 3. **Summarize** the operations of different biasing configurations of BJTs and amplifiers. 4. **Discuss** the performance characteristics and applications of various electrical Machines.					
urse Content:						
dule 1	**Introduction to Electrical Circuits**	ignment/ Quiz	merical solving Task	**10 Sessions**		
DC Circuits: Concept of Circuit and Network, Types of elements, Network Reduction Techniques- Series and parallel connections of resistive networks, Star–to-Delta Transformations, Mesh Analysis, Nodal Analysis, Numerical examples. **AC Circuits:** Fundamentals of single phase circuits - Series RL, RC and R-L-C Circuits, Concept of active power, reactive power and Power factor, Numerical examples. Introduction to three phase system and relation between line and phase values in Star & Delta connection, Numerical examples.						
dule 2	**niconductor and Diode applications**	ignment/ Quiz	mory Recall based Quizzes	**Sessions**		
Mass Action Law, Charge densities in a semiconductor, Types of SC, Junction diodes -Ideal and practical behaviour, Modelling the Diode Forward Characteristic, and Diode applications like rectifiers, Zener diode, characteristics and its applications like voltage regulator.						
dule 3	**nsistors and its Applications**	ignment/ Quiz	mory Recall-based Quizzes	**10 Sessions**		
nsistor characteristics, Current components, BJT Configurations (CB, CC, CE configurations) and their current gains. Operating point, Biasing, Fixed Bias, and load line analysis. Single Stage amplifier. JFET (Construction, principal of Operation and Volt –Ampere characteristics). Pinch- off voltage, Comparison of BJT and FET. MOSFET (Construction, principal of Operation and symbol), MOSFET characteristics in Enhancement and Depletion modes.						
dule 4	**Fundamentals of ctrical Machines**	ignment/ Quiz	merical solving Task	**Sessions**		
Electrical Machines: Single phase transformers: principle of operation and EMF equation, Numerical examples. DC Motor: principle of operation, Back EMF, torque equation, Numerical examples. AC Motor: Principle operation of Induction Motors and its Applications.						
f-Learning Topics: pping and clamping circuits, Stabilization Techniques, Voltage divider bias and its stability factor, Multistage amplifier, Darlington pair. **Special Machines:** Introduction to special electrical machines and its applications.						
geted Application & Tools that can be used: **Targeted Applications:** Application Area includes all electrical and electronic circuits (power supply unit, regulator unit, embedded devices, hardware electronics etc.). The students will be able to join a profession which involves basics to high level of electronic circuit design. Professionally Used Software: Multisim/ P Spice ides these software tools hardware equipment such as Multimeters, Function Generators, Power Supplies, Oscilloscopes etc., can be used to perform component/circuit testing and analysis..						

oject Work/ Assignment:

1. Article review: At the end, of course an article topic will be given to an individual or a group of students. They need to refer the library resources and write a report on their understanding about the assigned article in appropriate format.

2. Presentation: There will be a group presentation, where the students will be given a topic. They will have to explain/demonstrate the working and discuss the applications for the same.

3. Case Study: - At the end of the course students will be given a 'real-world' application based circuits like Power Amplifier, Signal/Function Generator etc. as a case study. Students will be submitting a report which will include Circuit Diagrams, Design, Working Mechanism and Results etc. in appropriate format

xt Book(s):

1. Kothari D. P. & Nagrath I. J., "Basic Electrical and Electronics Engineering", Tata McGraw-Hill
2. Education
3. Theraja B.L. and Theraja A.K., "A Textbook of Electrical Technology: Basic Electrical Engineering" in S.I. System of Units, 23rd ed., New Delhi: S. Chand, 2002.
4. A.P.Malvino, Electronic Principles,7thEdition, Tata McGraw Hill,2007
5. J. Millman, C. C. Halkias and C. D. Parikh, "Millman's Integrated Electronics", McGraw Hill Education, 2nd Edition.
6. Basics of Electrical & Electronics Laboratory Manual.

ference Book (s):

1. John Hiley, Keith Brown and Ian McKenzie Smith, "HUGHES Electrical and Electronic Technology", 10th Edition (Indian Edition published by Dorling Kindersley), Pearson,2011
2. Samarajit Ghosh, "Fundamentals of Electrical and Electronics Engineering", 2nd Edition, Prentice Hall India, 2007.
3. K Uma Rao, A Jaya Lakshmi, "Basic Electrical engineering" IK International publishing house Pvt. Ltd
4. R. L. Boylestad and L. Nashelsky, "Electronic Devices and Circuit Theory", Pearson Education India 7th Edition.
5. A K. Maini, V. Agrawal, "Electronic Devices & Circuits", Wiley, 2nd Edition
6. A.S Sedra, K. C. Smith, "Microelectronic Circuits", Oxford University Press, 6th Edition

Online Resources (e-books, notes, ppts, video lectures etc.):

1. https://presidencyuniversity.linways.com
2. https://www.digimat.in/nptel/courses/video/108105112/L01 "Fundamentals of Electrical Engineering-Basic Concepts, Examples"
3. Seminar Topic: https://nptel.ac.in/courses/108/105/108105153/ "Electrical Measurements"
4. Video lectures on "Electronic Devices" by Prof. Dr. A. N. Chandorkar, IIT Bombay http://www.satishkashyap.com/2013/03/video-lectures-on-electron-devices-by.html
5. Video lectures on "Analog Electronics" by Prof. S.C. Dutta Roy, IIT Delhi https://nptel.ac.in/courses/108/102/108102095/
6. Video lectures on "Diodes", by Prof. Chitralekha Mahanta, IIT Guwahati,

https://nptel.ac.in/courses/117/103/117103063/

E-content:

1. "Introduction to Electrical Machines https://nptel.ac.in/courses/108/102/108102146/"

M. -Y. Kao, H. Kam and C. Hu, "Deep-Learning-Assisted Physics-Driven MOSFET Current Voltage Modeling," in IEEE Electron Device Letters, vol. 43, no. 6, pp. 974-977, June 2022, doi: 10.1109/LED.2022.3168243

https://ieeexplore-ieee-org-resiuniv.knimbus.com/document/9758727

2. F. Bonet, O. Aviñó-Salvadó, M. Vellvehi, X. Jordà, P. Godignon and X. Perpiñà, "Carrier

Concentration Analysis in 1.2 kV SiC Schottky Diodes Under Current Crowding," in IEEE Electron Device Letters, vol. 43, no. 6, pp. 938-941, June 2022, doi: 10.1109/LED.2022.3171112. https://ieeexplore-ieeeorg-presiuniv.knimbus.com/document/9764749

3. M. Chanda, S. Jain, S. De and C. K. Sarkar, "Implementation of Subthreshold Adiabatic Logic for Ultralow-Power Application," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. .23, no. 12, pp. 2782-2790, Dec. 2015.

https://ieeexplore.ieee.org/document/7018053

4. R. Raut and O. Ghasemi, "A power efficient wide band trans-impedance amplifier in submicron

CMOS integrated circuit technology," 2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference, 2008, pp. 113-116, doi: 0.1109/NEWCAS.2008.4606334. https://ieeexplore.ieee.org/document/4606334

Topics relevant to "SKILL DEVELOPMENT": Performing suitable experiments to compute the electric circuit parameters, performance operation of machines, and operation of semiconductor devices for **Skill Development** through **Participative Learning techniques.** This is attained through assessment component mentioned in course plan.

urse Code EEE1250	urse Title: Basics of Electrical and Electronics Engineering Laboratory pe of Course: Laboratory - ESC	L-T- P- C	0	0	2	1
rsion No.						
urse Pre-requisites	L					
ti-requisites	L					
urse Description	This fundamental laboratory provides an opportunity to validate the concepts taught in the basics of electrical and electronics engineering and enhances the ability to visualize real system performance, using both hardware and simulation tools.					
urse Objective	The objective of the course is to familiarize the learners with the concepts of Basics of Electrical and Electronics Engineering and attain Skill Development through Experiential Learning techniques.					
sic skill sets required for the laboratory:						
	The students shall be able to develop: **1) An attitude of enquiry.** **2) Confidence and ability to tackle new problems.** **3) Ability to interpret events and results.** **4) Ability to work as a leader and as a member of team.** 5) Assess errors and eliminate them. **6) Observe and measure physical phenomenon.** **7) Write Reports.** **8) Select suitable equipment, instrument and materials.** 9) Locate faults in systems. 10) Manipulative skills for setting and handling equipment. **11) The ability to follow standard test procedures.** **12) An awareness of the need to observe safety precautions.** 13) To judge magnitudes without actual measurement.					
urse Out Comes	**On successful completion of the course the students shall be able to:** 1. Apply basic laws of Electrical Engineering to compute voltage, currents, and other parameters in the circuits. 2. Demonstrate the working of electrical machines to observe performance characteristics. 3. Demonstrate the working of electronic circuits to obtain the V-I Characteristics of various semiconductor devices. 4. Sketch the characteristics and waveforms relevant to standard electrical and electronic circuits					
urse Content:						

t of Laboratory Tasks:

Experiment No 1: Verification of KVL and KCL for a given DC circuit.
Level 1: Study and Verify KVL and KCL for the given electrical Circuit.
Level 2: For the same circuit considered in level 1, perform the simulation using NI LabVIEW/Multisim/MATLAB.
Experiment No 2: Analyse AC series circuits – RL, RC and RLC .
Level 1: Conduct an experiment to perform and verify the impedance, current and power of Series RL and RC circuits
Level 2: Conduct an experiment to perform and verify the impedance and current of RLC series circuits.
Experiment No 3: Calculation of power and power factor of the given AC Circuit.
Level 1: Conduct an experiment to measure the power and power factor for given resistive load.
Level 2: Conduct an experiment to measure the power and power factor for given inductive load.
Experiment No 4: Perform the experiments on given Transformer.
Level 1: Verify the EMF equation of a transformer and compute the voltage transformation ratio.
Level 2: Study the effect of load on the secondary side of the transformer and verify the EMF equation under load conditions.
Experiment No 5: Load test on DC shunt motor
Level 1: Conduct load test on DC shunt motor and find its efficiency at different loads
Level 2: Conduct load test on DC shunt motor and plot the performance characteristics.

periment 6: Study of PN-Junction Diode Characteristics in Forward and Reverse Bias Conditions.

vel 1: Carry out an experiment to plot VI Characteristics and hence find the cut-in voltage on forward characteristics for the Silicon P-N Junction diode.

vel 2: Carry out an experiment to plot VI Characteristics of Zener diode and hence find the zener voltage on reverse characteristics for the Silicon P-N Junction zener diode.

periment No. 7: To observe the output waveform of half wave and full wave rectifier circuit and compute ripple factor and efficiency

vel 1: Identify the components required for a rectifier circuit, rig up the circuit, and sketch the output waveforms without filter.

vel 2: Rig up the rectifier circuit with RC filter, observe the output waveforms, determine the efficiency and ripple factor.

periment 8: To construct clipping and clamping circuits for different reference voltages and to verify the responses.

vel 1: Identify the components required for building a Clipper / Clamper circuit. Rig up the circuit according to the circuit diagram given and sketch the output waveform.

vel 2: Given a sinusoidal input of 10 V p-p, implement a positive / negative clipper with output clipped at 2 V.

periment 9: To calculate various parameters of emitter follower circuit using BJT

vel 1: Identify the components required to implement an emitter follower circuit. Rig up the circuit and observe the variations in output waveform with respect to the variations in input waveform.

vel 2: Determine the values of Z_{in} input impedance and Z_{out} output impedance for Emitter Follower.

periment 10: To Implement RC Coupled amplifier using a BJT and sketch the frequency response.

vel 1: Identify the components required to implement an RC coupled amplifier circuit. Rig up the circuit and sketch the frequency response.

Level 2: From the frequency response curve determine the value of the mid band gain and the bandwidth.

rgeted Application & Tools that can be used:

Targeted Applications: Application Area includes all electrical and electronic circuits (power supply unit, regulator unit, embedded devices, hardware electronics etc.). The students will be able to join a profession which involves basics to high level of electronic circuit design.

Professionally Used Software: Multisim/ P Spice

sides these software tools hardware equipment such as Multimeters, Function Generators, Power Supplies, Oscilloscopes etc., can be used to perform component/circuit testing and analysis.

urse Material

1. Basics of Electrical and Electronics Engineering Laboratory Manual, Presidency University, Bengaluru.

xt Book:

2. Kothari D. P. & Nagrath I. J., “Basic Electrical and Electronics Engineering”, Tata McGraw-Hill

Reference Books:

3. John Hiley, Keith Brown and Ian McKenzie Smith, "HUGHES Electrical and Electronic Technology", 10th Edition (Indian Edition published by Dorling Kindersley), Pearson,2011
4. Samarajit Ghosh, "Fundamentals of Electrical and Electronics Engineering", 2nd Edition, Prentice Hall India, 2007.
5. K Uma Rao, A Jaya Lakshmi, “Basic Electrical engineering” IK International publishing house Pvt. Ltd
6. R. L. Boylestad and L. Nashelsky, “Electronic Devices and Circuit Theory”, Pearson Education India 7th Edition.
7. A K. Maini, V. Agrawal, “Electronic Devices & Circuits”, Wiley, 2nd Edition
8. A.S Sedra, K. C. Smith, “Microelectronic Circuits”, Oxford University Press, 6th Edition

Online Learning Resources:

9. https://presidencyuniversity.linways.com
10. https://www.digimat.in/nptel/courses/video/108105112/L01 “Fundamentals of Electrical Engineering-Basic Concepts, Examples”
11. Video lectures on “Diodes”, by Prof. Chitralekha Mahanta, IIT Guwahati, https://nptel.ac.in/courses/117/103/117103063/

Topics relevant to “SKILL DEVELOPMENT”: All the experiments which are listed are for **Skill Development** through **Experiential Learning Techniques**. This is attained through the assessment component mentioned in course handout.

urse Code: **.W7601**	**lian Constitution** **pe of Course: MOOC course**	**T- P- C**				
		ntact hours				
urse Pre-requisites	NIL					
ti-requisites	L					
urse Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamentals of Indian Constitution concepts and their relevance to 75+ Years of Republic of India (https://constitution75.com/) as well as **#AzaadiKaAmrutMahotsav / Azadi Ka Amrit Mahotsav** (https://amritmahotsav.nic.in). It is designed to equip students with the knowledge about the Constitution of India. This course aims to introduce the constitutional law of India to students from all walks of life and help them understand the constitutional principles as applied and understood in everyday life. The objective of making the Constitution of India, familiar to all students, and not only to law students, this course aims and objectifies legal understanding in the simplest of forms. **This course is designed to cater to Constitutional Studies.**					
urse jective	The objective of the course is **'SKILL DEVELOPMENT'** of the student by using **'PARTICIPATIVE LEARNING'** techniques					
urse Outcomes	On successful completion of this course the students shall be able to: 1. Describe the basic understanding of the Indian Constitution and the concepts and issues relevant to day-to-day life of the nation and to equip the Citizen with the zeal of capacity building. Recognizing and identify the values of the Constitution of India. 2. Enabling the Citizen-centric Awareness of Rights and Responsibilities of the State 3. Explain the role of the State actors in building India. 4. Understanding the Gandhian vision over the power of the LSG (Local Self-Governance)					
urse Content:						
dule 1	**derstanding the Making of the Constitution: The Constituent Assembly & The Constitution of India**					
pics: Historical Context of Constituent Assembly - Compositions & Functions of Constituent Assembly What is a Constitution? – Why have a Constitution? – Constitutional Change - Features of Indian Constitution – Preamble of Indian Constitution						
dule 2	**izen's Fundamental Rights and State's Responsibilities (Directive Principles)**					
pics: Introduction to Fundamental Rights - Right to Equality – Facets of Right to Equality - Right to Freedom - Constitutional Position of Some Democratic Rights - Right Against Exploitation - Right to Freedom of Religion - Right to Constitutional Remedies Directive Principles of the State Policy						
dule 3	**gans Of the Government**					
pics: **Executive**: The President of India - Powers and Functions of President of India - Emergency Powers and the Position of the President **Legislature**: Union Council of Ministers - Prime Minister - The Rajya Sabha - The Lok Sabha - Relation between the Lok Sabha & Rajya Sabha - Office of the Speaker – Important Parliamentary Committees **Judiciary**: The Structure and Organization of the Judiciary & the High Court - The Supreme Court - Role of The Supreme Court - Judicial Activism in India - Basic Structure Doctrine & PIL						
dule 4	**deralism & Decentralization**					
pics:						

What is Federalism? - Centre-State Legislative Relations - Centre-State Administrative Relations - Centre-State Financial Relations
The 5th & 6th Schedules - Municipality- (History of Indian Municipality, Organization & Functions) – Panchayat 1 (Idea of Panchayat, Organization and Powers of Panchayats in India)

rgeted Application & Tools that can be used:

plication areas to familiarize students with fundamentals of Indian Constitutional concepts.

ols: Online Tools – NPTEL and Swayam.

oject work/Assignment:

ssessment Type

- Online end term exam will be conducted as notified by the Presidency University.

line Link*:

1) Prof. Amitabha Ray, SWAYAM Course: "Constitutional Government & Democracy in India"
https://onlinecourses.swayam2.ac.in/cec19_hs13/preview

* Other source links are available in below Resources link.

xt Book

1. Durga Das Basu --- Introduction to the Constitution of India, 23rd Edition (Gurgaon; LexisNexis, 2018).
2. MP Jain's Constitutional Law of India, Lexis Nexis
3. V.N Shukla's Indian Constitutional Law, M.P Singh 13th Edition
4. MV Pylee's Constitution of India
5. J.C.Johari -- The Constitution of India: A Politico-Legal Study (Greater Noida: Sterling Publishers Pvt. Ltd. 2013).
6. Himangshu Roy and M.P.Singh – Indian Political System, 4th Edition (Bengaluru; Pearson Education, 2018)
7. Vidya Bhushan & Vishnoo Bhagwan--- Indian Administration (S. Chand, 2011)
8. S.R.Maheswari --- Indian Administration (Orient Blackswan, 2001)
9. Dr. A.Avasthi & A.P. Avasthi --- Indian Administration (L.N. Agarwal Educational Publishing, 2017).
10. B. L. Fadia --- Indian Government and Politics (Sahitya a. Bhawan, 13th Revised Edition, 2017).
11. P.M.Bakshi – The Constitution of India (Prayagraj, UP; a. Universal Law Publishing, January, 2018)

ference Books

1. HM Seervai, Constitutional Law of India, 4th Ed. Vol I, II, & III
2. Uday Raj Rai, Constitutional Law-I
3. Democracy and Constitutionalism in India, Oxford University Press 2009

Resources:

1. https://onlinecourses.nptel.ac.in/noc20_lw03/course?&force_user=true
2. https://onlinecourses.swayam2.ac.in/cec19_hs13/course?&force_user=true
3. https://nptel.ac.in/courses/129106003
4. https://nptel.ac.in/courses/129106411
5. https://nptel.ac.in/courses/129105608
6. https://nptel.ac.in/courses/129106002

Topics relevant to Skill Development:

1. An attitude of inquiry.
2. Write reports

e topics related to Constitutional Studies and its application :

topics in theory component are relevant to Indian Constitution.

<table>
<tr><td>Course Code:
PPS1026</td><td colspan="2">Course Title: Industry Readiness Program – II (Audited Course)
Type of Course: Practical Only Course</td><td>L- T - P- C</td><td>0</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>Version No.</td><td colspan="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7"></td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">This course is designed to enable students learn styles of communication, team building and use empathy in leadership. The course will benefit learners in preparing themselves effectively through various activities and learning methodologies.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is to familiarize the learners with the concepts of “Industry Readiness for Young Professionals” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.</td></tr>
<tr><td>Course Out Comes</td><td colspan="7">On successful completion of this course the students shall be able to:
CO 1 Apply different communication skills for success in workplace
CO 2 Practice team building skills for career success
CO3 Demonstrate ethical leadership skills in workplace</td></tr>
<tr><td>Course Content</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Effective Communication</td><td>Classroom activities</td><td colspan="5">10 Hours</td></tr>
<tr><td colspan="8">Topics: Practice effective communication skills (Verbal, Non-verbal, Written and Visual)
Activity: Use social media prompts to prepare self-introduction videos</td></tr>
<tr><td>Module 2</td><td>Team Building</td><td>Group Activity</td><td colspan="5">10 Hours</td></tr>
<tr><td colspan="8">Topics: Skills of an effective team player
Activity: Student group activity to build class networking</td></tr>
<tr><td>Module 3</td><td>Leadership</td><td>Case study</td><td colspan="5">10 Hours</td></tr>
<tr><td colspan="8">Topics: Types of leadership, using empathy in leadership
Activity: Individual presentation by students on corporate leaders.</td></tr>
<tr><td colspan="8">Faculty : L&D</td></tr>
<tr><td colspan="8">Targeted Application & Tools that can be used:
1. TED Talks
2. You Tube Links
Activities</td></tr>
<tr><td colspan="8">Assignment proposed for this course
Assignment 1: One minute reel
Assignment 2: Team building assignment</td></tr>
<tr><td colspan="8">Continuous Individual Assessment
Module 1: L-S-R-W class assessment
Module 2: Team Presentation
Module 3: Individual Assessment</td></tr>
<tr><td colspan="8">The topics related to skill development:
Students acquire knowledge on effective communication skills, team building skills and how to prepare themselves to be leaders in workplace using empathy and implement various skill sets during the course of their time in the university.</td></tr>
</table>

<table>
<tr><td>urse Code:
ECE1511</td><td>urse Title: Design Workshop
urse Type : ESC</td><td>T-P- C</td><td>1</td><td>0</td><td>2</td><td>2</td></tr>
<tr><td>rsion No.</td><td colspan="6"></td></tr>
<tr><td>urse Pre-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>ti-requisites</td><td colspan="6">L</td></tr>
<tr><td>urse Description</td><td colspan="6">This course is designed to provide an in-depth understanding of Arduino, microcontrollers Raspberry pi and their application in various real time projects involving sensors. Throughout the course, students will learn the fundamentals of Arduino and Raspberry Pi programming and gain hands-on experience with a wide range of sensors. Students will explore how to connect and interface sensors with Arduino and Raspberry Pi boards, read sensor data, and use it to control various output devices This course is suitable for beginners who are interested in exploring the world of electronics and developing practical applications using Arduino, Raspberry Pi and sensors.</td></tr>
<tr><td>urse Objective</td><td colspan="6">The objective of the course is Employability Skills of student by using PARTICIPATIVE LEARNING techniques.</td></tr>
<tr><td>urse Outcomes</td><td colspan="6">On successful completion of the course the students shall be able to
1. Explain the main features of the Arduino & the Raspberry Pi prototype board.
2. Demonstrate the hardware interfacing of the peripherals to Arduino and Raspberry Pi system.
3. Understand the types of sensors and its functions
4. Demonstrate the functioning of live projects carried out using Arduino and Raspberry Pi system.</td></tr>
<tr><td>urse Content:</td><td colspan="6"></td></tr>
<tr><td>dule 1</td><td>Basic concepts of Microcontrollers</td><td>Hands-on</td><td>Interfacing Task and Analysis</td><td colspan="3">3 Sessions</td></tr>
<tr><td colspan="7">pics:
Introduction to Arduino, ESP and Node MCU Pin configuration and architecture, Device and platform features, Concept of digital and analog ports, Familiarizing with Arduino Interfacing Board, API's , Introduction to Embedded C and Arduino platform, Arduino Datatypes and variables, Arduino i/o Functions, Arduino Communications, Arduino IDE, Various Cloud Platforms.</td></tr>
<tr><td>dule 2</td><td>nsory Devices</td><td>nds-on</td><td>erfacing Task and Analysis</td><td colspan="3">essions</td></tr>
<tr><td colspan="7">duino Sensors: Humidity Sensor, Temperature Sensor, Water Detector / Sensor, PIR Sensor, Ultrasonic Sensor, Connecting Switches and actuators, sensor interface with Arduino.
roduction to 3D Printer: 3D Printer technology and its working Principles, Applications. Introduction to online Simulators: Working with AutoCAD/Fusion 360 Simulator.</td></tr>
<tr><td>dule 3</td><td>roduction to Micro python</td><td>nds-on</td><td>erfacing Task and Analysis</td><td colspan="3">Sessions</td></tr>
<tr><td colspan="7">pics:
roduction to MicroPython, Comparison with other programming languages, Setting up the MicroPython development environment, Basics of MicroPython syntax and structure.</td></tr>
<tr><td>dule 4</td><td>orking with Raspberry-pi</td><td>nds-on</td><td>erfacing Task and Analysis</td><td colspan="3">essions</td></tr>
<tr><td colspan="7">roduction to raspberry pi boards, pin-diagram, different types of raspberry pi boards and its application, LED and switch control. Mastering Modules, Setup Raspberry - PuTTY SSH,VNC Viewer to interface with more complicated sensors and actuators. Various Libraries and its functions.</td></tr>
<tr><td colspan="7">b: Name of the Experiments:
1. Introduction Lab 1:
Level 1: Overview on Arduino based Micro-controller, and sensors.</td></tr>
</table>

Level 2: Interfacing of Arduino and ESP boards with sensors and other components.

2. **Lab 2: Smart Plant Monitoring**

Level 1- Push button-controlled LED.

Level 2- Automatic Irrigation and monitoring System using Arduino

3. **Lab 3: Robotics with Arduino**.

Level 1- Servo Motor control using Arduino

Level 2: DC Motor Control Using Arduino for Robotics.

4. **Lab 4: Environmental pollution using ESP.**

Level 1 - IoT based air Pollution Monitoring System.

Level 2- IoT Based water pollution system

5. **Introduction Lab for raspberry pi:**

Level 1: Overview on Different Raspberry Pi Boards, and sensors.

Level 2: Configuring the Raspberry Pi and Interfacing with sensors and other components.

6. **Lab 7:** Raspberry Pi based Object Detection using TensorFlow and OpenCV.
7. **Lab 8:** Speech Recognition on Raspberry Pi for Voice Controlled Home Automation.
8. **Lab 9:** Design the website using HTML and CSS, and host the website on Raspberry Pi.
9. **Introduction Lab for 3D printing:**

Overview of 3D printing. Design of 3D structure using the CAD. Understand the steps of fabrication of simple rectangular box using 3D printer.

10. **Lab 10:** Design and print of Hollow Cylindrical structure using 3D CAD and 3D printer.
11. **Lab 11** Demonstration of Jetson nano board and its capability. **(OPTIONAL)**
12. **Lab 12:** Revision
13. **Lab 13:** Revision
14. **Lab 14:** Mini Project
15. **Lab 15:** Mini Project Evaluation.

Topics: Types of Arduino boards, Thonny Python, Python IDLE, **sensors, 3D Printer**

rgeted Application & Tools that can be used:

plication Area:

me Automation, Environmental Monitoring, Agriculture and Farming, Industrial Automation, Internet of Things (IoT), Robotics, Wearable Devices, Security Systems, Education and Learning. These are just a few examples of the many application areas where Arduino, Raspberry Pi and sensors can be applied. The flexibility and affordability of Arduino, and Raspberry Pi combined with the wide range of sensors available, allow for endless possibilities in creating innovative projects.

ofessionally Used Software: Students can use open SOURCE Software's Arduino IDE and Tincker CAD, Thonny Python, Python IDLE etc.

oject work/Assignment:

1. Projects: At the end of the course students will be completing the project work on solving many real time issues.

2. Book/Article review: At the end of each module a book reference or an article topic will be given to an individual or a group of students. They need to refer the library resources and write a report on their understanding about the assigned article in appropriate format. Presidency University Library Link .

Presentation: There will be a presentation from interdisciplinary students group, where the students will be given a project on they have to demonstrate the working and discuss the applications for the same

xtbook(s):

1. **Monk Simon "Programming Arduino: Getting Started with Sketches", Mc Graw Hill Publications Second Edition**
2. **Monk Simon "Raspberry Pi Cookbook: Software and Hardware Problems and Solutions", Publisher(s): O'Reilly Media, Inc. ISBN: 9781098130923 fourth Edition.**

ferences

ference Book(s)

1. Neerparaj Rai "Arduino Projects for Engineers" BPB publishers,first edition, 2016.
2. Ryan Turner "Arduino Programming " Nelly B.L. International Consulting Ltd. first edition,2019.
3. Charles Bell Micro Python for the Internet of Things: A Beginner's Guide to Programming with Python on Microcontrollers" by" Edition 1, 2017, ISBN 978-1-4842-3123-4
4. Stewart Watkiss "Learn Electronics with Raspberry Pi " Apress Berkeley, CA . second edition,2020. ISBN978-1-4842-6348-8
5. Jo Prusa, "Basic of 3D printing", Prusa Research, 3rd edition.
6. Volker Ziemann, "**A Hands-On Course in Sensors Using the Arduino and Raspberry Pi (Series in Sensors)**", CRC Press, 1st Edition. 2018.

line Resources (e-books, notes, ppts, video lectures etc.):

1. Arduino trending Projects < https://www. https://projecthub.arduino.cc/>
2. Introduction to Arduino < https://onlinecourses.swayam2.ac.in/aic20_sp04/preview>
3. Case studies on Wearable technology< https://www.hticiitm.org/wearables>
4. Raspberry-pi Projects < https://magpi.raspberrypi.com/articles/category/tutorials/>
5. Introduction to internet of things< https://nptel.ac.in/courses/106105166>

ontent:

1. Cattle Health Monitoring System Using Arduino and IOT (April 2021| IJIRT | Volume 7 Issue 11 | ISSN: 2349-6002)
2. M H Hemanth Kumar, Ravi Pratap Singh, Nishu Sharma, Pragya Singh" IOT BASED SMART SECURITY SYSTEM USING ARDUINO" 2021 JETIR August 2021, Volume 8, Issue 8.
3. R. Maheswar, P. Jayarajan, S. Vimalraj, G. Sivagnanam, V. Sivasankaran and I. S. Amiri, "Energy Efficient Real Time Environmental Monitoring System Using Buffer Management Protocol," 2018, pp. 1-5, doi: 10.1109/ICCCNT.2018.8494144. **https://ieeexplore.ieee.org/document/8494144**.
4. Yaser S Shaheen,Hussam., " Arduino Mega Based Smart Traffic Control System ," December 2021 Asian Journal of Advanced Research and Reports 15(12): 43-52, 2021(15(12): 43-52, 2021):15(12): 43-52, 2021.
5. Basil, Eliza Sawant, S.D. "IoT based traffic light control system using Raspberry Pi " DOI 10.1109/ICECDS.2017.8389604
6. Supriya S, 2Dr. Aravinda " Green leaf disease detection and identification using Raspberry Pi https://www.irjet.net/archives/V9/i8/IRJET-V9I847.
7. Dr. E.N. Ganesh., "Health Monitoring System using Raspberry Pi and IOT" DOI : http://dx.doi.org/10.13005/ojcst12.01.03

pics relevant to development of "SKILL": System design for achieving Sustainable Development Goals.

Course Code: MAT2303	**Course Title:** Linear Algebra & Vector Calculus **Type of Course: BSC**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	This course explores the fundamental concepts of vectors, matrices, and their operations within the context of calculus, including vector differentiation and integration, while applying these tools to solve problems related to linear systems, transformations, and geometric interpretations in higher dimensions, often with applications in fields like physics, engineering, and computer graphics; key topics include vector algebra, matrix operations, determinants, eigenvalues, eigenvectors, gradients, divergence, curl, line integrals, surface integrals, and the fundamental theorems of vector calculus like Green's Theorem, Stokes' Theorem, and the Divergence Theorem.					
Course Objective	The course is intended to develop computational proficiency involving procedures in Matrices, Linear Algebra and Vector Calculus which are useful to all engineering disciplines. This course is to equip students with the ability to understand and manipulate vectors in multidimensional space, apply matrix operations to solve systems of linear equations, and utilize concepts like gradients, divergence, and curl to analyze physical phenomena, all while developing a strong foundation for applying these tools in various scientific and engineering fields like physics, mechanics, and computer graphics.					
Course Out Comes	On successful completion of the course the students shall be able to: CO1 - Use matrix methods and certain techniques to solve the system of linear equations and to find eigen values, eigen vectors of a matrix to check whether it is diagonalizable. CO2 - Understand the abstract notions of vector space and dimensionality of it. CO3 - find the matrix representation of a linear transformation given bases of the relevant vector spaces. CO4 - Learn different notions of vector and scalar fields with their properties. Understanding the major theorems (Green's, Stokes', Gauss') and some applications of these theorems.					
Course Content:						

Module 1	**Systems of Linear Equations**		***6 Classes***
Systems of Linear Equations, Matrices and Elementary Row Operations, Echelon forms, Matrix operations, invertible matrices, Determinants and their properties, Cramer’s Rule, LU-decomposition, Applications of Systems of Linear Equations.			
Module 2	**Vector Space**	**Assignment**	***7 Classes***
Linear Combinations and Linear Independence, Vectors in n R^n, Linear Combinations, Linear Independence Vector Spaces, Definition of a Vector Space, Subspaces, Basis and Dimension, Coordinates and Change of Basis, Orthogonal bases and orthogonal projections.			
Module 3	**Linear Transformations**		***15 lectures***
Linear Transformations, Algebra of transformations, The Null Space and Range, Isomorphisms, Matrix Representation of Linear Transformations, Similarity Eigenvalues and Eigenvectors, Eigen values and Eigen vectors, Diagonalization. Inner Product Spaces, The Dot Product on R^n and Inner Product Spaces, Orthonormal Bases, Orthogonal Complements, Application: Least Squares Approximation, Diagonalization of Symmetric Matrices, Application: Quadratic Forms. Singular Value Decomposition: Singular values, computing singular value decomposition, and Introduction to principal component analysis.			
Module 4	**Vector Calculus**	**Assignment**	***(15 lectures)***
Vector & Scalar Functions and Fields, Derivatives, Curve, Arc length, Curvature & Torsion, Gradient of Scalar Field, Directional Derivative, Divergence of a Vector Field, Curl of a Vector Field, Physical interpretation, solenoidal and irrotational vector fields. Problems.			

Line Integrals, Path Independence of Line Integrals, Green`s Theorem in the plane, Surface Integrals, Divergence Theorem of Gauss, Stokes`s Theorem.

Targeted Application & Tools that can be used:

1. Solve systems of linear equations using various methods including Gaussian and Gauss Jordan elimination and inverse matrices.
2. Perform matrix algebra, invertibility, and the transpose and understand vector algebra in R^n.
3. Determine relationship between coefficient matrix invertibility and solutions to a system of linear equations and the inverse matrices.
4. Find eigenvalues and eigenvectors and use them in applications.
5. Find the dimension of spaces such as those associated with matrices and linear transformations.
6. Understand real vector spaces and subspaces and apply their properties.
7. Compute inner products in a real vector space and compute angle and orthogonality in inner product spaces.
8. Create orthogonal and orthonormal bases: Gram-Schmidt process and use bases and orthonormal bases to solve application problems.
9. Prove basic results in linear algebra using appropriate proof-writing techniques such as linear independence of vectors; properties of subspaces; linearity, injectivity and subjectivity of functions; and properties of eigenvectors and eigenvalues.

Assignment:

Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding the applications of Linear Algebra and Vector Calculus to engineering applications – The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus of Linear Algebra and Vector Calculus is covered.

Text Book

1. Gilbert Strang, Linear Algebra and its applications, Wellesley-Cambridge Press,U.S.; 6th edition.
2. B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

References:

1. Introduction to Linear Algebra with Application, Jim Defranza, Daniel Gagliardi, Tata McGraw-Hill
2. Elementary Linear Algebra, Applications version, Anton and Rorres, Wiley India Edition.
3. Advanced Engineering Mathematics, Erwin Kreysig, Wiley Publication.
4. Elementary Linear Algebra, Ron Larson, Cengage Learning .
5. Linear Algebra and its Applications, David C. Lay, Pearson Education.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_9607
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_143156
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=CUSTOM_PACKAGE_EBSCO_29052023_270975
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_94555
5. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_243864
6. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_224531
7. NPTEL Video Lectures Matrices and Linear Algebra: https://nptel.ac.in/courses/111106051/
8. NPTEL Video Lectures Differential Equations: https://nptel.ac.in/courses/111106100/
9. NPTEL Vector Calculus: https://nptel.ac.in/courses/111/105/111105122/
10. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
11. https://www.scu.edu.au/study-at-scu/units/math1005/2022/

Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of Vector calculus and Linear Algebra with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem Solving**.** This is attained through the assessment component mentioned in the course handout.

<table>
<tr><td>Code Code:
CSE2251</td><td colspan="3">Course Title: Data Communications and Computer Networks
Type of Course: Theory / PCC</td><td>L- T-P-C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8"></td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">The objective of this course is to provide knowledge in data communications and computer networks, its organization and its implementation, and gain practical experience in the installation, monitoring, and troubleshooting of LAN systems.
The associated laboratory is designed to implement and simulate various networks using Cisco packet tracer, NS2. All the lab exercises will focus on the fundamentals of creating multiple networks, topologies and analyzing the network traffics.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Data Communications and Computer Networks and attain Employability through
Problem Solving Methodologies.</td></tr>
<tr><td>Course Outcomes</td><td colspan="8">On successful completion of the course, the students shall be able to:
1] Ilustrate the Basic Concepts Of Data Communication and Computer Networks.
2] Analyze the functionalities of the Data Link Layer.
3] Apply the Knowledge of IP Addressing and Routing Mechanisms in Computer Networks.
4] Demonstrate the working principles of the Transport layer and Application Layer.</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction and Physical Layer-CO1</td><td colspan="2">Assignment</td><td colspan="3">Problem Solving</td><td colspan="2">7 Sessions</td></tr>
<tr><td colspan="9">Introduction to Computer Networks and Data communications, Network Components – Topologies, Transmission Media –Reference Models -OSI Model – TCP/IP Suite.
Physical Layer -Analog and Digital Signals – Digital and Analog Signals – Transmission - Multiplexing and Spread Spectrum.</td></tr>
<tr><td>Module 2</td><td>Reference Models and Data Link Layer – CO2</td><td colspan="2">Assignment</td><td colspan="3">Problem Solving</td><td colspan="2">7 Sessions</td></tr>
<tr><td colspan="9">Data Link Layer - Error Detection and Correction – Parity, LRC, CRC, Hamming Code, Flow Control and Error Control, Stop and Wait, ARQ, Sliding Window, Multiple Access Protocols, CSMA/CD,CSMA/CA, IEEE 802.3, IEEE 802.11 Ethernet.</td></tr>
<tr><td>Module 3</td><td>Network Layer –CO3</td><td colspan="2">Assignment</td><td colspan="3">Problem Solving</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="9">Network Layer Services - Network Layer Services, Switching Techniques, IP Addressing methods- IPv4 IPV6 – Subnetting. Routing, - Distance Vector Routing – RIP-BGP-Link State Routing –OSPF-Multi cast Routing-MOSPF- DVMRP – Broad Cast Routing. EVPN-VXLAN, VPLS, ELAN.</td></tr>
<tr><td>Module 4</td><td>Transport and Application Layer - CO3</td><td colspan="2">Assignment</td><td colspan="3">Problem Solving</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="9">Transport Layers - Connection management – Flow control – Retransmission, UDP, TCP, congestion control, – Congestion avoidance (DECbit, RED)
The Application Layer: Domain Name System (DNS), Domain Name Space, SSH, FTP, Electronic Mail (SMTP, POP3, IMAP, MIME) – HTTP – – SNMP, Web Services, Virtual Networking.</td></tr>
</table>

Targeted Application & Tools that can be used: Cisco Packet Tracer, Wireshark, and NS2.

Case Study/Assignment: Choose and analyze a network from any organization/Assignment proposed for this course in CO1-CO4

Problem Solving: Choose and appropriate devices and implement various network concepts.

Programming: Simulation of any network using NS2.

Text Book(s):

1. 1. Behrouz A. Forouzan, "Data Communications and Networking 5E", 5 th Edition, Tata McGraw-Hill, 2017.

2. Andrew S Tanenbaum, Nick Feamster & David J Wetherall, "Computer Networks" Sixth Edition, Pearson Publication, 2022

Reference(s):

1.References

1. "Computer Networking: A Top-Down Approach", Eighth Edition, James F. Kurose, Keith W. Ross, Pearson publication, 2021.

2. William Stallings, Data and Computer Communication, 8th Edition, Pearson Education, 2007.

3. Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.

E- Resources:

1. https://archive.nptel.ac.in/courses/106/105/106105183/
2. http://www.nptelvideos.com/course.php?id=393
3. https://www.youtube.com/watch?v=3DZLItfbqtQ
4. https://www.youtube.com/watch?v=_fIdQ4yfsfM
5. https://www.digimat.in/keyword/106.html
6. https://puniversity.informaticsglobal.com/login

<table>
<tr><td>Course Code:
CSE2252</td><td colspan="2">Course Title: Data Communications and Computer Networks Lab
Type of Course: Lab / PCC</td><td>L- T-P- C</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td>Version No.</td><td colspan="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7"></td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">This lab-based course provides hands-on experience in the principles and practices of data communications and computer networking. It is designed to complement theoretical concepts covered in the associated lecture course. Through a series of structured experiments and practical exercises, students will gain proficiency in configuring, analyzing, and troubleshooting computer networks.
Key topics include network topology design, IP addressing and subnetting, Ethernet and LAN technologies, routing and switching, TCP/IP protocol suite, and basic network security measures. Students will work with industry-standard tools and equipment, including routers, switches, protocol analyzers, and network simulation software such as Cisco Packet Tracer or Wireshark.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of this lab course is to provide students with practical, hands-on experience in the configuration, operation, and troubleshooting of data communication systems and computer networks. Through guided experiments and real-world scenarios, students will reinforce theoretical knowledge, develop essential technical skills, and gain a deeper understanding of networking concepts, protocols, and devices used in modern communication systems.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course, the students shall be able to:
1. Design and configure basic network topologies using routers, switches, and end devices to meet specified requirements.
2. Analyze and troubleshoot network connectivity and performance issues using tools such as Wireshark and network simulators.
3. Demonstrate understanding of key networking protocols (e.g., TCP/IP, ARP, ICMP, DHCP) through practical implementation and observation.
4. Apply IP addressing and subnetting techniques to efficiently allocate and manage network resources in various networking scenarios.</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1,2,3,4</td><td>Physical Layer, Network Layer, Transport Laye</td><td>Lab Assignment</td><td colspan="3">Problem Solving</td><td colspan="2">24 Sessions</td></tr>
<tr><td colspan="8">List of Laboratory Tasks:
Lab sheet -1, M-1, 3 [2 Hours]
Experiment No 1:
Level 1: Study of basic network commands and network configuration commands.

Lab sheet -2, M-1[2 Hours]
Experiment No 1:
Level 1: Identify and explore Network devices, models and cables. Introduction to Cisco packet tracer.
Experiment No. 2:
Level 2 – Create various network topologies using a cisco packet tracer.

Lab sheet -3, M-2,3 [2 Hours]
Experiment No. 1:
Level 2 - Basic Configuration of switch/router using Cisco packet tracer.
Experiment No. 2:
Level 2 -Configure the privilege level password and user authentication in the switch/router.

Lab sheet – 4, M-3 [2 Hours]
Experiment No. 1:
Level 2 - Configure the DHCP server and wireless router and check the connectivity

Lab sheet – 5, M-3 [2 Hours]</td></tr>
</table>

Experiment No. 1:
Level 2 - Configure the static routing in the Cisco packet tracer.
Experiment No. 2:
Level 2 - Configure the dynamic routing protocol in the Cisco packet tracer.

Lab sheet – 6, M-4 [2 Hours]
Experiment No. 1: Configuration of DNS Server with Recursive & Integrative approach in Cisco packet tracer.
Lab sheet – 7, M-4 [2 Hours]
Experiment No. 1:
Configure the telnet protocol in the router using the Cisco packet tracer.
Lab sheet – 8, M-4[2 Hours]
Experiment No. 1:
Level1- Introduction to NS2 and basic TCL program.
Lab sheet – 9, M-4 [2 Hours]
Experiment No. 1:
Level 1: Simulate three node Point to point network using UDP in NS2.
Experiment No. 2:
Simulate transmission of Ping message using NS2.
Lab sheet – 10, M-4[2 Hours]
Experiment No. 1:
Simulate Ethernet LAN using N-node in NS2.
Experiment No. 2:
Simulate Ethernet LAN using N-node using multiple traffic in NS2
Lab sheet –11, M-3,4 [2 Hours]
Experiment No. 1:
Level 1- Introduction to Wire Shark.
Experiment No. 2:
Level 2- Demonstration of packet analysis using wire shark.

Lab sheet –12, M-1,2,3 [2 Hours]
Experiment No. 1:
Level 2- Demonstration of switch and router configuration using real devices

Targeted Application & Tools that can be used: Cisco Packet Tracer, Wireshark, and NS2.
Case Study/Assignment: Choose and analyze a network from any organization/Assignment proposed for this course in CO1-CO4
Problem Solving: Choose and appropriate devices and implement various network concepts.
Programming: Simulation of any network using NS2.

Text Book(s):

1. Behrouz A. Forouzan, "Data Communications and Networking 5E", 5 th Edition, Tata McGraw-Hill, 2017.
2. Andrew S Tanenbaum, Nick Feamster & David J Wetherall, "Computer Networks" Sixth Edition, Pearson Publication, 2022

Reference(s):

1. "Computer Networking: A Top-Down Approach", Eighth Edition, James F. Kurose, Keith W. Ross, Pearson publication, 2021.
2. William Stallings, Data and Computer Communication, 8th Edition, Pearson Education, 2007.
3. Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.

E- Resources:

1. https://archive.nptel.ac.in/courses/106/105/106105183/
2. http://www.nptelvideos.com/course.php?id=393
3. https://www.youtube.com/watch?v=3DZLItfbqtQ
4. https://www.youtube.com/watch?v=_fIdQ4yfsfM
5. https://www.digimat.in/keyword/106.html
6. https://puniversity.informaticsglobal.com/login

urse Code: E2253	urse Title: Data Structures pe of Course: Theory	T- P- C	3	0		
rsion No.						
urse Pre-requisites						
ti-requisites						
urse Description	This course introduces the fundamental concepts of data structures and to emphasize the importance of choosing an appropriate data structure and technique for program development .This course has theory and lab component which emphasizes on understanding the implementation and applications of data structures using Java programming language .With a good knowledge in the fundamental concepts of data structures and practical experience in implementing them, the student can be an effective designer, developer for new software applications.					
urse Objective	The objective of the course is SKILL DEVELOPMENT of student by using EXPERIENTIAL LEARNING techniques					
urse Out Comes	**On successful completion of the course the students shall be able to:** **CO1 :**Describe the concept of basic data structure, stacks, queues, and arrays and their operations. **[Understand]** **CO2:** Utilize linked lists for real-time scenarios. **[Apply]** **CO3:** Apply an appropriate non-linear data structure for a given scenario. **[Apply]** **CO4:** Demonstrate different searching and sorting techniques. **[Apply]**					
urse Content:						
dule 1	**roduction to Data Structure and Linear Data Structure –Stacks and Queues**	signment	gram activity	**9 Hours**		
Introduction –Introduction to Data Structures, Types and concept of Arrays **.** **Stack -**Concepts and representation, Stack operations, stack implementation using array and Applications of Stack**.** **Queues -**Representation of queue, Queue Operations, Queue implementation using array, Types of Queue and Applications of Queue**.**						
dule 2	**near Data Structure - Linked List**	signment	Program activity	**12 Hours**		
Topics: Linked List - Singly Linked List, Operation on linear list using singly linked storage structures, Circular List, Applications of Linked list. **Recursion** - Recursive Definition and Processes.						
dule 3	**n-linear Data Structures -Trees**	signment	Program activity	**12 Hours**		
pics: Trees - Introduction to Trees, Binary tree :Terminology and Properties, Use of Doubly Linked List, Binary tree traversals :Pre-Order traversal, In-Order traversal, Post - Order traversalBinary , .Red Black Tree, Expression Tree , Heaps -Serach Tree, AVL Trees						
dule 4	**n-linear Data Structures - Graphs and Hashing**	signment	gram activity	**Hours**		
Topics: Graphs: Basic Concept of Graph Theory and its Properties, Representation of Graphs . ADT, Elementary graph operations, Minimum Cost spanning trees, Shortest path and Transitive closure. **Hashing**: Introduction, Static Hashing, Dynamic Hashing						
Module 5	**Searching & Sorting**	Assignment	Program activity	**6 Hours**		

Topic: Sorting & Searching - Sequential and Binary Search, Sorting – Selection and Insertion sort, Quick sort, Merge Sort, Bubble sort.
t of Laboratory Tasks: **b sheet -1** **Level 1:** Prompt the user, read input and print messages.Programs using class, methods and objects **Level 2:** Programming Exercises on fundamental Data structure - Arrays based on Scenario. **b sheet -2** **Level 1:** Programming Exercises on Stack and its operations **Level 2:** Programming Exercises on Stack and its operations with condition **b sheet -3** **Level 1:** Programming on Stack application infix to postfix Conversion **Level 2:** - **b sheet -4** **vel 1:** Programming on Stack application – Evaluation of postfix **b sheet -5** **Level 1:** Programming Exercises on Queues and its operations with conditions **Level 2:** - **b sheet -6** **Level 1:** Programming Exercises on Linked list and its operations. **Level 2:** Programming Exercises on Linked list and its operations with various positions **b sheet -7** **Level 1:** Programming Exercises on Circular Linked list and its operations. **Level 2:** Programming Exercises on Circular Linked list and its operations with various positions **b sheet -8** **Level 1:** Programming Exercises on factorial of a number **Level 2:** Programming the tower of Hanoi using recursion **b sheet -9** **Level 1:** - **Level 2:** Programming the tower of Hanoi using recursion **b sheet -10** **Level 1:** Programming Exercise on Doubly linked list and its operations **Level 2:** - **b sheet -11** **Level 1:** Program to Construct Binary Search Tree and Graph **Level 2:** Program to traverse the Binary Search Tree in three ways)in-order, pre-order and post-order(and implement BFS and DFS **b sheet -12** **Level 1:** Program to Implement the Linear Search & Binary Search **Level 2:** Program to Estimate the Time complexity of Linear Search **b sheet -13** **Level 1:** Program to Implement and Estimate the Time complexity of Selection Sort **Level 2:** Program to Implement and Estimate the Time complexity of Insertion Sort **b sheet -14 (Beyond syllabus activity)** **vel 1:** Program to Construct AVL Tree **vel 2:** **b sheet -15 (Beyond syllabus activity)** **vel 1:** Program to Construct RED BLACK Tree
rgeted Application & Tools that can be used e of PowerPoint software for lecture slides and use of Modern IDE like VS Code and Eclipse for lab programs to execute.
oject work/Assignment:

signment: Students should complete the lab programs by end of each practical session and module wise assignments before the deadline.

Text Book

T1 Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, reprint 2018.

T2 Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.

ferences

R1 Data structures and program design in C by Robert Kruse, Tondo C L, Bruce Leung, Pearson education publishers, 2017.

R2 Programming and Data Structure by Jackulin C Salini etal., Ane books publishers, 2019.

Web resources:

1. For theory :https://onlinecourses.nptel.ac.in/noc20_cs85/preview
2. https://puniversity.informaticsglobal.com/login

Topics relevant to development of “Skill Development”:

ıked list and stacks

pics relevant to development of “Environment and sustainability: Queues

urse Code: E2254	urse Title: Data Structures Lab pe of Course:Lab	- P- C	0	0		
rsion No.						
urse Pre- requisites						
ti-requisites						
urse Description	This course introduces the fundamental concepts of data structures and to emphasize the importance of choosing an appropriate data structure and technique for program development .This course has theory and lab component which emphasizes on understanding the implementation and applications of data structures using Java programming language .With a good knowledge in the fundamental concepts of data structures and practical experience in implementing them, the student can be an effective designer, developer for new software applications.					
urse Objective	The objective of the course is SKILL DEVELOPMENT of student by using EXPERIENTIAL LEARNING techniques					
urse Out Comes	**On successful completion of the course the students shall be able to:** **CO1 :**Describe the concept of basic data structure, stacks, queues, and arrays and their operations. **[Understand]** **CO2:** Utilize linked lists for real-time scenarios. **[Apply]** **CO3:** Apply an appropriate non-linear data structure for a given scenario. **[Apply]** **CO4:** Demonstrate different searching and sorting techniques. **[Apply]**					
urse Content:						

dule 1	**roduction to Data Structure and Linear Data Structure – Stacks and Queues**	signment	gram activity	**9 Hours**
Introduction –Introduction to Data Structures, Types and concept of Arrays **.** **Stack -**Concepts and representation, Stack operations, stack implementation using array and Applications of Stack**.** **Queues -**Representation of queue, Queue Operations, Queue implementation using array, Types of Queue and Applications of Queue**.**				
Module 2	**near Data Structure - Linked List**	signment	Program activity	**12 Hours**
Topics: Linked List - Singly Linked List, Operation on linear list using singly linked storage structures, Circular List, Applications of Linked list. **Recursion** - Recursive Definition and Processes.				
Module 3	**n-linear Data Structures - Trees**	signment	Program activity	**12 Hours**
pics: Trees - Introduction to Trees, Binary tree :Terminology and Properties, Use of Doubly Linked List, Binary tree traversals :Pre-Order traversal, In-Order traversal, Post - Order traversalBinary , .Red Black Tree, Expression Tree , Heaps -Serach Tree, AVL Trees				
dule 4	**n-linear Data Structures - Graphs and Hashing**	signment	gram activity	**Hours**
Topics: Graphs: Basic Concept of Graph Theory and its Properties, Representation of Graphs . ADT, Elementary graph operations, Minimum Cost spanning trees, Shortest path and Transitive closure. **Hashing**: Introduction, Static Hashing, Dynamic Hashing				
Module 5	**Searching & Sorting**	Assignment	Program activity	**6 Hours**

Topic: Sorting & Searching - Sequential and Binary Search, Sorting – Selection and Insertion sort, Quick sort, Merge Sort, Bubble sort.

t of Laboratory Tasks:

b sheet -1
Level 1: Prompt the user, read input and print messages.Programs using class, methods and objects
Level 2: Programming Exercises on fundamental Data structure - Arrays based on Scenario.

b sheet -2
Level 1: Programming Exercises on Stack and its operations
Level 2: Programming Exercises on Stack and its operations with condition

b sheet -3
Level 1: Programming on Stack application infix to postfix Conversion
Level 2: -

b sheet -4
vel 1: Programming on Stack application – Evaluation of postfix

b sheet -5
Level 1: Programming Exercises on Queues and its operations with conditions
Level 2: -

b sheet -6
Level 1: Programming Exercises on Linked list and its operations.
Level 2: Programming Exercises on Linked list and its operations with various positions

b sheet -7
Level 1: Programming Exercises on Circular Linked list and its operations.
Level 2: Programming Exercises on Circular Linked list and its operations with various positions

b sheet -8
Level 1: Programming Exercises on factorial of a number
Level 2: Programming the tower of Hanoi using recursion

b sheet -9
Level 1: -
Level 2: Programming the tower of Hanoi using recursion

b sheet -10
Level 1: Programming Exercise on Doubly linked list and its operations
Level 2: -

b sheet -11
Level 1: Program to Construct Binary Search Tree and Graph
Level 2: Program to traverse the Binary Search Tree in three ways)in-order, pre-order and post-order(and implement BFS and DFS

b sheet -12
Level 1: Program to Implement the Linear Search & Binary Search
Level 2: Program to Estimate the Time complexity of Linear Search

b sheet -13
Level 1: Program to Implement and Estimate the Time complexity of Selection Sort
Level 2: Program to Implement and Estimate the Time complexity of Insertion Sort

b sheet -14 (Beyond syllabus activity)
vel 1: Program to Construct AVL Tree
vel 2:

b sheet -15 (Beyond syllabus activity)
vel 1: Program to Construct RED BLACK Tree

rgeted Application & Tools that can be used

e of PowerPoint software for lecture slides and use of Modern IDE like VS Code and Eclipse for lab programs to execute.

ject work/Assignment:
signment: Students should complete the lab programs by end of each practical session and module wise assignments before the deadline.
Text Book **T1** Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, reprint 2018. **T2** Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.
ferences **R1** Data structures and program design in C by Robert Kruse, Tondo C L, Bruce Leung, Pearson education publishers, 2017. **R2** Programming and Data Structure by Jackulin C Salini etal., Ane books publishers, 2019. **Web resources:** 3. For theory :https://onlinecourses.nptel.ac.in/noc20_cs85/preview 4. https://puniversity.informaticsglobal.com/login
Topics relevant to development of "Skill Development": ked list and stacks **pics relevant to development of "Environment and sustainability: Queues**

Course Code: **CSE2255**	**Course Title:** Object Oriented Programming Using Java **Type of Course:** **Theory - PCC**	**L-T- P-C**	3	0	0	3

Version No.	2.0
Course Pre-requisites	Nil
Anti-requisites	Nil
Course Description	This course introduces the core concepts of object-oriented programming. This course has theory and lab component which emphasizes on understanding the implementation and application of object-oriented programming paradigm. It helps the student to build real time secure applications by applying these concepts and also for effective problem solving. The students interpret and understand the need for object oriented programming to build applications.
Course Objective	The objective of the course is to familiarize the learners with the concepts of Problem-Solving using JAVA and attain **SKILL DEVELOPMENT** through **EXPERIENTIAL LEARNING** techniques
Course Out Comes	**On successful completion of the course the students shall be able to:** **CO1:** Describe the basic programming concepts. [Understand] **CO2:** Apply the concept of classes, objects and methods to solve problems. [Application] **CO3:** Apply the concept of arrays and strings. [Appy] **CO4:** Implement inheritance and polymorphism building secure applications. [Apply] **CO5:** Apply the concepts of interface and error handling mechanism. [Apply]
Course Content:	

Module 1	**Basic Concepts of Programming and Java**	Assignment	Problem Solving	**9 Sessions**

Topics: Introduction to Principles of Programming: Process of Problem Solving, Java program structure, Download Eclipse IDE to run Java programs, Sample program, Data types, Identifiers, Variables, Constants in java, Operators, Assignments and Expression, Basic Input/ Output functions, Control Statements: Branching and Looping.

Module 2	**Classes, objects, methods and Constructors**	Assignment	Problem Solving	**10 Sessions**

Topics: Classes, Objects and Methods: Introduction to object Oriented Principles, defining a class, adding data members and methods to the class, access specifiers, instantiating objects, reference variable, accessing class members and methods.

Static Polymorphism: Method overloading, constructors, constructor overloading, this keyword, static keyword, Nested classes, Accessing members in nested classes.

Module 3	**Arrays, String and String buffer**	Assignment	Problem Solving	**8 Sessions**

Topics: Arrays: Defining an Array, Initializing & Accessing Array, Multi –Dimensional Array, Array of objects. String: Creation & Operation. String builder class, methods in String Buffer.

Module 4	**Inheritance and Polymorphism**	Assignment	Problem Solving	**10 Sessions**

Topics: Inheritance: Defining a subclass, Types of Inheritance, super keyword. Dynamic Polymorphism: Method overriding. Final keyword: with data members, with member functions and with class. Abstract keyword: with data members, with member functions and with class, Exception handling.

Module 5	**Input & Output Operation in Java**	Assignment	Problem Solving	**8 Sessions**

Input/output Operation in Java(java.io Package), Streams and the new I/O Capabilities, Understanding Streams, working with File Object, File I/O Basics, Reading and Writing to Files, Buffer and Buffer Management, Read/Write Operations with File Channel, Serializing Objects, Observer and Observable Interfaces.

Text Book

T1 Herbert Schildt, "The Complete Reference Java 2", Tata McGraw Hill Education, 11th Edition,2019.

References

R1. Cay S Horstmann and Cary Gornell, "CORE JAVA volume I-Fundamentals", Tenth Edition, Pearson 2015.

R2: James W. Cooper, "Java TM Design Patterns – A Tutorial", Addison-Wesley Publishers.4th Edition, 2000.

R3. E. Balagurusamy, "Programming with Java", Tata McGraw Hill Education, 6th Edition, 2019.

E book link R1: **http://rmi.yaht.net/bookz/core.java/9780134177373-Vol-1.pdf**

E book link R2: Java(tm) Design Patterns: A Tutorial([PDF] [7qmsenjl97t0] (vdoc.pub)

Web **resources**

https://youtube.com/playlist?list=PLu0W_9lII9agS67Uits0UnJyrYiXhDS6q

https://puniversity.informaticsglobal.com:2229/login.aspx

Topics relevant to development of "Skill Development":

1. Static Polymorphism
2. Method overloading, constructors
3. constructor overloading
4. this keyword
5. static keyword and Inner classes
6. Inheritance and Polymorphism.

for **Skill Development** through **Experiential Learning** techniques. This is attained through assessment component mentioned in course handout.

<table>
<tr><td>Course Code:
CSE2256</td><td colspan="3">Course Title: Object Oriented Programming Using Java Lab
Type of Course: Lab - PCC</td><td>L-T- P- C</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td>Version No.</td><td colspan="8">2.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8"></td></tr>
<tr><td>Anti-requisites</td><td colspan="8">Nil</td></tr>
<tr><td>Course Description</td><td colspan="8">This course introduces the core concepts of object-oriented programming. This course has theory and lab component which emphasizes on understanding the implementation and application of object-oriented programming paradigm. It helps the student to build real time secure applications by applying these concepts and also for effective problem solving. The students interpret and understand the need for object oriented programming to build applications.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Problem-Solving using JAVA and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of the course the students shall be able to:
CO1: Demonstrate basic programming concepts. [Apply]
CO2: Apply the concept of classes, objects and methods to solve problems. [Application]
CO3: Apply the concept of arrays and strings. [Appy]
CO4: Implement inheritance and polymorphism building secure applications. [Apply]
CO5: Apply the concepts of interface and error handling mechanism. [Apply]</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Basic Concepts of Programming and Java</td><td>Assignment</td><td colspan="2">Problem Solving</td><td colspan="4">12 Sessions</td></tr>
<tr><td colspan="9">Download Eclipse IDE to run Java programs, Sample programs on Data types, Identifiers, Variables, Constants in java, Operators, Assignments and Expression, Basic Input/ Output functions, Control Statements: Branching and Looping.</td></tr>
<tr><td>Module 2</td><td>Classes, objects, methods and Constructors</td><td>Assignment</td><td colspan="2">Problem Solving</td><td colspan="4">14 Sessions</td></tr>
<tr><td colspan="9">Problem solving using Classes, Objects and Methods: defining a class, adding data members and methods to the class, access specifiers, instantiating objects, reference variable, accessing class members and methods.
Use Static Polymorphism: Method overloading, constructors, constructor overloading, this keyword, static keyword, Nested classes, Accessing members in nested classes.</td></tr>
<tr><td>Module 3</td><td>Arrays, String and String buffer</td><td>Assignment</td><td colspan="2">Problem Solving</td><td colspan="4">10 Sessions</td></tr>
<tr><td colspan="9">Using Arrays and Strings : Defining an Array, Initializing & Accessing Array, Multi –Dimensional Array, Array of objects. String: Creation & Operation. String builder class, methods in String Buffer.</td></tr>
<tr><td>Module 4</td><td>Inheritance and Polymorphism</td><td>Assignment</td><td colspan="2">Problem Solving</td><td colspan="4">12 Sessions</td></tr>
<tr><td colspan="9">Inheritance: Defining a subclass, Types of Inheritance, super keyword. Dynamic Polymorphism: Method overriding. Final keyword: with data members, with member functions and with class. Abstract keyword: with data members, with member functions and with class, Exception handling.</td></tr>
<tr><td>Module 5</td><td>Input & Output Operation in Java</td><td>Assignment</td><td colspan="2">Problem Solving</td><td colspan="4">12 Sessions</td></tr>
<tr><td colspan="9">Input/output Operation in Java(java.io Package), Streams and the new I/O Capabilities, Understanding Streams, working with File Object, File I/O Basics, Reading and Writing to Files, Buffer and Buffer Management, Read/Write Operations with File Channel, Serializing Objects, Observer and Observable Interfaces.</td></tr>
<tr><td colspan="9">P1: Programming Exercises on Basic Concepts.
LEVEL 1: Discuss about datatypes and variables.</td></tr>
</table>

LEVEL 2: Demonstrate a simple java program
P2: Programming Exercises on Basic Concepts.
LEVEL 1: Discuss about datatypes and variables.
LEVEL 2: Demonstrate a simple java program
P3: Programming Exercises on operators, expressions based on a given scenario.
LEVEL 1: Explain operators, expressions.
LEVEL 2: Demonstrate operators
P4: Programming Exercises Command Line Arguments based on a given scenario.
LEVEL 1: Explain command line arguments
LEVEL 2: Demonstrate command line arguments
P5: Programming Exercises on basic Input/ Output functions and Control Statements: Branching
LEVEL 1: Explain Input/ Output functions
LEVEL 2:Demonstrate Control Statements: Branching
P6: Programming Exercises on Control Statements: Looping
LEVEL 1: Explain variour loops.
LEVEL 2:Demonstrate Control Statements: Looping
P7: Programming Exercises on Creating Objects, classes on a given scenario.
LEVEL 1: Illustrate class, object and methods.
LEVEL 2: Execute java program using class and objects
P8: Programming Exercises on Adding methods and Constructors to the class based on a given scenario.
LEVEL 1: Illustrate methods and constructors
LEVEL 2: Execute java program using methods and constructors
P9: Programming Exercises on methods based on a given scenario.
LEVEL 1: Illustrate method overloading
LEVEL 2: Apply method overloading for the given scenario.
P10: Programming Exercises on methods based on a given scenario.
LEVEL 1: Illustrate constructors overloading
LEVEL 2: Apply constructor overloading for the given scenario
P11: Programming Exercises on methods for static members bassed on a given scenario.
LEVEL 1: Benefits of usage static members
LEVEL 2: Usage of Static Members for the given scenario
P12: Programming Exercises on static methods based on a given scenario.
LEVEL 1: Benefits of usage static methods
LEVEL 2: Usage of Static Methods for the given scenario.
P13: Programming Exercises on nested Classes based on a given scenario.
LEVEL 1: Benefits of usage nested classes
LEVEL 2: Apply the concept of usage of nested classes for the given scenario
P14: Programming Exercises on Arrays and its built-in functions based on a given scenario.
LEVEL 1: Illustrate one dimensional arrays and its functions.
LEVEL 2: Demonstrate programs with single-dimensional arrays and operations.
P15: Programming Exercises on Arrays and its built-in functions based on a given scenario.
LEVEL 1: Illustrate multi dimensional arrays and its functions.
LEVEL 2: Demonstrate programs with multi-dimensional arrays and operations.
P16: Programming Exercises on String Class and its built-in functions based on a given scenario.
LEVEL 1: Explain about String class and String methods.
LEVEL 2: Execute simple java applications for String and StringBuffer operations
P17: Programming Exercises on String Buffer Class and its built-in functions based on a given scenario.
LEVEL 1: Explain about StringBuffer class and String methods.
LEVEL 2: Execute simple java applications for String and StringBuffer operations
P18: Programming Exercises on String Builders and its built-in functions based on a given scenario.
LEVEL 1: Explain about String Builders.
LEVEL 2: Execute java applications for String Builders
P19: Programming Exercises on single, multi level Inheritance and super keyword based on given scenario.
LEVEL 1: Explain single and multi level inheritance.
LEVEL 2: Demonstrate simple applications for the different types of inheritance

P20: Programming Exercises hierarchical Inheritance and super keyword based on given scenario.
LEVEL 1: Explain hierarchical inheritance.
LEVEL 2: Demonstrate simple applications for hierarchical inheritance
P21: Programming Exercises on Overriding.
LEVEL 1: Differentiate method overloading and method overriding.
LEVEL 2: Demonstrate simple program with dynamic method dispatch.
P22: Programming Exercises on Final based on given scenario.
LEVEL 1: Implement programs using concept of final.
LEVEL 2: Use final keyword for the given problem
P23: Programming Exercises on Abstract keyword based on given scenario.
LEVEL 1: Implement programs using concept of Abstract.
LEVEL 2: Use abstract keyword for the given problem
P24: Programming Exercises on Interface based on a given scenario.
LEVEL 1: Differentiate abstract class about interface
LEVEL 2: Implement interfaces in the given problem
P25: Programming Exercises on Exception Handling based on a given scenario.
LEVEL 1: Explain exception handling
LEVEL 2: Solve the given problem using exception handling mechanism.
P26: Programming Exercises on Character Stream Classes based on a given scenario.
LEVEL 1: Explain Character Stream Classes
LEVEL 2: Solve the given problem using Character Stream Class.
P27: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.
LEVEL 1: Explain Read/Write Operations with File Channel
LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.
P28: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.
LEVEL 1: Explain Read/Write Operations with File Channel
LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.
P29: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.
LEVEL 1: Explain Read/Write Operations with File Channel
LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.
P30: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.
LEVEL 1: Explain Read/Write Operations with File Channel
LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.

Targeted Application & Tools that can be used : JDK /Eclipse IDE/Visual Studio Code / net Beans IDE.

Text Book

T1 Herbert Schildt, "The Complete Reference Java 2", Tata McGraw Hill Education, 11th Edition,2019.

References

R1. Cay S Horstmann and Cary Gornell, "CORE JAVA volume I-Fundamentals", Tenth Edition, Pearson 2015.
R2: James W. Cooper, "Java TM Design Patterns – A Tutorial", Addison-Wesley Publishers.4th Edition, 2000.
R3. E. Balagurusamy, "Programming with Java", Tata McGraw Hill Education, 6th Edition, 2019.

E book link R1: **http://rmi.yaht.net/bookz/core.java/9780134177373-Vol-1.pdf** **E book link R2:**
Java(tm) Design Patterns: A Tutorial([PDF] [7qmsenjl97t0] (vdoc.pub)

Web **resources**
https://youtube.com/playlist?list=PLu0W_9lII9agS67Uits0UnJyrYiXhDS6q
https://puniversity.informaticsglobal.com:2229/login.aspx

Topics relevant to development of "Skill Development":
1. Static Polymorphism
3. Method overloading, constructors
4. constructor overloading

5. this keyword
6. static keyword and Inner classes
7. Inheritance and Polymorphism.

for **Skill Development** through **Experiential Learning** techniques. This is attained through assessment component mentioned in course handout.

urse Code: E2257	**urse Title: Computer Organization and Architecture** **pe of Course: PCC**	**T-P- C**	3	0	0	
rsion No.						
urse Pre-requisites						
ti-requisites						
urse Descripti on	This course introduces the core principles of computer architecture and organization from basic to intermediate level. This theory based course emphasizes on understanding the interaction between computer hardware and software. It equips the students with the intuition behind assembly-level instruction set architectures. It helps the students to interpret the operational concepts of computer technology as well as performance enhancement.					
urse Objective	The objective of the course is to familiarize the learners with the concepts of **Computer Organization and Architecture** and attain **Skill Development** through **Participative Learning** techniques.					
urse Outcomes	On successful completion of the course the students shall be able to: 1] Describe the basic components of a computer and their interconnections. [Remember] 2] Explain Instruction Set Architecture and Memory Unit[Understand] 3] Apply appropriate techniques to carry out selected arithmetic operations [Apply] 4] Explain the organization of memory and processor sub-system [Understand]					
urse Content:						

Module 1	sic Structure of Computer	Assignment	Data Analysis task	**12 Sessions**

Topics:
Computer Types, Functional Units, Basic Operational concepts, Bus Structures, Computer systems RISC & CISC, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Arithmetic Operations on Signed numbers. Instructions and Instruction Sequencing, Instruction formats,
Memory Instructions.

dule 2	Instruction Set chitectureand Memory Unit	signment	alysis, Data Collection	**12 Sessions**

Topics:
Instruction Set Architecture: Addressing Modes, Stacks and Subroutines.
mory System: Memory Location and Addresses, Memory Operations, Semiconductor RAM Memories, Internal Organization of Memory chips, Cache memory mapping Techniques.

dule 3	Arithmetic d Input/output Design	se Study	ta analysis task	**10 Sessions**

Topics:
Arithmetic: Carry lookahead Adder, Signed-Operand Multiplication, Integer Division, and Floating pointoperations.
ut/output Design: Accessing I/O Devices, I/O communication, Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits

Module 4	BPU and Pipelining	Assignment	halysis, Data Collection	**11 Sessions**

Topics:
Basic Processing Unit: Fundamental Concepts, Single Bus organization, Control sequence, Execution of a Complete Instruction, Multiple Bus Organization.
elining: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, Hazards.

rgeted Application & Tools that can be used:
Targeted employment sector is processor manufacturing and memory chip fabrication vendors like Intel,AMD, Motorola, NVidia, Samsung, Micron Technology, western Digital etc. Targeted job profiles include Memory circuit design and verification engineers, Physical system design engineer, System programmer,Fabrication engineer etc.

Tools:
Virtual Lab, IIT KGP
Tejas – Java Based Architectural Simulator, IIT Delhi

ject work/Assignment:
h batch of students (self-selected batch mates – up to 4 in a batch) will be allocated case studies/assignments
xtbook(s): **1.** Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", Sixth Edition, McGraw-Hill Higher Education, 2023 reprint. 2. William Stallings, "Computer Organization & Architecture – Designing for Performance", 11th Edition, Pearson Education Inc., 2019.
ferences 1. David A. Patterson & John L. Hennessy, "Computer Organization and Design MIPS Edition- The Hardware/Software Interface", 6th Edition, Morgan Kaufmann, Elsevier Publications, November 2020. 2. Web References: 3. NPTEL Course on "Computer architecture and organization" IIT Kharagpur By Prof. Indranil Sengupta, Prof. Kamalika Datta. https://nptel.ac.in/courses/106105163 4. NPTEL Course on "Computer Organization", IIT Madras By Prof. S. Raman. 5. https://nptel.ac.in/courses/106106092 6. https://puniversity.informaticsglobal.com:2229/login.aspx
Topics relevant to "SKILL DEVELOPMENT": Generation of Computers, CISC and RISC processors, Bus Arbitration, Collaboration and Data collection for Term assignments and Case Studies for Skill Developmentthrough Participative Learning techniques. This is attained through assessment component mentioned in course handout.

urse Code: N1002	urse Title: Essentials of Finance pe of Course: HSMC		L-T-P-C	3	)	0	3
rsion No.						
urse Pre-requisites	This course is designed to be accessible to all students, regardless of their prior financial knowledge.					
ti-requisites						
urse Descriptio n	This course is designed to equip students with a **foundational understanding of key financial concepts and principles**. It will enable them to comprehend the **core functions of finance**, delve into the intricacies of **financial management within organizations**, and gain insights into the **fundamental aspects of taxation**. The course aims to develop students' abilities to **interpret financial statements, evaluate investment opportunities, understand capital structure decisions**, and navigate the **basics of tax implications**.					
urse Objective	Upon successful completion of this course, students will be able to: 1. **Understand the basic forms of business organization and their financial implications.** 2. **Understand the fundamental principles and concepts** that influence financial decision-making in various contexts. 3. **Analyse and interpret financial statements** to assess the financial health and performance of an organization. 4. **Identify income under various heads of income as per Income Tax Act, 1961 and determine the tax liability.**					
urse Outcomes	**On successful completion of this course the students shall be able to:** **1. Understand the basic concepts of finance and financial markets and organizations.** 2. **Apply and interpret financial information for business decision making.** 3. **Identify various heads of income and deduction under Income Tax Act, 1961.**					
urse Content:						
dule 1	**Introduction to Finance**	signment/ Quiz	merical solving Task	**10 Sessions**		
Definition and Scope of Finance, Areas of Finance: Corporate Finance, Investments, Financial Institutions, International Finance; Types of Financial Markets: Money Markets vs. Capital Markets, Primary vs. Secondary Markets; Forms of Business Organization and Financial Goals: Shareholder Wealth Maximization vs. Profit Maximization; Understanding Financial Statements: Balance Sheet and Income Statement- Simple Numerical.						
dule 2	**ancial Manageme nt**	signment/ Quiz	merical solving Task	**Sessions**		
Capital Budgeting Decisions: Payback Period, Net Present Value (NPV), Profitability Index (PI), Internal Rate of Return (IRR); Leverage- Basic Numerical; Capital Structure Decisions: Optimal Capital Structure, Trade-off Theory of Capital Structure; Cost of Capital: Equity, Debt, WACC; Dividend Policy: Factors influencing Dividend Policy.						
dule 3	**xation**	signment/ Quiz	merical solving Task	**17 Sessions**		
Principles of a Good Tax System: Equity, Certainty, Convenience, Economy; Direct vs. Indirect Taxes; Residential Status of an Individual- Basic Problems; Heads of Income; Salary, House Property- Basic Numerical; Deductions under Chapter VI-A; Computation of Taxable Income and Tax Liability; E-Filing procedure.						
rgeted Application & Tools that can be used: **xtbooks, PPT, Spreadsheet Software (e.g., Microsoft Excel), Official Website of Income Tax Department.**						
oject Work/ Assignment:						
1. Presentation: There will be a group presentation, where the students will be given a topic. They will have to explain/demonstrate the working and discuss the applications for the same. **2. Case Study**: - At the end of the course students will be given a 'real-world' cases like business models of successful companies or tax evasion by reputed companies on which they have to come up with detailed analysis and assessment.						
xt Book(s):						
1. **Dr. Vinod K. Singhania & Dr. Monica Singhania.** (Latest Assessment Year Edition). *Students' Guide to Income Tax including GST*. Taxmann Publications. **2. Pandey, I. M.** (2025). *Financial Management*. Vikas Publishing House.						
ference Book (s):						

1. **Bhole, L.M., & Mahakud, J.** (Current Edition). *Financial Institutions and Markets: Structure, Growth and Innovations*. McGraw Hill Education India.
2. **Mehrotra, H.C., & Goyal, S.P.** (Latest Assessment Year Edition). *Income Tax Law & Practice*. Sahitya Bhawan Publications.
3. **Gordon, E., & Natarajan, K.** (Current Edition). *Financial Markets and Services*. Himalaya Publishing House.

Online Resources (e-books, notes, ppts, video lectures etc.):

7. https://presidencyuniversity.linways.com
8. https://onlinecourses.nptel.ac.in/noc24_ec01/preview
9. https://www.incometax.gov.in/iec/foportal/

Topics relevant to "SKILL DEVELOPMENT": This course is designed to provide practical financial skills through participative learning techniques. Students will engage in performing suitable calculations to determine financial parameters (e.g., time value of money, investment returns, tax liabilities) and analysing financial statements to assess organizational performance and make informed decisions.

<table>
<tr><td>Course Code:
CIV7601</td><td colspan="2">Course Title: Universal Human Values and Ethics
Type of Course: MAC course</td><td>L-T-P-C</td><td>-</td><td>-</td><td>-</td><td>0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">The purpose of the course is to develop a holistic perspective in students' life. The course adopts a self-reflective methodology of teaching and is designed to equip the students to explore their role in all aspects of living as a part of the society. It presents a universal approach to value education by developing the right understanding of reality through the process of self-exploration.
This self-exploration develops more confidence and commitment in students enabling them to critically evaluate their pre-conditioning and present beliefs. As an outcome of the holistic approach, the students will be able to practice the ethical conduct in the social and professional life. The prime focus throughout the course is toward affecting a qualitative transformation in the life of the student rather than just a transfer of information.
This course is designed to cater to Human Values and Professional Ethics.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is 'SKILL DEVELOPMENT' of the student by using 'SELF LEARNING' techniques</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of this course the students shall be able to:
CO.1 Recognize the importance of Value Education through the process of self-exploration
CO.2 Explain the human being as the co-existence of the self and the body in harmony.
CO.3 Describe the role of foundational values in building harmonious relationships.
CO.4 Summarize the importance of a holistic perspective in developing ethical professional behavior.</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Introduction to Value Education</td><td>Online Assessment</td><td colspan="2">MCQ Quiz</td><td colspan="3">5 Sessions</td></tr>
<tr><td colspan="8">Topics:
Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education), Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfil the Basic Human Aspirations.</td></tr>
<tr><td>Module 2</td><td>Harmony in the Human Being</td><td>Online Assessment</td><td colspan="2">MCQ Quiz</td><td colspan="3">5 Sessions</td></tr>
<tr><td colspan="8">Topics:
Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health</td></tr>
<tr><td>Module 3</td><td>Harmony in the Family and Society</td><td>Online Assessment</td><td colspan="2">MCQ Quiz</td><td colspan="3">5 Sessions</td></tr>
<tr><td colspan="8">Topics:
Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.</td></tr>
<tr><td>Module 4</td><td>Implications of the Holistic Understanding – A Look at Professional Ethics</td><td>Online Assessment</td><td colspan="2">MCQ Quiz</td><td colspan="3">5 Sessions</td></tr>
<tr><td colspan="8">Topics:
Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics, Holistic Technologies, Strategies for Transition towards Value-based Life and Profession</td></tr>
<tr><td colspan="8">Targeted Application & Tools that can be used:
Application areas are Personal life, Education and Career, Workplace , Society and Environmental Responsibility
Tools: Online Tools – NPTEL and Swayam.</td></tr>
<tr><td colspan="8">Project work/Assignment:</td></tr>
<tr><td colspan="8">Assessment Type
• Online exams (MCQs) will be conducted by the Department of Civil Engineering through Linways.</td></tr>
<tr><td colspan="8">Online Link*:
1) UHV II -
https://www.youtube.com/watch?v=NhFBzn5qKIM&list=PLWDeKF97v9SO8vvjC1KyqteziTbTjN1So&pp</td></tr>
</table>

=0gcJCWMEOCosWNin
2) Lecture by Dr. Kumar Sambhav, NPTEL course: Universal Human Values, https://onlinecourses.swayam2.ac.in/aic22_ge23/preview
3) Lecture by Dr. Padmavati, Dr Narendran Thiruthy, NPTEL Course: Biodiversity Protection, Farmers and Breeders Rights, https://nptel.ac.in/courses/129105008, 2024.

* Other source links are available in below Resources link.

Text Book

1. A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2019.
3. Premvir Kapoor, Professional Ethics and Human Values, Khanna Book Publishing, New Delhi, 2022.

Reference Books

1. E.F. Schumacher, 1973, Small is Beautiful: a study of economics as if people mattered, Blond & Briggs, Britain.
2. Sussan George, 1976, How the Other Half Dies, Penguin Press. Reprinted 1986.
3. Donella H. Meadows, Dennis L. Meadows, Jorgen Randers, William W. Behrens III, 1972, Limits to Growth – Club of Rome's report, Universe Books.
4. A Nagraj, 1998, Jeevan Vidya Ek Parichay, Divya Path Sansthan, Amarkantak.
5. P L Dhar, RR Gaur, 1990, Science and Humanism, Commonwealth Publishers.
6. A N Tripathy, 2003, Human Values, New Age International Publishers.
7. E G Seebauer& Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers, Oxford University Press
8. M Govindrajran, S Natrajan& V.S. Senthil Kumar, Engineering Ethics (including Human Values), Eastern Economy Edition, Prentice Hall of India Ltd.
9. B P Banerjee, 2005, Foundations of Ethics and Management, Excel Books.
10. William P. Cunningham and Mary Ann Cunningham (2020), Principles of Environmental Science: Inquiry & Applications, 9th Edition, McGraw-Hill Education, USA.

Resources:

1. https://onlinecourses.swayam2.ac.in/imb25_mg195/preview
2. https://onlinecourses.nptel.ac.in/noc25_mg141/preview
3. https://onlinecourses.swayam2.ac.in/ini25_hs52/preview
4. https://onlinecourses.nptel.ac.in/noc25_hs219/preview
5. https://onlinecourses.swayam2.ac.in/cec25_mg14/preview
6. https://onlinecourses.swayam2.ac.in/imb25_mg195/preview
7. https://onlinecourses.swayam2.ac.in/imb25_mg196/preview

Topics relevant to Skill Development:

1. An attitude of enquiry.
2. Write reports

The topics related to Human values and Professional ethics:

All topics in are relevant to Human values and Professional ethics.

Course Code: APT4002	Course Title: **Introduction to Aptitude** (Audited)	L-T-P- C	0	0	2	0
Version No.	1.0					
Course Pre-equisites						
Anti-requisites	Nil					
Course Description	The objective of this course is to prepare the trainees to tackle the questions on various topics and various difficulty levels based on Quantitative Ability, and Logical Reasoning asked during the placement drives. There will be sufficient focus on building the fundamentals of all the topics, as well as on solving the higher order thinking questions. The focus of this course is to teach the students to not only get to the correct answers, but to get there faster than ever before, which will improve their employability factor.					

Course Objective	The objective of the course is to familiarize the learners with the concepts of Aptitude and attain Skill Development through Problem Solving techniques.			
Course Outcomes	On successful completion of the course the students shall be able to: CO1] **Recall** all the basic mathematical concepts they learnt in high school. CO2] **Identify** the principle concept needed in a question. CO3] **Solve** the quantitative and logical ability questions with the appropriate concept. CO4] **Analyze** the data given in complex problems. CO5] **Rearrange** the information to simplify the question			
Course Content:				
Module 1	Quantitative Ability	Assignment	Bloom's Level : Application	**12 Hours**
Topics: Introduction to Aptitude, working of Tables, Squares, Cubes				
Module 2	Logical Reasoning	Assignment	Bloom's Level : Application	**18 Hours**
Topics: Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Ordering and Ranking, Clocks and Calendars, Number Series, Wrong number series, Visual Reasoning				
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS				
Text Book 1. Quantitative Aptitude by R S Aggarwal 2. Verbal & Non-Verbal Reasoning by R S Aggarwal				
References 1. www.indiabix.com 2. www.youtube.com/c/TheAptitudeGuy/videos				
Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.				

Course Code: MAT2404	**Course Title: Discrete Mathematics** **Type of Course: Theory - ESC**		**L-T- P- C**	**3**	**1**	**0**	**4**

Version No.	1.0
Course Pre-requisites	MAT2302
Anti-requisites	**NIL**
Course Description	The course explores the study of mathematical structures that are fundamentally discrete (not continuous), focusing on concepts like set theory, logic, graph theory, combinatorics, and number theory, with applications primarily in computer science fields like algorithms, software development, and cryptography; it covers topics such as propositional logic, proof techniques, relations, functions, counting principles, and basic graph algorithms, providing a foundation for analyzing discrete problems and structures within computer science.
Course Objective	The main objective of the course is that students should learn a particular set of mathematical facts and how to apply them. It teaches students how to think logically and mathematically through five important themes: mathematical reasoning, combinatorial analysis, discrete structures, algorithmic thinking, and applications and modeling. A successful discrete mathematics course should carefully blend and balance all five themes.
Course Outcomes	On successful completion of the course the students shall be able to: CO1 - Explain logical sentences through predicates, quantifiers and logical connectives. CO2 - Deploy the counting techniques to tackle combinatorial problems CO3 - Comprehend the basic principles of set theory and different types of relations. CO4 - Apply different types of structures of trees for developing programming skills
Course Content:	

Module 1	**Fundamentals of Logic**		**(10 Classes)**
Basic Connectives and Truth Tables, Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.			
Module 2	**Principle of Counting**	**Assignment**	**(15 Classes)**
The Well Ordering Principle – Mathematical Induction The Basics of Counting, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Generating Permutations and Combinations Advanced Principle Counting: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials.			
Module 3	**Relations and Functions**		**(10 Classes)**
Cartesian Products and Relations, Functions, One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions. Relations, Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders, Lattice, Hasse Diagrams, Equivalence Relations and Partitions.			
Module 4	**Recurrence Relations and Generating Functions**		**(10 Classes)**
Homogeneous and inhomogeneous recurrences and their solutions - solving recurrences using generating functions - Repertoire method - Perturbation method - Convolutions - simple manipulations and tricks.			
Module 5	**Graph Theory & Algorithms on Networks**	**Assignment**	**(15 Classes)**
Definitions and basic results - Representation of a graph by a matrix and adjacency list - Trees - Cycles - Properties - Paths and connectedness - Sub graphs - Graph Isomorphism - Operations on graphs - Vertex and edge cuts - Vertex and edge connectivity, Euler and Hamilton Paths, Shortest-Paths. Tree - Definitions, Properties, and Examples, Routed Trees, Binary search tree, Decision tree, spanning tree: BFS, DFS. Algorithms on Networks - Shortest path algorithm- Dijikstra's algorithm, Minimal spanning tree- Kruskal			

algorithm and Prim's algorithm.

Targeted Application & Tools that can be used:
Discrete mathematics provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems.

Assignment:

Assignment 1: Logic Equivalences and Predicate calculus.
Assignment 2: Equivalence Relations and Lattices
Assignment 3: Recurrence Relations

Text Book

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", McGraw-Hill,s 8th Edition,2019.
2. Harary – Graph Theory, Addison-Wesley Publishing Company.

References:

1. Arthur Gill, "Applied Algebra for Computer Science", Prentice Hall.
2. K.D. Joshi, "Discrete Mathematics", Wiley Eastern Ltd.
3. Ralph. P. Grimaldi., "Discrete and Combinatorial Mathematics: An Applied Introduction", 4th Edition, Pearson Education Asia.

E-resources/ Web links:

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_iBSCO95_30102024_54588
https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_iBSCO95_30102024_375
https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
https://www.scu.edu.au/study-at-scu/units/math1005/2022/

Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of calculus and differential equation with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem solving**.** This is attained through the assessment component mentioned in course handout.

<table>
<tr><td>Course Code:
CSE2258</td><td colspan="3">Course Title: Web Technologies
Type of Course: Program Core Theory (PCC)</td><td>L-T- P- C</td><td>2</td><td>0</td><td>0</td><td>2</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8"></td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course highlights the comprehensive introduction to scripting languages that are used for creating web-based applications.
The associated laboratory provides an opportunity to implement the concepts and enhance critical thinking and analytical skills.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Web Technology and attain Skill Development through Experiential Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="8">On successful completion of this course the students shall be able to:
CO1: Implement web-based application using client-side scripting languages. (Apply)
CO2: Apply various constructs to enhance the appearance of a website. (Apply)
CO3: Apply server-side scripting languages to develop a web page linked to a database. (Apply)</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction to XHTML</td><td>Quizzes and Assignments</td><td colspan="2">Quizzes on various features of XHTML, simple applications</td><td colspan="4">20 Sessions</td></tr>
<tr><td colspan="9">Basics: Web, WWW, Web browsers, Web servers, Internet.
XHTML: Origins and Evolution of HTML and XHTML: Basic Syntax, Standard XHTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, Frames, Syntactic Differences between HTML and XHTML, Demonstration of applications using XHTML for Responsive web pages.</td></tr>
<tr><td>Module 2</td><td>Advanced CSS</td><td>Quizzes and assignments</td><td colspan="2">Comprehension based Quizzes and assignments; Application of CSS in designing webpages</td><td colspan="4">20 Sessions</td></tr>
<tr><td colspan="9">Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks
XML: Basics, Demonstration of applications using XML with XSLT.</td></tr>
<tr><td>Module 3</td><td>PHP –
Application Level</td><td>Quizzes and assignments</td><td colspan="2">Application of PHP in web designing</td><td colspan="4">20 Sessions</td></tr>
<tr><td colspan="9">PHP: Introduction to server-side Development with PHP, Arrays, Superglobal Arrays, $GET and $ POST, $_SERVER Array, $_Files Array, Reading/Writing Files, PHP Classes and Objects, Object Oriented Design, Working with Databases, SQL, Database APIs, Managing a MySQL Database. Accessing MySQL in PHP, Applications.</td></tr>
</table>

List of Laboratory Tasks:

Experiment No. 1: Demonstration of XHTML features

Level 1: Demonstration of various XHTML Tags (Level 1)

Level 2: Design and develop static web pages for an online Book store (Level 2).

Experiment No. 2: Application of CSS in web designing

Level 1: Design a document using XHTML and CSS to create a catalog of items for online electronic shopping.

Level 2: Create and save XML document for students' information and display the same using cascaded style sheet.

Experiment No. 3: Application of PHP in web designing.

Level 1: Write a PHP program to read the personal information of a person such as first name, last name, age, permanent address, and pin code entered by the user into a table created in MySQL. Read the same information from the database and display it on the front end.

Level 2: Using PHP develop a web page that accepts book information such as ISBN number, title, authors, edition, and publisher and store information submitted through the web page in MySQL database.

Experiment No. 4: Building a website.

Build a website for organizing an International Conference. The conference website must be able to collect the author's details and upload a file.

Targeted Application & Tools that can be used:

Xampp web server to be used to demonstrate PHP.

Project work/Assignment:

Assignments are given after completion of each module which the student need to submit within the stipulated deadline.

Textbook(s):

1. Robert. W. Sebesta, "*Programming the World Wide Web"*, Pearson Education, 9th Edition, 2016.
2. Paul Deitel, Harvey Deitel, Abbey Deital,"Internet & World Wide Web How to Program", Fifth Edition, Pearson Education, 2021.
3. *CSS Notes for Professionals*, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022)
4. Deitel, Deitel, Goldberg,"*Internet & World Wide Web How to Program*", Fifth Edition, Pearson
5. Education, 2021.

Reference Book(s):

R1. Randy Connolly, Ricardo Hoar,"Fundamentals of Web Development", Pearson Education India, 1st. Edition.2016.

R2. Jeffrey C. Jackson,"Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition,2016.

Additional web-based resources

W1. W3schools.com

W2. Developer.mozilla.org/en-US/docs/Learn

W3. docs.microsoft.com

W4. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking

https://presiuniv.knimbus.com/user#/home

Topics related to development of "FOUNDATION":

1. Web, WWW, Web browsers, Web servers, Internet.
2. CSS, PHP.
3. Designing the website for healthcare.

The objective of the course is to familiarize the learners with the concepts of Web Technology and attain Skill Development through Experiential Learning techniques.

Course Code: **CSE2259**	**Course Title: Web Technologies Lab** **Type of Course: Program Core Lab**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	This course highlights the comprehensive introduction to scripting languages that are used for creating web-based applications. The associated laboratory provides an opportunity to implement the concepts and enhance critical thinking and analytical skills.					
Course Objective	**The objective of the course is to familiarize the learners with the concepts of Web Technology and attain Skill Development through Experiential Learning techniques.**					
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1:** Implement web-based application using client-side scripting languages. **(Apply)** **CO2**: Apply various constructs to enhance the appearance of a website. **(Apply)** **CO3:** Apply server-side scripting languages to develop a web page linked to a database. **(Apply)**					
Course Content:						

Module 1	**Introduction to XHTML Features**	Quizzes and Assignments	Quizzes on various features of XHTML, simple applications		**8 Sessions**
Standard XHTML Document Structure, Basic Text Markup such as headings, paragraphs, lists, tables, forms, and semantic tags.					
Module 2	**CSS Styling**	Quizzes and assignments	Comprehension based Quizzes and assignments; Application of CSS in designing webpages		**10 Sessions**
Apply CSS3 to style HTML elements, including layout techniques, color schemes, typography, and responsive design principles.					
XML: Basics, Demonstration of applications using XML with XSLT.					

Module 3	**PHP – Application Level**	Quizzes and assignments	Application of PHP in web designing	**12 Sessions**
PHP: Introduction to server-side Development with PHP, Arrays, Superglobal Arrays, $GET and $ POST, $_SERVER Array, $_Files Array, Reading/Writing Files, PHP Classes and Objects, Object Oriented Design, Working with Databases, SQL, Database APIs, Managing a MySQL Database. Accessing MySQL in PHP, Applications.				

List of Laboratory Tasks:

Experiment No. 1: Demonstration of XHTML features

Level 1: Demonstration of various XHTML Tags (Level 1)

Level 2: Design and develop static web pages for an online Book store (Level 2).

Experiment No. 2: Application of CSS in web designing

Level 1: Design a document using XHTML and CSS to create a catalog of items for online electronic shopping.

Level 2: Create and save XML document for students' information and display the same using cascaded style sheet.

Experiment No. 3: Application of PHP in web designing.

Level 1: Write a PHP program to read the personal information of a person such as first name, last name, age, permanent address, and pin code entered by the user into a table created in MySQL. Read the same information from the database and display it on the front end.

Level 2: Using PHP develop a web page that accepts book information such as ISBN number, title, authors, edition, and publisher and store information submitted through the web page in MySQL database.

Experiment No. 4: Building a website.

Build a website for organizing an International Conference. The conference website must be able to collect the author's details and upload a file.

Targeted Application & Tools that can be used:

Xampp web server to be used to demonstrate PHP.

Project work/Assignment:

Assignments are given after completion of each module which the student need to submit within the stipulated deadline.

Textbook(s):

1. Robert. W. Sebesta, "*Programming the World Wide Web*", Pearson Education, 9th Edition, 2016.
2. Paul Deitel, Harvey Deitel, Abbey Deital,"Internet & World Wide Web How to Program", Fifth Edition, Pearson Education, 2021.
3. *CSS Notes for Professionals*, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022)
4. Deitel, Deitel, Goldberg,"*Internet & World Wide Web How to Program*", Fifth Edition, Pearson
5. Education, 2021.

Reference Book(s):

R1. Randy Connolly, Ricardo Hoar,"Fundamentals of Web Development", Pearson Education India, 1st. Edition.2016.

R2. Jeffrey C. Jackson,"Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition,2016.

Additional web-based resources

W1. W3schools.com

W2. Developer.mozilla.org/en-US/docs/Learn

W3. docs.microsoft.com

W4. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking

https://presiuniv.knimbus.com/user#/home

Topics related to development of "FOUNDATION":

1. Web, WWW, Web browsers, Web servers, Internet.
2. CSS, PHP.
3. Designing the website for healthcare.

The objective of the course is to familiarize the learners with the concepts of Web Technology and attain Skill Development through Experiential Learning techniques.

Course Code: **CSE2260**	**Course Title: Database Management Systems** **Type of Course: Theory only - PCC**	**L-T-P-C**	3	0	0	3

Version No.	1.0				
Course Pre-requisites	Foundational understanding of data types, data structures, basic programming knowledge, familiarity with operating systems and file management. Basic knowledge of set theory, logic, and discrete mathematics to understand relational algebra and query formulation.				
Anti-requisites	NIL				
Course Description	This course introduces the foundational principles of database management systems, including data models, schemas, and architectures. This course provides a solid foundation on the relational model of data and the use of relational algebra. It develops skills in SQL for data definition, manipulation, and control, enabling students to construct and execute complex queries. The course also introduces the concept of object oriented and object relational databases and modern database technologies like **NoSQL.** The also course allows the students to gain insights into data storage structures and indexing strategies for optimizing query performance.				
Course Objective	The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain **Employability** through **Problem Solving** Methodologies.				
Course Out Comes	On successful completion of the course the students shall be able to: 1. Describe the fundamental elements of relational database management systems. [Understand] 2. Examine databases using SQL query processing and Optimization. [Apply] 3. Design simple database systems applying the normalization constraints and demonstrate the database transaction processing, recovery, and security. [Apply] 4. Interpret the concept of advanced databases and its applications. [Apply]				
Course Content:					
Module 1	**Introduction to Database Modelling and Relational Algebra(Understand)**	Assignment	**Problem Solving**	**10 Sessions**	
Topics: **Introduction to Database:** Schema, Instance, 3-shema architecture, physical and logical data independence, Data isolation problem in traditional file system, advantages of database over traditional file systems. Entity Relationship (ER) Model, ER Model to Relational Model, Examples on ER model. **Relational Algebra** with selection, projection, rename, set operations, Cartesian product, joins (inner and outer joins), and division operator. Examples on Relational Algebra Operations.					
Module 2	**Fundamentals of SQL and Query Optimization (Apply)**	Assignment	**Programming**	**11 Sessions**	
Topics: **SQL Database Querying,** DDL, DML, Constraints, Operators, Set Operators, Aggregate Functions, Joins, Views, Procedures, Functions and Triggers. **Database programming issues and techniques:** Embedded SQL, Dynamic SQL; SQL / PSM and NoSQL. **Query Optimization:** Purpose, transformation of relational expressions, estimating cost and statistics ofexpression, choosing evaluation plans, linear and bushy plans, dynamic programming algorithms.					
Module 3	**Relational Database Design & Transaction Management (Apply)**	Assignment	**Problem Solving**	**12 Sessions**	
Topics: **Relational database design:** Problems in schema design, redundancy and anomalies, Normal Forms based on Primary Keys-(1NF,2NF, 3NF), Boyce-Codd Normal Form, Multi valued Dependency (Fourth Normal Form), Join Dependencies (Fifth Normal Form), lossy and lossless decompositions, Database De-normalization. **Transaction Management:** The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; The write-ahead log protocol; Check pointing; Recovering from a System Crash; Media Recovery; Other approaches and interaction with concurrency control.					
Module 4	**Advanced DBMS Topics (Apply)**	**Assignment**	**Case Study**	**12 Sessions**	

Topics:

Advanced topics: Object oriented database management systems, Deductive database management systems, Spatial database management systems, Temporal database management systems,Constraint database management systems.

New database applications and architectures such as Data warehousing, Multimedia, Mobility, NoSQL, NativeXML databases (NXD), Document-oriented databases, Statistical databases.

Targeted Application & Tools that can be used:

Application Area: Relational database systems for Business, Scientific and Engineering Applications.Tools/Simulator used: MySQL DB for student practice.

Also demonstration of ORACLE DB on object-relational database creation and JDBC connection.

1. Problem Solving: Constructing ER-Diagrams for a given real time requirements, Normalizing the databases, querying the databases using relational algebra.
2. Programming: Implementation of any given scenario using MySQL.

Text Books:

T1. Elmasri R and Navathe S B, “Fundamentals of Database System”, Pearson Publication, 7th Edition, 2018.

T2. RamaKrishna & Gehrke, “Database Management Systems” 3rd Edition, 2018, McGraw-Hill Education.

T3. W. Lemahieu, S. vanden Broucke and B. Baesens, “Principles of Database Management: Practical Guide to Storing, Managing and Analyzing Big and Small Data”, Cambridge University Press, 2018.

References

R1 Avi Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, McGraw-Hill ,7th Edition, 2019.

R2 M. Kleppmann, “Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems”, O’Reilly, 2017.

Topics relevant to development of “FOUNDATION SKILLS”: S - Skill Development: Relational database design using ER- Relational mapping, Implementation of given database scenario using MYSQLDB.

Topics relevant to development of Employability: Develop, test and implement computer databases, creating sophisticated, interactive and secure database applications

Topics relevant to “HUMAN VALUES &PROFESSIONAL ETHICS”: Nil

Course Code: CSE2261	**Course Title: Database Management Systems Laboratory** **Type of Course: 1) Laboratory - PCC**	**L-T-P-C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Foundational understanding of data types, basic programming knowledge, operating systems and file management.					
Anti-requisites	NIL					
Course Description	The Database Management Systems (DBMS) Laboratory is designed to provide students with hands-on experience in database design, implementation, and management using SQL and database management tools such as MySQL. The lab complements theoretical concepts learned in database courses by allowing students to practice database creation, querying, and optimization techniques. The DBMS Lab enables students to develop industry-relevant skills in database management, preparing them for careers in **software development, data engineering, and database administration**.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain **Employability** through **Problem Solving** Methodologies.					
Course Out Comes	On successful completion of the course the students shall be able to: 5. Demonstrate the database concepts, practice, and SQL queries. [Apply] 6. Design and implement database schemas while applying normalization techniques to optimize structure. [Apply]] 7. Develop and implement stored procedures, triggers, and views for automation and efficiency. [Apply] 8. To Design and build database applications for real world problems. [Apply]					
Course Content:						

List of Laboratory Tasks:
Create Employee, Student, Banking and Library databases and populate them with required data. Do the following experiments of different lab sheets on those databases.

Labsheet-1 [3 Practical Sessions]
Experiment No 1: [1 Session]
1. To study and implement the different language of Structured Query Language.
Level 1: Perform operations using Data Definition Language and Data Manipulation Language commands including different variants of SELECT on Student DB.
Level 2: Identify the given requirements; valid attributes and data types and Perform DDL and DML operations on a given scenario. [Banking Databases]
Experiment No. 2: [2 Sessions]
2. To study and implement the concept of integrity constraints in SQL.
Level 1: Create tables on Banking database using PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY and demonstrate the working of relational, logical, pattern matching, BETWEEN, IS NULL, IN and NOT IN Special Operators on Student Database.
Level 2: Enforce different types of data and referential integrity constraints. Then try queries with special operators based on the student database. [Banking Database].

Labsheet-2 [3 Practical Sessions]
Experiment No. 3: [1 Session]
3. Implement complex queries in SQL.
Level 1: Implement the conjugate of GROUP BY, ORDER BY and aggregate functions on Banking Database. Level 2: Implement MySQL DB queries on library database using appropriate clauses and aggregate functions. Also order the data either in ascending and descending order using corresponding clause. [Library databases].

Experiment No. 4: [2 Session]
4. To study and implement different types of Set and Join Operations [2 Slots]
Level 1: Demonstrate different types of Set Operations (UNION, UNION ALL, INTERSECT, MINUS) and Join Operations (INNER JOINs, OUTER JOINS, CROSS JOIN, NATURAL JOIN) on two or more tables of Airline Database. Level 2: Use Set and Join operations to retrieve the data from two or more relations(tables) as per the given scenario. [Airline Database]

Labsheet-3 [2 Practical Sessions]
Experiment No. 5: [2 sessions]
5. To study and implement Views, and Procedures in MySQL DB.
Level 1: Implement MySQL Views, and Procedures in ORACLE DB on Employee database.
Level 2: Analyze the requirement and construct views, and Procedures on Mini Project Domain. [Banking Database]

Labsheet-4 [2 Practical Sessions]
Experiment No. 6: [2 Sessions]
6. To study and implement Functions, and Triggers in MySQL DB.
Level 1: Implement Oracle Functions and Triggers in Oracle on Employee database.
Level 2: Analyze the requirement and construct Functions and Triggers. [Supply chain Database]

Labsheet-5 [2 Practical Sessions]
Experiment No. 7: [2 Sessions]
7. To study and implement Functions, and Triggers in MySQL DB.
Level 1: Implement Oracle Functions and Triggers in Oracle on Employee database.
Level 2: Determine the requirement and construct Functions and Triggers. [Supply chain Database]

Labsheet-6 [4 Practical Sessions]
Experiment No. 8: [2 Sessions]
8. To implement the concept of forms and reports.
Level 1: Implement the concept of forms and reports.
Level 2: Examine the schema relationship.

Experiment No. 9: [2 Sessions]
9. Create the database using the given schema. (Flight Management)
Level 1: Implement a relational database based on the provided schema for the Flight Management system, including the creation of tables, relationships, and constraints.
Level 2: Demonstrate schema relationships by defining primary and foreign keys to ensure data integrity within the Flight Management database.

Labsheet-7 [4 Practical Sessions]
Experiment No. 10: [2 Sessions]
10. Create the database using the given schema. (Company database)
Level 1: Implement the database schema by defining tables, relationships, and constraints according to the given Company Database schema.
Level 2: Demonstrate the schema's relationships and data integrity by creating and linking tables as per the specified requirements.
Experiment No. 11: [2 Sessions]
11. Create the database using the given schema. (Student Library)
Level 1: Implement forms and reports based on the provided Student Library database schema, ensuring effective data entry and reporting mechanisms.
Level 2: Demonstrate the schema relationships within the Student Library database, demonstrating how these relationships influence the creation and functionality of forms and reports.

Labsheet-8 [1 Sessions]
12. Design a mini project based on the databases such as Inventory Management System, University Management System, Hospital Management System, etc.
Level 1: Implement the real time database.
Level 2: Analyze the working of database in real time.

Targeted Application & Tools that can be used:
Application Area: Relational database systems for Business, Scientific and Engineering Applications.Tools/Simulator used: MySQL DB for student practice.
Also demonstration of ORACLE DB on object-relational database creation and JDBC connection.

Percentage of changes in this version: 50% of changes from earlier version. New topics are highlighted initalic.

1. Problem Solving: Constructing ER-Diagrams for a given real time requirements, Normalizing the databases, querying the databases using relational algebra.
2. Programming: Implementation of any given scenario using MySQL.

Text Books:

T1. Elmasri R and Navathe S B, "Fundamentals of Database System", Pearson Publication, 7th Edition, 2018.

T2. RamaKrishna & Gehrke, "Database Management Systems" 3rd Edition, 2018, McGraw-Hill Education.

T3. W. Lemahieu, S. vanden Broucke and B. Baesens, "Principles of Database Management: Practical Guide to Storing, Managing and Analyzing Big and Small Data", Cambridge University Press, 2018.

References

R1 Avi Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill ,7th Edition, 2019.

R2 M. Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems", O'Reilly, 2017.

Topics relevant to development of "FOUNDATION SKILLS": S - Skill Development: Relational database design using ER- Relational mapping, Implementation of given database scenario using MYSQLDB.
Topics relevant to development of Employability: Develop, test and implement computer databases, creating sophisticated, interactive and secure database applications
Topics relevant to "HUMAN VALUES &PROFESSIONAL ETHICS": Nil

<table>
<tr><td>Course Code:
CSE2500</td><td colspan="3">Course Title: Data Analytics
Type of Course: Theory - PCC</td><td>L-T-P- C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">MAT2402</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">Fundamentals of Data Analytics is designed for inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, and supports in decision-making. The course begins by covering Data extraction, pre-processing, and transformation. It delivers the basic statistics and taught in an intuitive way to analysis the data. This course will help the students to apply the knowledge on data analysis to a wide range of applications.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Fundamentals of Data Analytics and attain SKILL DEVELOPMENT through PROBLEM SOLVING Methodologies.</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of this course, the students shall be able to:
CO1:Describe different types of data and variables.
CO2: Explain data using appropriate statistical methods.
CO3: Demonstrate the collection, processing and analysis of data for any given application and illustrate various charts using visualization methods.
CO4: Apply the Data Analysis techniques by R Programming</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction to Data Analysis- CO1</td><td>Assignment</td><td colspan="2">Data Collection, data analysis, Programming</td><td colspan="4">06 classes</td></tr>
<tr><td colspan="9">Topics: Introducing Data, overview of data analysis: Data in the Real World, Data vs. Information, The Many "Vs" of Data, Structured Data and Unstructured Data, Types of Data, Data Analysis Defined, Types of Variables, Central Tendency of Data, Scales of Data, Sources of Data. Data preparation.

R Studio: Base R-R Studio IDE-Introduction to R Projects and R Markdown. Basic R: R as a calculator-Scripts and Comments-R Variables. Data I/O: Working Directories-Importing Data Exporting Data-More ways to save-Data I/O in Base R.</td></tr>
<tr><td>Module 2</td><td>Data Analysis and Visualization-CO2</td><td>Case studies</td><td colspan="2">Programming</td><td colspan="4">10 classes</td></tr>
<tr><td colspan="9">Topics: Data Summarization: One Quantitative and Categorical Variable. Data Classes: One Dimensional Data Classes-Data Frames and Matrices-Lists. Data Cleaning: Dealing with Missing Data-Strings and Recoding Variables. Manipulating Data in R: Reshaping Data-Merging Datasets. Data Visualizations: Plotting with ggplot2- Plotting with Base R</td></tr>
<tr><td>Module 3</td><td>Statistical Analysis -CO3</td><td>Case studies</td><td colspan="2">R programming</td><td colspan="4">7 classes</td></tr>
<tr><td colspan="9">Topics: Proportion tests-Chi squared test-Fisher exact test-Correlation-T test-Wilcoxon Rank sum tests-Wilcoxon signed rank test- one-way ANOVA test- Kruskal Wallis test</td></tr>
<tr><td>Module 4</td><td>Predictive Analysis-CO4</td><td>Case studies</td><td colspan="2">Programming</td><td colspan="4">7 classes</td></tr>
<tr><td colspan="9">Topics: Linear least-squares – implementation – the goodness of fit – testing a linear model – weighted resampling. Regression using Stats models – multiple regression – nonlinear relationships – logistic regression – estimating parameters – accuracy. Time series analysis – moving averages – missing values – serial correlation – autocorrelation. Introduction to survival analysis</td></tr>
</table>

Targeted Application & Tools that can be used:

Application Area are Decision making in business, health care, financial sector, Medical diagnosis etc.

Text Books

1. Glenn J. Myatt and Wayne P. Johnson, "Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining Paperback", Import, 22 July 2014.
2. Introduction to statistics and Data analytics, Christian H, Michael S, Springer,2016
3. Introduction to R- Robert Parker, John Mushcelli and Andrew Jaffe, Johns Hopkins University, 2020 (E-resource)
4. Introduction to Time Series and Forecasting (Springer Texts in Statistics), Peter Brockwell, Richard A. Davis, Springer, 2016.

References

1. Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining Paperback, Glenn J. Myatt and Wayne P. Johnson, Import, 22 July 2014.
2. The R Software-Fundamentals of Programming and Statistical Analysis -Pierre Lafaye de Micheaux, Remy Drouilhet, Benoit Liquet, Springer 2013.

Online resources:

http://www.modernstatisticswithr.com/solutions.html#solutionsch3
https://johnmuschelli.com/intro_to_r/
https://users.phhp.ufl.edu/rlp176/Courses/PHC6089/R_notes/

Topics relevant to development of "FOUNDATION SKILLS":

1. Statistical Concepts for data, visualization techniques.
2. Data collection for project based assignments.
3. Inferential Statistics (T test, Z test)
4. Probability Calculation

for Skill Development through Problem Solving methodologies. This is attained through assessment component mentioned in course handout.

Course Code: CSE2501	Course Title: Data Analytics Lab Type of Course: Lab - PCC	L-T-P- C	0	0	2	1
Version No.	1.0					
Course Pre-requisites	MAT2402					
Anti-requisites	NIL					
Course Description	**Data Analytics is designed for inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, and supports in decision-making. The course begins by covering Data extraction, pre-processing, and transformation. It delivers the basic statistics and taught in an intuitive way to analysis the data. This course will help the students to apply the knowledge on data analysis to a wide range of applications.**					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Fundamentals of Data Analytics** and attain **SKILL DEVELOPMENT** through **PROBLEM SOLVING** Methodologies.					
Course Out Comes	**On successful completion of this course, the students shall be able to:** **CO1**:Describe different types of data and variables. **CO2**: Explain data using appropriate statistical methods. **CO3**: Demonstrate the collection, processing and analysis of data for any given application and illustrate various charts using visualization methods. **CO4**: Apply the Data Analysis techniques by R Programming					
Course Content:						

Module 1	Introduction to Data Analysis- CO1	Assignment	Programming	09 classes

List of Laboratory Tasks:

Experiment No. 1: Introduction to R and RStudio

Level 1: Getting Started with R and RStudio

- Installing R and RStudio.
- Basic R syntax and commands.

Level 2: Working with RStudio

- Understanding the RStudio interface.
- Creating and managing R scripts.

Experiment No. 2: Basic Data Handling in R

Level 1: Data Types and Structures in R

- Vectors, matrices, and data frames.
- Lists and factors.

Level 2: Data Import and Export

- Reading data from CSV, Excel, and text files.
- Exporting data to different formats.

Level 3: Exploring Datasets

- Using functions like head(), summary(), and str().

Experiment No. 3: Basic Data structure in R

Level 1:

a. Demonstrate a program to join columns and rows in a data frame using cbind() and rbind() in R.

b. Implement different data structures in R (Vectors, Lists, Data Frames)

Level 2: R AS CALCULATOR APPLICATION a. Using with and without R objects on console

a. Using mathematical functions on console

b. Write an R script, to create R objects for the calculator application

Module 2	**Data Analysis and Visualization-CO2**	Assignment	Programming	**13 classes**

Experiment No. 1: Data Cleaning and Preprocessing

Level 1: Handling Missing Data in R

- Identifying missing values.
- Imputing missing values using mean, median, or other methods.

Level 2: Data Transformation in R

- Standardizing and normalizing data.
- Log-transformations and scaling.

Experiment No. 2: Exploratory Data Analysis (EDA) with R

Level 1: Descriptive Statistics

- Calculating mean, median, and standard deviation.
- Visualizing data using histograms, box plots, and scatter plots.

Experiment No. 3: Data Visualization with ggplot2

Level 1: Demonstrate various graphs that can be made and altered using the ggplot2 package.

Level 2: Create 500 random temperature readings for six cities over a season and then plot the generated data using ggplot2 packages in R

Module 3	**Statistical Analysis -CO3**	Assignment	programming	**10 classes**

Experiment No. 1: Perform Tests of Hypotheses hypothesis test (parametric)

Level 1: How to perform tests of hypotheses about the mean when the variance is known. How to compute the p-value. Explore the connection between the critical region, the test statistic, and the p-value.

Level 2: A teacher claims that people who work for only five hours per week will score significantly lower than people who work for ten hours per week on a quantitative abilities test. He brings twenty people and randomly assigned them to one or two groups. In one group he has participants who work for ten hours and in another group, he has participants who work for five hours. He conducts the test for all participants. Scores on the test range from one to ten with higher scores representing better performance. Test if there is any significant difference between those who work for five hours per week versus those who work for ten hours per week based on the test performance.

Experiment No 2: Hypothesis – Non-Parametric Test

Level 1: A car manufacturing company like to find the sales of three types of cars produced by them in three regions and is given. Test if there is an association between the regions and types of cars purchased.

Experiment No 3: Correlation and Covariance

Level 1: Using the iris data set in R

a. Find the correlation matrix.
b. Plot the correlation plot on dataset and visualize giving an overview of relationships among data on iris data.
c. Analysis of covariance: variance (ANOVA), if data have categorical variables on iris data.

Level 2 : Ramesh is doing a statistics paper in his post-graduation course. He met his friend Amal who is a textile engineer. Ramesh, who is doing his internship at ABC Researchers, is interested in a question. He poses this question to Amal and tries to find if he can answer. The question is as follows: The data regarding sales of soft- drinks and sales of cotton clothes in a place during the last 12 months are given. Find if there is any association between sales of soft drinks and sales of cotton clothes. Also explain the reason if there is any relationship.

Module 4	Predictive Analysis-CO4	Assignment	Programming	**10 classes**

Experiment No 1: Regression Model

Level 1: Import data from web storage (http://www.ats.ucla.edu/stat/data/binary.csv). Name the dataset and now do Logistic Regression to find out the relation between variables that are affecting the admission of a student in an institute based on his or her GRE score, GPA obtained, and rank of the student. Also check the model is fit or not. Require (foreign), require (MASS).

Level 2: Demonstrate multiple regressions, if data have a continuous Independent variable. Apply on the above dataset

Experiment No. 2: Time Series Analysis in R

Level 1: Demonstrate Timeseries analysis using Time Series Data Library at http://robjhyndman.com/TSDL/.

Targeted Application & Tools that can be used:

Application Area are Decision making in business, health care, financial sector, Medical diagnosis etc.

Text Books

5. Glenn J. Myatt and Wayne P. Johnson, "Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining Paperback", Import, 22 July 2014.
6. Introduction to statistics and Data analytics, Christian H, Michael S, Springer,2016
7. Introduction to R- Robert Parker, John Muschelli and Andrew Jaffe, Johns Hopkins University, 2020 (E-resource)
8. Introduction to Time Series and Forecasting (Springer Texts in Statistics), Peter Brockwell, Richard A. Davis, Springer, 2016.

References

3. Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining Paperback, Glenn J. Myatt and Wayne P. Johnson, Import, 22 July 2014.
4. The R Software-Fundamentals of Programming and Statistical Analysis -Pierre Lafaye de Micheaux, Remy Drouilhet, Benoit Liquet, Springer 2013.

Online resources:

http://www.modernstatisticswithr.com/solutions.html#solutionsch3
https://johnmuschelli.com/intro_to_r/
https://users.phhp.ufl.edu/rlp176/Courses/PHC6089/R_notes/

Topics relevant to development of "FOUNDATION SKILLS":

2. Statistical Concepts for data, visualization techniques.
3. Data collection for project based assignments.
4. Inferential Statistics (T test, Z test)

5. Probability Calculation

for Skill Development through Problem Solving methodologies. This is attained through assessment component mentioned in course handout.

urse Code: E2262	urse Title: **Analysis of Algorithms** pe of Course: **Theory PCC**			T-P- C	3	1	0	4
rsion No.								
urse Pre-requisites								
ti-requisites								
urse Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of applications. This course discusses the classic approaches for algorithm design such as Divide and Conquer, Dynamic Programming, Greedy method. This course also describes other basic strategies searching solution space. The core concepts of analyzing algorithms and classifying them into various complexity classes is covered in the end.							
urse Objective	The objective of the course is to familiarize the learners with the concepts of **Analysis of Algorithms** and attain **Skill Development** through **Problem Solving** Methodologies.							
urse Out Comes	On successful completion of the course the students shall be able to: 1. Compute efficiency of a given algorithm.[Apply] 2. Apply divide and conquer technique for searching and sorting Problems.[Apply] 3. Apply the Dynamic Programming technique for a given problem. [Apply] 4. Apply greedy technique for solving a Problem.[Apply] 5. Demonstrate Back tracking technique and limitations of Algorithms.[Apply]							
urse Content:								
dule 1	**roduction**	signment	hulation/Data Analysis	**10 Sessions**				
Introduction, Asymptotic Notations and its properties, Best case, worst case and average case- Sequential search, Sorting; Mathematical analysis for Recursive and Non-recursive algorithms: Substitution method and Master's Theorem.								
dule 2	**vide-and-conquer**	signment	hulation/Data Analysis	**08 Sessions**				
roduction. Insertion Sort; Merge sort, Quick sort, Binary search.								
dule 3	**namic programming**	m paper/Assignment	hulation/Data Analysis	**10 Sessions**				
roduction with examples, Principles of Memoization, 0-1 Knapsack Problem, Bellman-Ford algorithm, Floyd-Warshall's Algorithms. Chain Matrix Multiplication.								
dule 4	**eedy technique**	m paper/Assignment	hulation/Data Analysis	**09 Sessions**				
Introduction, Fractional Knapsack Problem, Minimal Spanning Tree: Prim's Algorithm and Kruskal's Algorithm, Single-source Shortest Path: Dijkstra's Algorithm								
dule 5	**mplexity Classes**	m paper/Assignment	hulation/Data Analysis	**08 Sessions**				
Complexity Classes- P,NP- NP Hard and NP Complete - Boolean Satisfiability Problem (SAT). nch and Bound: Knapsack problem; Backtracking, - N-Queens problem.								

xt Book

1. Anany Levitin, *"Introduction to the Design and Analysis of Algorithms*", 3rd edition, Pearson Education, 2018.
2. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, *"Introduction to Algorithms*", 4th edition, MIT Press, 2022.

ferences

1. J. Kleinberg and E. Tardos, "*Algorithm Design*", Addison-Wesley, 2005.
2. Tim Roughgarden, *"Algorithms Illuminated"* (books 1 through 3), "Operating Systems Design and Implementation", Soundlikeyourself Publishing, 2017-2019.
3. AV Aho, J Hopcroft, JD Ullman, *"The Design and Analysis of Algorithms"*, Addison-Wesley, 1974.
4. Donald E. Knuth, "*The Art of Computer Programming",* Volumes 1and 3 Pearson.

eb-Resources

1. NPTEL: https://onlinecourses.nptel.ac.in/noc19_cs47/preview
2. Coursera: Analysis of Algorithms by Princeton University
3. Algorithms Specialization in Coursera by Stanford University(Group of 4 courses).
4. Algorithms Coding Contest Links maintained by Prof Gerth Stølting Brodal of Aarhus University

pics relevant to "SKILL DEVELOPMENT": knapsack, prim's, kruskal's algorithm, quick sort, binary search for **Skill Development** through **Problem Solving methodologies**. This is attained through assessment component mentioned in course handout.

Course Code: **CSE2263**	**Course Title: Analysis of Algorithms Lab** **Type of Course: Lab - PCC**	**L- T-P-C**	0	0	2	1
Version No.	1					
Course Pre-requisites	**Nil**					
Anti-requisites	NIL					
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of applications. This course discusses the classic approaches for algorithm design such as Divide and Conquer, Dynamic Programming, Greedy method. This course also describes other basic strategies searching solution space. The core concepts of analyzing algorithms and classifying them into various complexity classes is covered in the end.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Analysis of Algorithms** and attain **Skill Development** through **Experiential Learning** Methodologies.					
Course Out Comes	**On successful completion of the course the students shall be able to:** 1. Compute efficiency of a given algorithm. [Applying] 2. Apply divide and conquer technique for searching and sorting Problems.[Applying] 3. Apply the Dynamic Programming technique for a given problem. [Applying] 4. Apply greedy technique for solving a Problem.[Applying] 5. Demonstrate Back tracking technique and limitations of Algorithms.[Applying]					
Course Content						

Module 1	**Introduction**	**3 Sessions**
Measuring running time of an algorithm, Compare running time of algorithms, Implement sorting algorithms such as bubble sort, selection sort		
Module 2	**Divide-and-conquer**	**3 Sessions**
Compare searching algorithms: Linear Search, Binary Search; Compare Sorting algorithms: Insertion Sort, Merge Sort, QuickSort.		
Module 3	**Dynamic programming**	**3 Sessions**
Introduction and memorization: Factorial; Coin Change Problem ; Floyd-Warshall's Algorithm.		
Module 4	**Greedy technique**	**3 Sessions**
Fractional Knapsack Problem; Minimal Spanning Tree Algorithms-Prim's Algorithm, Kruskal's algorithm		
Module 5	**Complexity Classes**	**3 Sessions**
Branch and Bound: Knapsack problem; Backtracking, - N-Queens problem.		

List of Laboratory Tasks:

1. Measuring running time of an algorithm

Objective: To experimentally determine the running time of basic algorithms for input size n=10, 100, 1000, etc. by taking difference of starting time and ending time.

2. Compare running time of algorithms

Objective: To execute two algorithms to solve the same problem, and to comparatively evaluate the better algorithm for large values of N.

3. Implement sorting algorithms such as bubble sort, selection sort

Objective: To implement comparison based sorting strategies.

4. Compare searching algorithms

Objective: To implement two searching strategies and compare their performance.

5. Compare Sorting algorithms

Objective: To implement searching strategies that follow top down design approach(Insertion sort, merge sort).

6. Quick Sort

Objective: To demonstrate Quick sort and its variants, and their impact on running time.

7. Dynamic Programming

Objective: To demonstrate Dynamic Programming approach with the help of Factorial algorithm.

8. Coin Change Problem

Objective: To implement an efficient algorithm for the Coin Change problem.

9. Floyd-Warshall's Algorithm

Objective: To demonstrate how dynamic programming is used with the help of Floyd-Warshall's algorithm.

10. Fractional Knapsack Problem

Objective: To demonstrate how greedy method can be used to solve the Fractional Knapsack Problem.

11. Minimal Spanning Tree Algorithm

Objective: To implement greedy strategy to solve the Minimal Spanning Tree problem using Prim's Algorithm.

12. Kruskal's Minimal Spanning Tree Algorithm

Objective: To implement greedy strategies to solve the Minimal Spanning Tree problem using Kruskal's Algorithm.

13. Knapsack Problem

Objective: To implement Knapsack problem using branch and bound technique.

14. N-Queen's Problem

Objective: To demonstrate backtracking method with the help of N-Queen's problem.

15. Case Study

Objective: To demonstrate how various techniques can be used to solve the same problem with the help of Knapsack problem.

Targeted Application & Tools that can be used

1. PyTorch/Jupyter Notebook – For Python programming

Text Book

T1 Anany Levitin, *"Introduction to the Design and Analysis of Algorithms*", 3rd edition, Pearson Education, 2018.

T2 Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, *"Introduction to Algorithms*", 4th edition, MIT Press, 2022.

References

R1. J. Kleinberg and E. Tardos, "*Algorithm Design*", Addison-Wesley, 2005.

R2. Tim Roughgarden, *"Algorithms Illuminated"* (books 1 through 3), "Operating Systems Design and Implementation", Soundlikeyourself Publishing, 2017-2019.

R3. AV Aho, J Hopcroft, JD Ullman, *"The Design and Analysis of Algorithms"*, Addison-Wesley, 1974.

R4. Donald E. Knuth, "*The Art of Computer Programming",* Volumes 1and 3 Pearson.

Web Based Resources and E-books:

W1. NPTEL: https://onlinecourses.nptel.ac.in/noc19_cs47/preview

W2. Coursera: Analysis of Algorithms by Princeton University W3. Algorithms Specialization in Coursera by Stanford University(Group of 4 courses). W4. Algorithms Coding Contest Links maintained by Prof Gerth Stølting Brodal of Aarhus University
Topics relevant to "EMPLOYABILITY SKILLS": The lab experiments and assessments enable the student to acquire Skill Development through Experiential Learning techniques

urse Code: CSE2264	Course Title: Essentials of Artificial Intelligence Type of Course: Program Core Course -Theory	T-P-C	3			3
rsion No.						
urse Pre-requisites	NIL					
ti-requisites						
urse Description	This course introduces the student to the basics of artificial intelligence. In this course, the student first learns the various search methods for problem-solving, followed by knowledge-based logic representations. After that, the student will learn about uncertainty in AI, as well as approaches to solve such challenges such as Naïve Bayes Classifier and Hidden Markov Models. Topics: Uninformed search, Heuristic search, Local search, Adversarial search, Constraint satisfaction, logic, First Order Resolution, Probability, Naïve Bayes Classifier, and Hidden Markov Model (HMM).					
urse Objectives	The objective of the course is EMPLOYBILITY of student by using EXPERIENTIAL LEARNING techniques.					
urse Out Comes	On successful completion of this course the students shall be able to: 1. Explain different methods of searching, proving, and analysis in AI [Understand] 2. Implement various graphical and adversarial search algorithms. [Apply] 3. Prove, by resolution, different situations using First Order Logic [Apply] 4. Solve sequence labeling problems using HMM [Apply]					

Course Content:

dule 1	rch Methods for Problem-Solving	blem-Solving Tests	TEL Assignments	No. of Sessions: 13
Introduction – History of AI, Agents and Environment, Types of AI and Learning. State Space Search; General Formulation of Search Problems; Data Structures used in Searching. **Uninformed Search Algorithms** – Breadth First Search, Depth First Search, Uniform Cost Search, Generalized Uniform Cost Search (a.k.a Dijkstra's Single-Source Shortest Path), Iterative Deepening Depth-First Search, Time and Space Complexity Analysis of Uninformed Search Algorithms. **Heuristic Search Algorithms** – Heuristics and Admissibility, Greedy Best-First Search, A* Search and weighted A* Search.				
dule 2	vanced Search Methods	blem-Solving Tests	TEL Assignments	No. of Sessions: 12
Local Search – Local Search, Hill Climbing, Genetic Algorithms, Gradient Descent. **Adversarial Search** – Minimax Search, Alpha-Beta Pruning, Ideal Ordering. **Constraint Satisfaction** – Constraint Satisfaction Problems Definitions and Examples – Map Colouring, N Queens, Cryptarithmetic, Generalized CSP; Back-tracking Heuristics; Arc Consistency and Path Consistency				
dule 3	owledge-Based Logic Representation	tomated Theorem Proving using FOL Resolution	TEL Assignments	No. of Sessions: 10
Propositional Logic – Syntax and Semantics of Propositional Logic. Logical connectives. Inference Rules. Conjunctive and Disjunctive Normal Forms. **First Order Logic** – Syntax and Semantics of Propositional Logic. Logical connectives. Inference Rules. Conjunctive and Disjunctive Normal Forms. **Resolution** – Resolution Principle. Propositional and First Order Resolution. Applications for solving story problems using Resolution				
dule 4	certainty in AI	presenting problems as HMM	TEL Assignments	No. of Sessions: 06
Probability – Probability Definitions. Conditional Probability. Bayes Theorem. Naïve Bayes Classifier. Using Naïve Bayes Classifier for Supervised Learning. **Hidden Markov Models** – Definition of HMM. Sequence Labeling and Markov Assumption. Sub-Problems in HMM and their solutions – Forward				

Probability and Viterbi Algorithm. Applications of Sequence Labeling in Natural Language Processing (Eg. Part-of-Speech Tagging). **Introduction to Deep Learning** – Artificial Neurons, Activation Functions, Multilayer Perceptron.

Targeted Application & Tools that can be used:

1. Implementation of a shortest-path finder using different search algorithms.
2. Implementation of a sequence labeler using Viterbi Algorithm.

oject work/Assignment: Mention the Type of Project /Assignment proposed for this course

1. Group project on one of the topics mentioned above (Eg. Adversarial search).

xtbook(s):

1. Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th Edition. Pearson Education. 2022.
2. Lavika Goel. *Artificial Intelligence: Concepts and Applications*. 1st Edition. Wiley. 2021.
3. Elaine Rich, Kevin Knight and Shivashankar B Nair. *Artificial Intelligence*. 4th Edition. MedTech Science Press. 2024.

ferences:

1. Deepak Khemani. *A First Course in Artificial Intelligence*. 1st Edition. 6th Reprint, 2018.
2. Munesh Chandra Trivedi. *A Classical Approach to Artificial Intelligence*. 2nd Edition. Khanna Publishers. 2018.
3. George Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. 6th Edition. Pearson Education. 2021.

blinks

1. NPTEL Courses: Mausam (IIT Delhi), "An Introduction to Artificial Intelligence" Link: https://nptel.ac.in/courses/106102220.
2. Shyamanta M. Hazarika (IIT Guwahati), "Fundamentals of Artificial Intelligence". Link: https://nptel.ac.in/courses/112103280. Useful for the full course.
3. Deepak Khemani (IIT Madras), "Artificial Intelligence: Search Methods for Problem-Solving". Link: https://nptel.ac.in/courses/106106226. Useful for Module 1 and 2
4. Deepak Khemani (IIT Madras), "Artificial Intelligence: Knowledge Representation and Reasoning".
   Link: https://nptel.ac.in/courses/106106140. Useful for Module 3.
5. Deepak Khemani (IIT Madras), "AI: Constraint Satisfaction". Link: https://nptel.ac.in/courses/106106158. Useful for Module 2.

urse Code: CSE2265	**Course Title:** **Essentials of AI Lab** **pe of Course:** **Program Core Course - Lab**	**T-P-C**	0			1
rsion No.						
urse Pre-requisites	NIL					
ti-requisites						
urse Description	This course introduces the student to the basics of artificial intelligence. In this course, the student first learns the various search methods for problem-solving, followed by knowledge-based logic representations. After that, the student will learn about uncertainty in AI, as well as approaches to solve such challenges such as Naïve Bayes Classifier and Hidden Markov Models. Topics: Uninformed search, Heuristic search, Local search, Adversarial search, Constraint satisfaction, logic, First Order Resolution, Probability, Naïve Bayes Classifier, and Hidden Markov Model (HMM).					
urse Objectives	The objective of the course is EMPLOYBILITY of student by using EXPERIENTIAL LEARNING techniques.					
urse OutComes	On successful completion of this course the students shall be able to: 1. Explain different methods of searching, proving, and analysis in AI [Understand] 2. Implement various graphical and adversarial search algorithms. [Apply] 3. Prove, by resolution, different situations using First Order Logic [Apply] 4. Solve sequence labeling problems using HMM [Apply]					

Course Content: **No. of Sessions: 15 (30 hours)**

Experiment No. 1: File Handling
Level 1: Read text files using Python
Level 2: Parse text files using Python

Experiment No. 2: Implementation of Graph Representations
Level 1: Implement graph representations by taking input from the console
Level 2: Implement graph representations by taking input from files.

Experiment No. 3 & 4: Implementation of Uninformed Search Algorithms
Level 1: Implement uninformed search algorithms – BFS and DFS – on unweighted graphs.
Level 2: Implement uninformed search algorithms – Uniform Cost Search and Dijkstra's SSSP – on weighted graphs

Experiment No. 5: Implementation of Heuristic Search Algorithms
Level 1: Calculate the upper-bounds of admissible heuristics using Dijkstra's SSSP.
Level 2: Implement Greedy Best-First Search and A* Search Algorithms.

Experiment No. 6 & 7: Implementation of Adversarial Search
Level 1: Implement a Game Tree
Level 2: Perform Alpha-Beta Pruning and Ideal Ordering

Experiment No. 8 & 9: Implementation of a CSP Solver
Level 1: Implement a CSP solver to solve a cryptarithmetic problem
Level 2: Implement a CSP solver for map colouring

Experiment No. 10: Using Python Packages for CSP
Level 1: Implement a CSP solver for Sudoku
Level 2: Implement a CSP solver for Addoku

Experiment No. 11: Implement a Family Tree Parser
Level 1: Perform logic programming using logpy.
Level 2: Implement a family tree parser

Experiment No. 12 & 13: Implement a Decision Maker
Level 1: Implement a Minesweeper solver
Level 2: Implement a Battleship solver

Experiment No. 14 & 15: Hidden Markov Model
Level 1: Implement a generic HMM
Level 2: Build a PoS Tagger using a HMM with the Brown Corpus and the Universal Dependencies Tagset.

Targeted Application & Tools that can be used:

3. Google Colab
4. Python IDEs like PyCharm

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

The course is a lab-based course with all the assessments centrally evaluated. Every experiment consists of **two sessions**. The first session involves exploring a solution to the problem. The second session involves solving a particular problem.

xtbook(s):

1. Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th Edition. Pearson Education. 2022.
2. Prateek Joshi and Alberto Artasanchez. *Artificial Intelligence with Python*. 2nd Edition. Packt. 2020.

ferences:

1. Deepak Khemani. *A First Course in Artificial Intelligence*. 1st Edition. 6th Reprint, 2018.
2. Munesh Chandra Trivedi. *A Classical Approach to Artificial Intelligence*. 2nd Edition. Khanna Publishers. 2018.

<table>
<tr><td>Course Code:
APT4004</td><td colspan="2">Course Title: Aptitude Training-Intermediate
Type of Course: Practical Only Course</td><td>L- T - P- C</td><td>0</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>Version No.</td><td colspan="7"></td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">Students should have the basic concepts of Quantitative aptitude along with its applications in real life problems.</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">This is a skill-based training program for the students. This course is designed to enable the students to enhance their skills in Quantitative Aptitude.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is to familiarize the learners with the concepts of Aptitude and attain Skill Development through Problem Solving techniques.</td></tr>
<tr><td>Course Out Comes</td><td colspan="7">On successful completion of this course the students shall be able to:
CO1: Recall all the basic mathematical concepts.
CO2: Identify the principle concept needed in a question.
CO3: Solve the quantitative and logical ability questions with the appropriate concept.
CO4: Analyze the data given in complex problems.</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Quantitative Ability 1</td><td colspan="2">Assignment</td><td colspan="4">16 Hours</td></tr>
<tr><td colspan="8">Topics:
Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss</td></tr>
<tr><td>Module 2</td><td>Quantitative Ability 2</td><td colspan="2">Assignment</td><td colspan="4">14 Hours</td></tr>
<tr><td colspan="8">Topics:
Time Speed and Distance, Boats and Streams, Simple Interest, Compound Interest, Probability, Permutation and Combination</td></tr>
<tr><td colspan="8">Targeted Application & Tools that can be used:
Application area: Placement activities and Competitive examinations. Tools: LMS</td></tr>
<tr><td colspan="8">Continuous Evaluation:</td></tr>
<tr><td colspan="8">CA1 – Online Test CA2 – Online Test CA3 – Online Test Assignment</td></tr>
<tr><td colspan="8">Text Book:

1.Fast Track Objective by Rajesh Verma
2. R S Aggarwal
3. Rakesh Yadav

References:

1. www.indiabix.com
2. www.testbook.com
3. www.youtube.com/c/TheAptitudeGuy/videos</td></tr>
<tr><td colspan="8">Topics relevant to Skill Development: Quantitative aptitude for Skill Development through Problem solving Techniques. This is attained through components mentioned in course handout.</td></tr>
</table>

Course Code:	**Course Title:** Theory of Computation	**L- T-P- C**	3	**0**	0	**3**

CSE2266	**Type of Course:** Theory Only					
Version No.	2.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	The course deals with introduction of formal languages and the correspondence between language classes and the automata that recognize them. Topics include: Formal definitions of grammars and acceptors, Deterministic and Nondeterministic systems, Grammar ambiguity, finite state and push-down automata; normal forms; Turing machines and its relations with algorithms.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Theory of Computation as mentioned above and attain Skill Development through Problem Solving Methodologies.					
Course Outcomes	On successful completion of the course the students shall be able to: 1. Describe various components of Automata. (Knowledge) 2. Illustrate Finite Automata for the given Language. (Application) 3. Distinguish between Regular grammar and Context free grammar. (Comprehension) 4. Construct Push down Automata. (Application) 5. Construct Turing machine for a Language. (Application)					
Course Content:						
Module 1	Introduction to automata theory	Assignment	Problems on Strings and Language operations	**6 classes**		
Topics: Introduction to Automata Theory, Applications of Automata Theory, Alphabets, Strings, Languages & operations on languages, Representation of automata, Language recognizers, Finite State Machines (FSM): Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs						
Module 2	Finite Automata	Assignment	Assignment Problems on DFA, NFA's	13 Sessions		
Topics: Basic concepts of Finite automata, DFA- definitions of DFA, Deterministic Accepters Transition Graphs and Languages and DFA's, Regular Languages, NFA- Definition of a Nondeterministic Accepter, Languages and NFA's Why Non- determinism? Equivalence of Deterministic and Nondeterministic Finite Accepters, Reduction of the Number of States in Finite Automata.						
Module 3	Regular Expressions & Context Free Grammar	Assignment	Problems on RE, CFG, PT, PL and Ambiguity	12 Sessions		

Topics: Formal Definition of a Regular Expression, Languages Associated with Regular Expressions, Languages, Regular Languages (RL) and Non-regular Languages: Closure properties of RLs, to show some languages are not RLs, Closure Properties of Regular Context Free Grammars-Examples of Context-Free Languages, Leftmost and Rightmost Derivations, Derivation Trees, Relation Between Sentential Forms and Derivation Trees, Ambiguity in Grammars and Languages: Ambiguous Grammars, Removing Ambiguity, Chomsky Normal Form, Gribiche Normal Form.				
Module 4	Push down Automata	Assignment	Problems on pushdown Automaton	08 Sessions
Topics: Definition of a Pushdown Automaton, Language Accepted by a Pushdown Automaton, Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack Equivalence of PDA's and CFG's: From Grammars to Pushdown Automata.				
Module 5	Turing Machine	Assignment	Problems on Turning Machine	07 Sessions
Topics: Definition of a Turing Machine, Turing Machines as Language Accepters, Example Languages to construct Turing machine, Turing Machines as Transducers, Halting Programming Techniques for Turing Machines				
Targeted Application & Tools that can be used: Targeted Application: 1. Text Processing 2. Compilers 3. Text Editors 4. Robotics Applications 5. Artificial Intelligence Tools: 1. JFLAP (Java Formal Language and Automata Package) Software simulation tool. It's interactive educational software written in Java to experiment topics in automata theory. 2. Turing machine Online simulators.				

Text Book(s):

1. Peter Linz, “An introduction to Formal Languages and Automata”, Jones and Bartlett Publications 6th Ed, 2018.

Reference(s):

1. Aho, Ullman and Hopcroft, “Theory of Computation”, Pearson India 3rd Edition 2008.

2. Michael Sipser, “Theory of Computation”, Cengage India 3rd Ed, 2014.

E-Resources

NPTEL course – https://onlinecourses.nptel.ac.in/noc21_cs83/preview

Course Code: CSE3502	**Course Title:** Cryptography and Network Security **Type of Course:** Theory - PCC	**L- T-P- C**	3	**0**	0	**3**
Version No.	2.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	The Course deals with the principles and practice of cryptography and network security, focusing in particular on the security aspects of the web and Internet					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Cryptography and Network Security above and attain Skill Development through Problem Solving methodologies.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Describe the basic concept of Cryptography 2. Classify different types of Cryptographic Algorithms 3. Solve Mathematical problems required for Cryptography 4. Illustrate Network Security concepts					
Course Content:						

Module 1	Introduction to Cryptography	Assignment	Recognize the techniques	**7 Sessions**
Introduction to Cryptography, Model of Network Security, OSI Security architecture, Security Attacks: active attacks, passive attacks, services: Authentication, Access Control, Data Confidentiality, Data Integrity, Nonrepudiation, Substitution Ciphers : Play-fair and Hill Cipher, Vigenere cipher, Introduction to Block Cipher and Stream Cipher, Feistel Structure, ECB modes of block cipher.				
Module 2	Symmetric Encryption Algorithm	Assignment	Analysis of solutions	9 Sessions
Overview of Public Key Cryptography, RSA, Diffie-Helman Key exchange, Man in the middle attack, Cryptographic Hash functions, Secure Hash Algorithm, Message Authentication Codes – HMAC, Digital Signature, Ei-gamal Encryption, Elliptic curve cryptography overview.				
Module 3	Public Key Cryptography	Assignments	Analysis of solutions	9 Sessions
Overview of Public Key Cryptography, RSA, Diffie-Helman Key exchange, Man in the middle attack, Cryptographic Hash functions, Secure Hash Algorithm, Message Authentication Codes – HMAC, Digital Signature, Ei-gamal Encryption, Elliptic curve cryptography overview				
Module 4	Network Security	Assignment	Analysis of solutions	05 Sessions

Topics:

Network Security fundamentals, Network Security applications: Authentication: Kerberos, PKI, Network Security applications: e-mail security: PGP, MIME, Network Security applications: IP Security: IPSec architecture, Network Security applications: DNS Security.

Targeted Application & Tools that can be used:

Students get the knowledge about cryptography techniques followed, the algorithms used for encryption and decryptions & the techniques for authentication and confidentiality of messages.

Text Book(s):

T1 William Stallings, "Cryptography and Network Security - Principles and Practices", 7th Edition, Pearson publication, ISBN: 978-93-325-8522-5, 2017

Reference(s):

R1 Bruice Schneier, “Applied Cryptography – Protocols, Algorithms and Source code in C”, Second Edition, Wiley

Publication, ISBN: 978-81-265-1368-0, 2017

R2 Cryptography and Network Security, Express Learning, ITL Education Solution Limited.

R3 e-pg pathshala UGC lecture series

Web references:
https://puniversity.informaticsglobal.com:2229/login.aspx?direct=true&db=nlebk&AN=2233842&site =ehost-live

https://nptel.ac.in/courses/106105031.

Topics relevant to “Skill Development”: Topics relevant to “Skill Development”:

1. Play-fair and Hill Cipher

2. Euclidean and Extended Euclidean Algorithm

3. Secure Hash Algorithm

4. Diffie-Helman Key exchange

5. Totient Function.

6. Fermat’s little theorem

urse Code: CSE2267	Course Title: Machine Learning Techniques Type of Course: Program Core -Theory	T-P-C	3			3
rsion No.						
urse Pre-requisites	NIL					
ti-requisites	L					
urse Description	Machine Learning algorithms are the key to developing intelligent systems such as Apple’s Siri, Google’s self-driving cars, and more. This course introduces the core concepts and essential algorithms of various machine learning techniques, including Regression learning, Bayesian learning, Ensemble learning, Perceptron learning (foundations of Neural Networks), Unsupervised learning, Competitive learning, learning from Gaussian Mixture Models, and learning to detect outliers. The course lectures cover both the theoretical foundations and practical algorithms for these learning methods. Lab sessions complement the lectures, enabling students to develop intelligent systems for real-life problems using Python libraries.					
urse Objectives	The primary objective of this course is to enhance the **EMPLOYABILITY** of students by leveraging **EXPERIENTIAL LEARNING** techniques. Students will gain practical skills in applying machine learning to real-world challenges.					
urse Out Comes	On successful completion of this course the students shall be able to: 5. **Apply** advanced supervised machine learning methods for predictive modeling. [APPLY] 6. **Produce** machine learning models with better predictive performance using meta-learning (ensemble) algorithms. [APPLY] 7. **Create** predictive models using Perceptron learning algorithms, understanding their foundational role in neural networks. [APPLY] 8. **Employ** advanced unsupervised learning algorithms for clustering, competitive learning, and outlier detection. [APPLY] 9. **Implement** machine learning-based intelligent models using Python libraries and frameworks. [APPLY]					

Course Content:

dule 1	pervised Learning	signment	dule Tests	No. of Sessions: 12
Overview of Machine Learning (ML): ML workflow, types of ML (supervised, unsupervised, reinforcement), types of features, Feature Engineering (Data Imputation Methods). Regression: Introduction, Simple Linear Regression, Loss Functions (MSE, MAE, RMSE), Polynomial Regression, Logistic Regression, Softmax Regression with Cross-Entropy as cost function. Bayesian Learning: Bayes Theorem, estimating conditional probabilities for categorical and continuous features, Naïve Bayes for supervised learning, Bayesian Belief Networks. Support Vector Machines (SVM): Soft margin and kernel tricks (Polynomial, RBF, Sigmoid). Evaluation Methodologies: Testing Dataset, Train-Validation-Testing split, N-Fold Cross Validation (K-Fold, Stratified K-Fold)..				
dule 2	semble Learning	signment	dule Tests	No. of Sessions: 11
Introduction to Ensemble Learning: Motivation, bias-variance trade-off. Bagging: Using subsets of instances (Bagging, Pasting), using subsets of features (Random Patches, Random Subspaces method). Ensemble Methods: Voting Classifier (Hard and Soft Voting), Random Forest (Algorithm, Feature Importance). Boosting: AdaBoost (Adaptive Boosting), Gradient Boosting Machines (GBM), Extremely Randomized Trees, Stacking (Meta-Learning).				
dule 3	rceptron Learning & Neural Networks Foundation	signment	dule Tests	No. of Sessions: 11

From Biological to Artificial Neurons: Basic structure of a neuron. Perceptrons: Single-layer Perceptrons, Linear Threshold Units, logical computations with Perceptrons (AND, OR, NOT). Activation Functions: Common activation functions (Sigmoid, Tanh, ReLU, Leaky ReLU, ELU, Softmax) and their properties. Loss Functions: Common loss functions for classification (Binary Cross-Entropy, Categorical Cross-Entropy) and regression (MSE, MAE). Multi-layer Perceptrons (MLP): Architecture, advantages over single Perceptrons. Backpropagation Algorithm: Understanding the concept of backpropagation, Gradient Descent and its variants (Stochastic Gradient Descent, Mini-batch Gradient Descent, Adam, RMSprop).				
dule 4	**supervised Learning & Advanced Topics**	**signment**	**dule Tests**	**No. of Sessions: 11**
Clustering: Simple K-Means Clustering (Algorithm, Simple and Mini-Batch), updating centroids incrementally. K-Means Enhancements: Finding the optimal number of clusters (Elbow Method, Silhouette Coefficient), drawbacks of K-Means, K-Means++. Hierarchical Clustering: Divisive hierarchical clustering (Bisecting K-Means), clustering using Minimum Spanning Tree (MST). Competitive Learning: Clustering using Kohonen's Self-Organizing Maps (SOM). Density-Based Clustering: DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Probabilistic Clustering: Clustering using Gaussian Mixture Models (GMM) with Expectation-Maximization (EM) algorithm. Outlier Detection Methods: Isolation Forest, Local Outlier Factor (LOF). Association Rule Mining: Introduction, Apriori Algorithm (concepts only). Collaborative Filtering: Introduction (User-based, Item-based).				

Targeted Application & Tools that can be used:

5. Google Colab
6. Python IDEs like PyCharm

aching Methodology

1. **Lectures:** Interactive lectures covering theoretical foundations and algorithmic details.
2. **Lab Sessions:** Hands-on sessions focusing on implementing algorithms using Python libraries (e.g., scikit-learn, NumPy, Pandas, Matplotlib, Seaborn, Keras/TensorFlow for Perceptron).
3. **Problem-Based Learning:** Real-world case studies and problems will be discussed and solved in labs.
4. **Experiential Learning:** Emphasis on practical application through assignments, mini-projects, and a final course project.
5. **Discussions:** Encouraging critical thinking and peer learning through in-class discussions.

oject work/Assignment:

2. Students can be assigned a mini project to develop a machine learning application for real-life problems in various domains such as health care, business intelligence, environmental modeling, etc.

xtbook(s):

T1. Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, Oreilly, 3rd Edition, 2022.

ferences:

R1. Raschka, S., Liu, Y., & Mirjalili, V. (2022). Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python. Packt Publishing Ltd.
R2. Alpaydin, E. (2021). Machine Learning, revised and updated edition. MIT Press.
R3. Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). Mathematics for machine learning. Cambridge University Press.
R4. Andreas C Muller, and Sarah Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly, 1st Edition, 2016.

eblinks

W1. W1. NPTEL Courses: https://nptel.ac.in/courses/106106139 (IIT M), https://nptel.ac.in/courses/106105152 (IIT Kgp)
W2. Scikit-learn documentation: https://scikit-learn.org/stable/
W3. TensorFlow documentation: https://www.tensorflow.org/
W4. Keras documentation: https://keras.io/
W5. Pandas documentation: https://pandas.pydata.org/

W6. NumPy documentation: https://numpy.org/ W7. Kaggle: For datasets and competitions.

urse Code: CSE2268	**Course Title: Machine Learning Techniques Lab** **ype of Course: Program Core -Laboratory**	**T-P-C**	0			1
rsion No.						
urse Pre-requisites						
ti-requisites						
urse Description	Machine Learning algorithms are the key to develop intelligent systems such as Apple's Siri, Google's self-driving cars etc. This course introduces the concepts of the core machine learning techniques such as Regression learning, Bayesian learning, Ensemble learning, Perceptron learning, Unsupervised learning, Competitive learning, learning from Gaussian mixture models and learning to detect outliers. Course lectures covers both the theoretical foundations as well as the essential algorithms for the various learning methods. Lab sessions complement the lectures and enable the students in developing intelligent systems for real life problems.					
urse Objectives	The objective of the course is EMPLOYBILITY of student by using EXPERIENTIAL LEARNING techniques.					
urse Out Comes	On successful completion of this course the students shall be able to: 1. **Apply** advanced supervised machine learning methods for predictive modeling. [Apply] 2. **Produce** machine learning models with better predictive performance using meta learning algorithms [Apply] 3. **Create** predictive models using Perceptron learning algorithms [Apply] 4. **Employ** advanced unsupervised learning algorithms for clustering, competitive learning and outlier detection [Apply] 5. **Implement** machine learning based intelligent models using Python libraries. [Apply]					

Course Content:

Experiment No. 1: File Handling Using Python
Level 1: Read a CSV file using Python
Level 2: Read a text file using Python

Experiment No. 2: Methods for handling missing values
Level 1: Given a data set from UCI repository, implement the different ways of handling missing values in it using Scikit-learn library of Python
Level 2: Implement one of these methods using a custom defined function in Python.

Experiment No. 3: Data Visualization
Level 1: Perform Exploratory Data Analysis for a given data set by creating Scatter Plot, Pair Plot, Count Plot using Matplotlib and Seaborn
Level 2: Create Heat Maps, WordCloud

Experiment No. 4: Regression learning
Level 1: Given a data set from UCI repository, implement the simple linear regression algorithm and estimate the models parameters and the performance metrics. Plot the learning curves.
Level 2: Implement the polynomial regression algorithm. Compare the learning curves of Polynomial and Linear Regression.

Experiment No. 5: Logistic Regression
Level 1: Write custom code for generating the logistic/sigmoid plot for a given input
Level 2: Given a data set from UCI repository, implement the Logistic regression algorithm. Estimate the class probabilities for a given test data set. Plot and analyze the decision boundaries.

Experiment No. 6: Bayesian Learning
Level 1: Given a data set from UCI repository, implement a classification model using the Bayesian algorithm.
Level 2: Implement a Naïve Bayes classifier using 5-fold cross-validation

Experiment No. 7: Support Vector Machine (SVM)
Level 1: Given data sets from UCI repository, implement a linear SVM and a non-linear SVM based classification model.
Level 2: Construct kernels with 5-fold cross-validation for SVM.

Experiment No. 8 & 9: Ensemble Learning
Level 1: Implement Ensemble Learning algorithms such as Bagging, Pasting and Out-of Bag Evaluation
Level 2: Random Patches and Random Subspace Method, Adaboost and Gradient Boosting, Stacking.

Experiment No. 10: Perceptron Learning
Level 1: Implement the Perceptron Classifier
Level 2: An Image Classifier Using the Sequential API of Keras

Experiment No. 11 & 12: Unsupervised Learning
Level 1: K-means – simple and mini-batch. Finding the optimal number of clusters using Elbow method and Silhoutte Coefficient . Compare the inertia of both as k increases. Tuning the hyperparameter 'k' using GridSearchCV.
Level 2: Using clustering for Image segmentation and Preprocessing. Kmeans++

Experiment No. 13: Density Based Clustering
Level 1: Implement DBSCAN – clustering using the local density estimation. Perform hard and soft clustering for new instances.
Level 2: Outlier Detection using Isolation Forest and Local Outlier Factor

Experiment No. 14: Association Rule Mining
Level 1: Implement the Apriori Algorithm for Association Rule Mining
Level 2: Implement the Dynamic Itemset Counting Algorithm for Association Rule Mining.

Experiment No. 15: Collaborative Filtering
Level 1: Implement Collaborative Filtering using Item-Based Filtering
Level 2: Implement Collaborative Filtering using User-Based Filtering

Targeted Application & Tools that can be used:

7. Google Colab
8. Python IDEs like PyCharm

oject work/Assignment: Mention the Type of Project /Assignment proposed for this course

Students can be assigned a mini project to develop a machine learning application for real-life problems in various domains such as health care, business intelligence, environmental modeling, etc.

xtbook(s):

3. Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, Oreilly, 3rd Edition, 2022.

ferences:

R1. Andreas C Muller, and Sarah Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly, 1st Edition, 2016.

eblinks

W1. NPTEL Courses: https://nptel.ac.in/courses/106106139 (IIT M), https://nptel.ac.in/courses/106105152 (IIT Kgp)

Course Code: CSE2269	Course Title: **Operating Systems** Type of Course: **Program Core and Theory Only**	L-T- P- C	3	0	0	3

Version No.	1.0			
Course Pre-requisites	Nil			
Anti-requisites	**NIL**			
Course Description	This course introduces the concepts of operating system operations, operating system structure and its design and implementation. It covers the classical operating systems internal algorithms such as process scheduling, synchronization, deadlocks detection and recovery and memory management. The course also enhances the problem solving, systems programming ability and case studies.			
Course Object	The objective of the course is to familiarize the learners with the concepts of Operating Systems and attain **Employability** through **Problem Solving** Methodologies.			
Course Out Comes	On successful completion of the course the students shall be able to: 1] Describe the fundamental concepts of operating Systems and case studies. [**Knowledge**] 2] Demonstrate various CPU scheduling algorithms. .[**Application**] 3] Apply various tools to handle synchronization problems.[**Application**] 4] Demonstrate deadlock detection and recovery methods **[Application]** 5] Illustrate various memory management techniques.[**Application**]			
Course Content:				
Module 1	Introduction to Operating System	Assignment	Programming	**9 Hours**
Topics: Introduction to OS , Operating-System Operations, Operating System Services, , System Calls and its types, Operating System Structure, System Program and its types, Linkers and Loaders, Overview of OS design and implementation, Open-source operating system				
Module 2	Process Management	Assignment/Case Study	Programming/Simulation	**11 Hours**
Topics: Process Concept, Operations on Processes, Inter Process Communication, Communication in client-server systems (sockets, RPC, Pipes), Introduction to threads - Multithreading Models, Thread Libraries, Threading Issues, Process Scheduling– Basic concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF, RR and Priority.				
Module 3	Process Synchronization and Deadlocks	Assignment	Programming	**11 Hours**
Topics: The Critical-Section Problem- Peterson's Solution, Synchronization hardware, Semaphores, Classic Problems of Synchronization with Semaphore Solution- Producer-Consumer Problem, Reader-Writer problems, Dining Philosopher's Problem, . Introduction to Deadlocks, Necessary conditions for deadlock, Resource allocation Graph, Methods for handling deadlock: Deadlock Prevention and Implementation, Deadlock Avoidance and Implementation, Deadlock detection & Recovery from Deadlock.				
Module 4	Memory Management	Assignment	Programming/Simulation	**10 Hours**
Topics: Introduction to Memory Management, Basic hardware-Base and Limit Registers, Memory Management Unit(MMU), Dynamic loading and linking, Swapping, Contiguous and Non-Contiguous Memory Allocation, Segmentation, Paging - Structure of the Page Table – Virtual Memory and Demand Paging – Page Faults and Page Replacement Algorithms, Copy-on-write, Allocation of Frames, Thrashing Introduction to File system management: File System Interface (access methods, directory structures), File system implementation.				
Targeted Application: **Application area is traffic management system, banking system, health care and many more systems**				

where in there are resources and entities that use and manage the resources.

Software Tools:

Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]. Used to install and work on multiple guest Operating Systems on top of a host OS.

Intel Processor identification utility: This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.

Project work/Assignment

1. **Demonstrate process concepts in LINUX OS.**
2. **Simulation of CPU scheduling algorithms.**
3. **Develop program to demonstrate use of Semaphores in threads.**
4. **Develop program to demonstrate use of deadlock avoidance algorithms.**
5. **Develop program to demonstrate use of page replacement algorithms.**
6. **Simulation of memory allocation strategies [first fit, best fit and worst fit].**

Text Book

1. Silberschatz A, Galvin P B and Gagne G , “Silberschatz's Operating System Concepts”, Paperback, Global Edition Wiley, 2019

References

1. Silberschatz A, Galvin P B and Gagne G, “Operating System Concepts”, 10th edition Wiley, 2018.
2. William Stallings, “Operating Systems”,Ninth Edition, By Pearson Paperback ,1 March 2018.
3. Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, “ Cracking the Operating System skills”, Dreamtech, paperback, 2020
4. Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , “Operating Systems: Three Easy Pieces, Amazon digital Services”, September 2018.

E-resources/Weblinks

https://www.os-book.com/OS9/
https://pages.cs.wisc.edu/~remzi/OSTEP/
https://codex.cs.yale.edu/avi/os-book/OS10/index.html

Course Code: CSE2270	**Course Title: Operating Systems Lab** **Type of Course: Lab Only**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Nil					
Anti-requisites	**NIL**					
Course Description	This laboratory course provides hands-on experience with the core concepts of operating systems through practical assignments, simulations, and case studies. It covers foundational aspects such as system calls, process and thread management, inter-process communication, synchronization, deadlocks, memory management, and file systems. Students will implement and simulate real-time OS components and scheduling algorithms, fostering deeper understanding of OS architecture and design. The lab also introduces modern OS tools, programming interfaces, and the basics of open-source OS environments.					
Course Object	The objective of the course is to familiarize the learners with the concepts of Operating Systems and attain **Employability** through **Problem Solving** Methodologies.					
Course Out Comes	On successful completion of the course the students shall be able to: 1] Demonstrate system-level programming using system calls and OS structures. [**Apply**] 2] Simulate process scheduling and multithreading techniques. [**Apply**] 3] Apply various tools to handle synchronization problems using semaphores and shared memory. [**Apply**] 4] Demonstrate memory management and file system concepts using simulation or scripting. **[Apply]**					
Course Content:						

Targeted Application:
Application area is traffic management system, banking system, health care and many more systems where in there are resources and entities that use and manage the resources.

Software Tools:
Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]. Used to install and work on multiple guest Operating Systems on top of a host OS.
Intel Processor identification utility: This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.

of Laboratory Tasks**:**
sheet -1

Write a program to demonstrate the use of fork() and exec() system calls in process creation.
A system has limited memory and high-priority real-time processes. Design a scheduling algorithm that ensures responsiveness while preventing starvation.
sheet -2

Implement First-Come-First-Serve (FCFS) process scheduling using C or Python.
You are designing a server that handles thousands of client connections. Compare multithreading and multiprocessing for this task and implement a basic server model.
sheet -3

Implement Round Robin Scheduling with a fixed time quantum.
In a banking system, concurrent access to accounts leads to data corruption. Design a synchronization solution to avoid race conditions.
sheet -4

Write a program to create threads using Pthreads or Python's threading module.

L2: You're tasked with building a file access tracker in an OS. Implement a system to log file access patterns and identify frequent accesses.

sheet -5

Demonstrate inter-process communication (IPC) using pipes.

A simulation tool needs to emulate process suspension and resumption. Design and implement such a mechanism using signals or condition variables.

sheet -6

Simulate the Producer-Consumer problem using semaphores.

You're developing a system where sensor devices (producers) generate temperature readings, and data processors (consumers) store and process these readings. To prevent race conditions and ensure buffer safety, implement a synchronization mechanism using semaphores.

sheet -7

Implement Dining Philosophers Problem using threads and synchronization.

In a multi-threaded cafeteria simulation, five philosophers sit around a circular table, each alternating between thinking and eating. To eat, a philosopher must hold two forks (represented by shared resources). Your task is to avoid deadlock and ensure no philosopher starves using thread synchronization techniques.

sheet -8

Write a program to simulate First Fit, Best Fit, and Worst Fit memory allocation strategies.

A system with limited memory blocks needs to allocate memory to processes arriving with various size requests. Your task is to implement three classic memory allocation strategies—First Fit, Best Fit, and Worst Fit—to allocate memory to each process efficiently. Simulate and compare how memory gets allocated in each strateg

sheet -9

Demonstrate paging using a simple page table simulation.

A program has a logical address space divided into pages. The system's memory is divided into equal-sized frames. When a program executes, its pages are loaded into available frames in main memory. Simulate the address translation process using a page table and demonstrate how a logical address is converted to a physical address.

sheet -10

Write a program to simulate page replacement algorithms like FIFO and LRU.

In a virtual memory system, a process accesses pages in a specific order. The memory can only hold a limited number of pages (frames). When a page is needed and the memory is full, a page replacement algorithm is used to decide which page to evict. Simulate and compare FIFO and LRU algorithms for a given page reference string.

sheet -11

Simulate file directory structure (single level/two level).

A university campus computer lab has limited memory space available for each student login session. When students open files or run programs, memory pages are loaded into available memory frames. Due to the limited number of frames, some pages must be replaced when new ones are needed. The lab system uses page replacement algorithms to decide which pages to evict when memory is full..

sheet -12

Write a shell script to demonstrate file handling commands in Linux.

Design a command-line mini shell that can run background and foreground processes and handle basic built-in commands like cd, pwd, exit.

Project work/Assignment

1. **Demonstrate process concepts in LINUX OS.**
2. **Simulation of CPU scheduling algorithms.**
3. **Develop program to demonstrate use of Semaphores in threads.**
4. **Develop program to demonstrate use of deadlock avoidance algorithms.**
5. **Develop program to demonstrate use of page replacement algorithms.**
6. **Simulation of memory allocation strategies [first fit, best fit and worst fit].**

Text Book

1. Silberschatz A, Galvin P B and Gagne G , “Silberschatz's Operating System Concepts”, Paperback, Global Edition Wiley, 2019

References

1. Silberschatz A, Galvin P B and Gagne G, “Operating System Concepts”, 10th edition Wiley, 2018.
2. William Stallings, “Operating Systems”,Ninth Edition, By Pearson Paperback ,1 March 2018.
3. Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, “ Cracking the Operating System skills”, Dreamtech, paperback, 2020
4. Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , “Operating Systems: Three Easy Pieces, Amazon digital Services”, September 2018.

E-resources/Weblinks

1. https://www.os-book.com/OS9/
2. https://pages.cs.wisc.edu/~remzi/OSTEP/
3. https://codex.cs.yale.edu/avi/os-book/OS10/index.html

<table>
<tr><td>Course Code:

CSE2503</td><td colspan="2">Course Title: Scalable Application Development using Java

Type of Course: Theory - PCC</td><td>L- T-P- C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="7">2.0</td></tr>
<tr><td>Course Prerequisites</td><td colspan="7"></td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">This course provides a comprehensive guide to designing, developing, and deploying scalable Java applications, covering high-performance architectures, distributed systems, microservices, cloud deployment, and DevOps integration.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is to Understand Scalability Principles, developing high performance Java applications, design and implement scalable architecture, deploy and manage scalable data.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course the students shall be able to:

CO 1: Design and Develop Scalable Java Applications

CO 2: Design and Implement Scalable Architectures and Micro services

CO 3: Implement Scalable Data Management Techniques

CO 4: Deploy and Monitor Applications in Cloud Environments</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Foundations of Scalable Java Applications</td><td>Assignment</td><td colspan="3">Implementation</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="8">Topics:

Introduction to Scalability - Defining scalability: vertical vs. horizontal scaling, Monolithic vs. Microservices Architecture, Performance bottlenecks and solutions.

Java Performance Optimization - JVM internals and tuning, Garbage Collection (GC) strategies, Profiling and monitoring Java applications

Concurrency and Multithreading - Java concurrency model and thread management, Executor framework, ForkJoinPool, Thread safety, locks, and synchronization.</td></tr>
<tr><td>Module 2</td><td>Scalable Architectures and Microservices</td><td>Assignment</td><td colspan="3">Implementation</td><td colspan="2">10 Sessions</td></tr>
</table>

Topics: **Microservices and Distributed Systems -** Principles of microservices, Service-to-service, communication (REST, gRPC, Kafka), API Gateway, Service Discovery, and Load Balancing **Design Patterns for Scalability -** Singleton, Factory, and Builder patterns, CQRS (Command Query Responsibility Segregation), Circuit Breaker and Retry patterns (Resilience4j) **Event-Driven Architecture & Asynchronous Processing -** Event Sourcing with Kafka, WebSockets and Reactive Programming, Handling failures in distributed systems.

Module 3	Scalable Data Management and Caching	Mini - Project	Implementation	10 Sessions

Topics: **Database Scalability -** SQL vs. NoSQL databases (MySQL, PostgreSQL, MongoDB, Cassandra), Sharding, Replication, and Partitioning strategies, Optimizing queries (Indexing, Query Caching) **Caching Strategies -** In-memory caching with **Redis, Memcached,** Distributed caching techniques, Cache Invalidation and Eviction Policies **Data Streaming & Batch Processing -** Apache Kafka for event-driven data pipelines, Apache Spark for large-scale data processing.

Module 4	Cloud Deployment and DevOps for Scalability	Quiz	Implementation	10 Sessions

Topics: **Cloud-Native Development -** Containerization with **Docker,** Kubernetes for scaling and orchestration, Cloud deployment strategies (AWS, GCP, Azure), **CI/CD & DevOps for Scalable Applications -** Building CI/CD pipelines (Jenkins, GitHub Actions), Canary & Blue-Green Deployments, Infrastructure as Code (Terraform, Kubernetes Helm) **Security & Reliability -** Securing APIs with **OAuth2, JWT, API Gateway,** Load Balancing & Rate Limiting, Monitoring with **Prometheus, ELK Stack**
Targeted Application & Tools that can be used: **Applications**: The course will focus on building and scaling the following types of applications: ● **E-Commerce Platforms** – Handling high traffic, product catalogs, and real-time inventory. ● **Banking & FinTech Applications** – Secure and high-availability transactions. ● **Streaming & Event-Driven Applications** – Using Kafka for real-time data processing.

- **Social Media & Messaging Platforms** – Scalable messaging and real-time updates.
- **SaaS & Cloud-Native Applications** – Multi-tenant, API-based scalable solutions.

Tools:

Programming & Frameworks

- **Java 17+** – Core programming language for scalable applications.
- **Spring Boot** – Microservices development, REST APIs, and dependency injection.
- **Spring Cloud** – Service discovery, load balancing, and resilience patterns.
- **Quarkus** – Lightweight, high-performance microservices alternative.

Database & Caching

- **SQL Databases:** MySQL, PostgreSQL – Scalable relational data management.
- **NoSQL Databases:** MongoDB, Cassandra – High-availability distributed databases.
- **Redis / Memcached** – In-memory caching for faster response times.

Messaging & Event-Driven Architecture

- **Apache Kafka** – Real-time event streaming and asynchronous communication.
- **RabbitMQ** – Message brokering for decoupling services.
- **gRPC** – High-performance inter-service communication.

Cloud & Deployment

1. **Docker** – Containerization for application portability.
2. **Kubernetes** – Scaling, orchestration, and auto-recovery.
3. **AWS / GCP / Azure** – Cloud deployment and auto-scaling.

CI/CD & DevOps

- **GitHub Actions / Jenkins** – Automated build and deployment pipelines.
- **Terraform** – Infrastructure as Code for cloud provisioning.
- **Helm** – Kubernetes package management for scalable applications.

Text Book(s):

T1: "Designing Data-Intensive Applications" – *By Martin Kleppmann*

T2: "Java Concurrency in Practice" – *By Brian Goetz*

T3: "Spring Microservices in Action" – *By John Carnell*

T4. "Cloud Native Java" – *By Josh Long & Kenny Bastani*

Reference(s):

- **"Designing Data-Intensive Applications"** – *Martin Kleppmann*
- **"Java Performance: The Definitive Guide"** – *Scott Oaks*
- **"Spring Microservices in Action"** – *John Carnell*
- **"Kubernetes Up & Running"** – *Kelsey Hightower, Brendan Burns, Joe Beda*

<table>
<tr><td>Course Code:

CSE2504</td><td>Course Title: Scalable Application Development using Java Lab

Type of Course: Lab - PCC</td><td>L- T-P- C</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td>Version No.</td><td colspan="6">2.0</td></tr>
<tr><td>Course Prerequisites</td><td colspan="6"></td></tr>
<tr><td>Anti-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td colspan="6">This course provides a hands-on, practical approach to building scalable, high-performance applications using Java and related technologies. This course is designed to complement theoretical concepts by offering real-world lab exercises focused on the development of microservices architectures, cloud-native applications, and distributed systems.

In this lab-intensive course, students will work on building and deploying scalable applications using Spring Boot, Spring Cloud, Docker, Kubernetes, and Apache Kafka. Students will gain experience in implementing RESTful APIs, asynchronous messaging, data caching, and load balancing to ensure that applications can handle increased traffic and scale efficiently. The course will also cover essential techniques for optimizing performance, including JVM tuning, database optimization, and memory management.</td></tr>
<tr><td>Course Objective</td><td colspan="6">The primary objectives of the course are to Develop hands-on expertise in building scalable applications using Java and modern frameworks like Spring Boot, Spring Cloud, and Apache Kafka, Implement microservices architectures that enable applications to handle increasing loads efficiently through distributed systems and cloud-native practices, Gain practical experience in optimizing performance by leveraging tools for JVM tuning, database optimization, and memory management to improve application responsiveness and scalability., Work with containerization technologies such as Docker and Kubernetes to deploy Java applications in cloud environments with automated continuous integration/continuous deployment (CI/CD) pipelines, Master service discovery, load balancing, and caching mechanisms to ensure high availability, fault tolerance, and low-latency operations in production-grade applications and Apply event-driven architectures to build scalable and resilient systems using tools like Apache Kafka for real-time data processing and messaging.</td></tr>
<tr><td>Course Outcomes</td><td colspan="6">On successful completion of the course the students shall be able to:

1. Implement Performance Optimization Techniques
2. Design and Build Scalable Microservices
3. Integrate Event-Driven Architectures and Caching.
4. Deploy and Scale Applications in Cloud Environments.</td></tr>
<tr><td>Course Content:</td><td colspan="6"></td></tr>
</table>

Module 1	Foundations of Scalable Java Applications	Assignment	Implementation	**10 Sessions**

Lab Assignment 1: Setting Up Development Environment

Objective: Set up the Java development environment and configure a Spring Boot project for scalability testing.
Tasks:

- Install **Java 17**, **Maven**, and **IDE** (IntelliJ or Eclipse).
- Set up a basic **Spring Boot project** using **Spring Initializr** with **RESTful endpoints**.
- Verify application functionality by running a **local server** and testing API responses via **Postman** or **curl**.
- Add a basic **Spring Boot Actuator** to monitor application health and performance.
Deliverables:
- Working **Spring Boot application** with basic endpoints.
- Screenshots of successful tests (Postman or curl).

Lab Assignment 2: Performance Optimization with JVM

Objective: Profile and optimize a Java application for better performance.
Tasks:

- Implement a **simple Java application** that performs a memory-intensive task (e.g., sorting a large dataset).
- Use **VisualVM** to monitor **JVM memory usage, CPU usage, and garbage collection**.
- Optimize the application by adjusting **JVM flags** (e.g., **heap size**, **garbage collection strategy**).
- Measure the impact of optimizations on **execution time and memory usage**.
Deliverables:
- Profiled and optimized **Java application** with performance comparison charts.
- Detailed report on **JVM tuning** and optimization strategies.

Lab Assignment 3: Implementing Multi-threading

Objective: Understand Java's concurrency model and implement multi-threading for parallel tasks.
Tasks:

- Implement a **multi-threaded Java application** that simulates multiple tasks (e.g., processing large files, image processing).
- Use the **Executor framework** to manage thread pools.
- Measure the **execution time** and compare the performance of **single-threaded** vs **multi-threaded** approaches.
Deliverables:

- Source code for **multi-threaded application** with explanations of thread management.
- Execution time comparison chart.

Module 2	Scalable Architectures and Microservices	Assignment	Implementation	10 Sessions

Lab Assignment 4: Building a Simple Microservice with Spring Boot

Objective: Develop a **basic microservice** with Spring Boot.
Tasks:

- Create a **Spring Boot microservice** that provides a RESTful API (e.g., a book or customer service).
- Implement **basic CRUD operations** (Create, Read, Update, Delete).
- Use **Spring Data JPA** to interact with an **SQL database** (e.g., MySQL).
- Write unit tests using **JUnit** for API endpoints.
Deliverables:
- Working **Spring Boot microservice** with API documentation.
- Source code with unit tests.

Lab Assignment 5: Implementing Service Discovery & Load Balancing

Objective: Enable **service discovery** and **load balancing** using **Spring Cloud Eureka**.
Tasks:

- Set up a **Spring Cloud Eureka server** for service discovery.
- Create two **Spring Boot microservices** that register with the Eureka server.
- Implement **load balancing** with **Spring Cloud Ribbon** by making API calls to different instances of the microservices.
- Test and verify load balancing behavior using **Postman** or **curl**.
Deliverables:
- Spring Cloud **Eureka server** and two **microservices**.
- Load balancing validation and test results.

Lab Assignment 6: Building an Event-Driven System with Kafka

Objective: Implement an **event-driven architecture** using **Apache Kafka** for inter-service communication.
Tasks:

- Set up **Apache Kafka** locally or in Docker.
- Create two Spring Boot applications: one as a **Kafka producer** and the other as a **consumer**.
- Implement asynchronous message communication where the producer sends messages (e.g., order

events) and the consumer processes them.

- Add error handling and retry logic using **Spring Kafka**.
  Deliverables:
- Kafka producer and consumer applications with **message processing logic**.
- Screenshots or logs showing messages being passed from producer to consumer.

Module 3	Scalable Data Management and Caching	Mini - Project	Implementation	10 Sessions

Lab Assignment 7: Integrating SQL and NoSQL Databases

Objective: Learn to integrate **relational (SQL)** and **non-relational (NoSQL)** databases with Spring Boot applications.
Tasks:

- Integrate a **MySQL** database into a **Spring Boot microservice** and implement CRUD operations.
- Set up a **MongoDB** instance (locally or via Docker) and create a second microservice using **Spring Data MongoDB**.
- Compare the performance and scalability aspects of both databases.
  Deliverables:
- Source code for Spring Boot microservices using **MySQL** and **MongoDB**.
- Database performance comparison with benchmarks.

Lab Assignment 8: Implementing Redis Caching

Objective: Improve application performance using **Redis** as an in-memory cache.
Tasks:

- Integrate **Redis** with your Spring Boot application.
- Cache frequently accessed data (e.g., product information, user profiles) in Redis.
- Implement **cache expiration** and **cache invalidation** strategies.
- Measure the performance improvement by comparing **cache hits vs. misses**.
  Deliverables:
- Redis-integrated Spring Boot application with caching logic.
- Performance comparison between cached and non-cached operations.

Lab Assignment 9: Data Streaming with Kafka

Objective: Implement a **data streaming pipeline** with Kafka for real-time data processing.
Tasks:

- Build a **data pipeline** where **Kafka producers** stream events (e.g., logs, transactions) to **Kafka brokers**.
- Use **Kafka consumers** to process these events in real-time (e.g., updating a database or triggering a

workflow).

- Implement **Kafka Streams** to process data within the Kafka ecosystem.
  Deliverables:
- Kafka producer-consumer setup with real-time data flow.

Module 4	Cloud Deployment and DevOps for Scalability	Quiz	Implementation	10 Sessions

Lab Assignment 10: Dockerizing a Spring Boot Application

Objective: Containerize a Spring Boot application using **Docker** for scalability.
Tasks:

- Write a **Dockerfile** for your Spring Boot application.
- Build a **Docker image** and run the application in a Docker container.
- Test the application running in the container and compare performance to local deployment.
  Deliverables:
- Dockerized **Spring Boot application** with a functional test.
- Docker image and run command documentation.

Lab Assignment 11: Deploying with Kubernetes

Objective: Deploy a Spring Boot microservice to a **Kubernetes cluster** for horizontal scaling.
Tasks:

- Deploy the **Dockerized Spring Boot application** to a **Kubernetes cluster** (local Minikube or cloud-based).
- Set up **Kubernetes Pods, Services, and Deployments** to scale the microservice.
- Implement **auto-scaling** based on CPU or memory usage.
  Deliverables:
- Kubernetes **deployment YAML files** for Spring Boot application.
- Running Kubernetes cluster with auto-scaling behavior.

Lab Assignment 12: Setting Up CI/CD for Scalable Applications

Objective: Automate the deployment process using **CI/CD pipelines**.
Tasks:

- Set up a **Jenkins** or **GitHub Actions** CI/CD pipeline to automate the build, test, and deployment process.
- Implement **continuous deployment** (CD) for deploying a **Dockerized microservice** to a Kubernetes cluster.
- Test the pipeline by pushing changes to a GitHub repository and triggering the deployment process automatically.

Deliverables:

- **CI/CD pipeline configuration** (Jenkins or GitHub Actions).
- Deployment automation logs and screenshots.

Targeted Application & Tools that can be used:

Applications:

The course will focus on building and scaling the following types of applications:

1. **E-Commerce Platforms** – Handling high traffic, product catalogs, and real-time inventory.
2. **Banking & FinTech Applications** – Secure and high-availability transactions.
3. **Streaming & Event-Driven Applications** – Using Kafka for real-time data processing.
4. **Social Media & Messaging Platforms** – Scalable messaging and real-time updates.
5. **SaaS & Cloud-Native Applications** – Multi-tenant, API-based scalable solutions.

Tools:

Programming & Frameworks

- **Java 17+** – Core programming language for scalable applications.
- **Spring Boot** – Microservices development, REST APIs, and dependency injection.
- **Spring Cloud** – Service discovery, load balancing, and resilience patterns.
- **Quarkus** – Lightweight, high-performance microservices alternative.

Database & Caching

- **SQL Databases:** MySQL, PostgreSQL – Scalable relational data management.
- **NoSQL Databases:** MongoDB, Cassandra – High-availability distributed databases.
- **Redis / Memcached** – In-memory caching for faster response times.

Messaging & Event-Driven Architecture

- **Apache Kafka** – Real-time event streaming and asynchronous communication.
- **RabbitMQ** – Message brokering for decoupling services.
- **gRPC** – High-performance inter-service communication.

Cloud & Deployment

- **Docker** – Containerization for application portability.

- **Kubernetes** – Scaling, orchestration, and auto-recovery.
- **AWS / GCP / Azure** – Cloud deployment and auto-scaling.

CI/CD & DevOps

- **GitHub Actions / Jenkins** – Automated build and deployment pipelines.
- **Terraform** – Infrastructure as Code for cloud provisioning.
- **Helm** – Kubernetes package management for scalable applications.

Text Book(s):

T1: "Spring in Action" by Craig Walls

T2: "Java Performance: The Definitive Guide" by Scott Oaks

T3: "Designing Data-Intensive Applications" by Martin Kleppmann

T4. "Spring Microservices in Action" by John Carnell

Reference(s):

4. **"Designing Data-Intensive Applications"** – *Martin Kleppmann*
5. **"Java Performance: The Definitive Guide"** – *Scott Oaks*
6. **"Spring Microservices in Action"** – *John Carnell*
7. **"Kubernetes Up & Running"** – *Kelsey Hightower, Brendan Burns, Joe Beda*

<table>
<tr><td>Course Code:
CSE7000</td><td>Course Title: Internship
Type of Course:</td><td>L- T-P- C</td><td>-</td><td>-</td><td>-</td><td>2</td></tr>
<tr><td>Version No.</td><td colspan="6">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="6">Knowledge and Skills related to all the courses studied in previous semesters.</td></tr>
<tr><td>Anti-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td colspan="6">Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems.</td></tr>
<tr><td>Course Objectives</td><td colspan="6">The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="6">On successful completion of this course the students shall be able to:
1. Identify the engineering problems related to local, regional, national or global needs. (Understand)
2. Apply appropriate techniques or modern tools for solving the intended problem. (Apply)
3. Design the experiments as per the standards and specifications. (Analyze)
4. Interpret the events and results for meaningful conclusions. (Evaluate)</td></tr>
</table>

<table>
<tr><td>Course Code:
APT4006</td><td colspan="2">Course Title: Logical and Critical Thinking
Type of Course: Audited</td><td>L- T-P- C</td><td>0</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>Version No.</td><td colspan="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">Students should have the basic concepts of Logical reasoning and Critical thinking, along with its applications in real life problems.</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">Nil</td></tr>
<tr><td>Course Description</td><td colspan="7">This is a skill-based training program for the engineering students (Undergraduate). This course is designed to enable the students to enhance their skills in Logical reasoning and Critical thinking.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is to familiarize the learners with concepts in Logical reasoning and Critical thinking through problem solving techniques suitable for their career development.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course the students shall be able to:
CO1] Understand all the concepts.
CO2] Apply the concepts in problem solving.
CO3] Analyze and structure the reasoning techniques and spatial visualization skills</td></tr>
<tr><td colspan="8">Course Content:</td></tr>
<tr><td>Module 1</td><td>Logical Thinking</td><td>Assignment</td><td colspan="4"></td><td>16 Hours</td></tr>
<tr><td colspan="8">Topics:
Syllogisms, Cubes and Dices, Mirror and Water images, Paper cutting and Folding, Embedded figures & Completion of figures, Data Interpretation, Data sufficiency</td></tr>
<tr><td>Module 2</td><td>Critical Thinking</td><td>Assignment</td><td colspan="4"></td><td>14 Hours</td></tr>
<tr><td colspan="8">Topics:
Analogy, Symbol and Notations, Statement and assumption, Cause of action, Statement and conclusion, Puzzles</td></tr>
<tr><td colspan="8">Targeted Application & Tools that can be used:
Application area: Placement activities and Competitive examinations. Tools: LMS</td></tr>
<tr><td rowspan="2">Evaluation</td><td colspan="7">Continuous Evaluation</td></tr>
<tr><td colspan="7">Topic wise evaluation
Internal Assessments</td></tr>
<tr><td colspan="8">Text Book
1. A new approach to reasoning verbal, non-verbal & analytical by BS Sijwali
2. R S Aggarwal
3. Kiran publications</td></tr>
<tr><td colspan="8">References
1. www.indiabix.com
2. www.testbook.com
3. www.youtube.com/c/TheAptitudeGuy/videos</td></tr>
<tr><td colspan="8">Topics relevant to Skill Development Logical reasoning and Critical thinking for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.</td></tr>
</table>

<table>
<tr><td>Course Code:
CSE2271</td><td colspan="3">Course Title: Software Design and Development
Type of Course: School Core [Theory Only]</td><td>L-T- P- C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">The objective of this course is to provide the fundamentals concepts of Software Engineering process and principles.
The course covers software requirement engineering processes, system analysis, design, implementation and testing aspects of software system development.
The course covers software quality, configuration management and maintenance.</td></tr>
<tr><td>Course Objectives</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Software Engineering and attain Skill Development through Participative Learning techniques.</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of this course the students shall be able to:
1] Describe the Software Engineering principles, ethics and process models(Knowledge)
2] Identify the requirements, analysis and appropriate design models for a given application(Comprehension)
3] Understand the Agile Principles(Knowledge)
4] Apply an appropriate planning, scheduling, evaluation and maintenance principles involved in software(Application)</td></tr>
<tr><td></td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction to Software Engineering and Process Models
(Knowledge level)</td><td>Quiz</td><td colspan="2"></td><td colspan="4">10 Hours</td></tr>
<tr><td colspan="9">Introduction: Need for Software Engineering, Professional Software Development, Software Engineering Ethics, Software Engineering Practice-Essence of Practice, General Principles Software Development Life Cycle
Models: Waterfall Model – Classical Waterfall Model, Iterative Waterfall Model, Evolutionary model-Spiral, Prototype.</td></tr>
<tr><td>Module 2</td><td>Software Requirements, Analysis and Design
(Comprehension level)</td><td>Assignment</td><td colspan="2">Development of SRS documents for a given scenario</td><td colspan="4">12 Hours</td></tr>
<tr><td colspan="9">Requirements Engineering: Eliciting requirements, Functional and non- Functional requirements, Software Requirements Specification (SRS), Requirement Analysis and validation. Requirements modelling-Introduction to Use Cases, Activity diagram and Swim lane diagram. CASE support in Software Life Cycle, Characteristics of CASE Tools, Architecture of a CASE Environment.
Design: Design concepts, Architectural design, Component based design, User interface design.</td></tr>
<tr><td>Module 3</td><td>Agile Principles & Devops
(Knowledge level)</td><td>Quiz</td><td colspan="2"></td><td colspan="4">10 Hours</td></tr>
<tr><td colspan="9">Agile: Scrum Roles and activities, Sprint Agile software development methods - Scaling, User Stories, Agile estimation techniques, Product backlogs, Stake holder roles, Dynamic System Development Method.
Devops: Introduction, definition, history, tools.</td></tr>
<tr><td>Module 4</td><td>Software Testing and Maintenance
(Application Level)</td><td>Assignment</td><td colspan="2">Apply the testing concepts using Programing</td><td colspan="4">13 Hours</td></tr>
<tr><td colspan="9">Software Testing-verification and validation, Test Strategies - White Box Testing, Black box Testing. Automation Tools for Testing.
Software Quality Assurance-Elements of software quality assurance, SQA Tasks, Goals and Metrics, Software configuration management- SCM process, SCM Tools (GitHub).
Maintenance- Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models.</td></tr>
</table>

Targeted Application & Tools that can be used: Selenium, GitHub, CASE Tools
Text Book R1. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", VII Edition, McGraw-Hill, 7. R2. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", VI Edition, McGraw-Hill, 8.
References 1. Rajib Mall, "Fundamentals of Software Engineering", VI Edition, PHI learning private limited, 2015. 2. Ian Sommerville, "Software Engineering", IX Edition, Pearson Education Asia, 2011. 3. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002
Topics Relevant to "Skill Development: Balck box Testing, White box Testing, Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout

Course Code: CSE2272	**Course Title:** Cloud computing **Type of Course:** Theory		**L- T-P- C**	3	**0**	0	**3**
Version No.	2.0						
Course Pre-requisites							
Anti-requisites	**NIL**						
Course Description	This Course is designed to introduce the concepts of Cloud Computing as a new computing paradigm. Cloud Computing has emerged in recent years as a new paradigm for hosting and delivering services over the Internet. The students can explore various Cloud Computing terminology, principles and applications. Understanding different views of the Cloud Computing such as theoretical, technical and commercial aspects. Topics include: Evolution of cloud computing and its services available today, Introduction, Architecture of cloud computing, Infrastructure, platform, software, Types of cloud, Business models, cloud services, Collaborating using cloud services, Virtualization for cloud, Security, Standards and Applications.						
Course Objective	The objective of the course is to familiarize the learners with the concepts of Could computing and Virtualization and attain Employability through Participative Learning techniques.						
Course Outcomes	On successful completion of the course the students shall be able to: · Describe fundamentals of cloud computing, virtualization and cloud computing services. · Discuss high-throughput and data-intensive computing. · Explain security and standards in cloud computing. · Demonstrate the installation and configuration of virtual machine.						
Course Content:							
Module 1	Introduction to Cloud and Virtualization	Assignment	Virtualization	**10 Sessions**			
Topics: Introduction to Cloud and Virtualization Cloud Computing at a Glance, Historical Developments, Building Cloud Computing Environments, Computing Platforms and Technologies, Virtualization, Characteristics of Virtualized Environments Taxonomy of Virtualization Techniques, Virtualization and Cloud Computing, Technology Examples, Cloud Computing Architecture, IaaS, PaaS, SaaS, Types of Clouds, Economics of Cloud							
Module 2	High Throughput and Data Intensive Computing	Assignment	Virtualization	**10 Sessions**			
Topics: High Throughput and Data Intensive Computing: Task computing, MPI applications, Task based programming, Introduction to DIC, Technologies for DIC, Aneka Map Reduce Programming.							
Module 3	Cloud Security and Standards	Assignment	Virtualization	**9 Sessions**			
Topics: Cloud Security and Standards: Cloud Security Challenges, Software-as-a-Service Security, Application standards, Client standards, Infrastructure and Service standards.							

Module 4	Cloud Platforms	Assignment	Virtualization	**9 Sessions**
Cloud Platforms, Advances in cloud: introduction to Amazon Web Services: Introduction to Google App Engine, Introduction to Microsoft Azure. Media Clouds - Security Clouds - Computing Clouds - Mobile Clouds – Federated Clouds – Hybrid Cloud				
Targeted Application & Tools that can be used:				
Text Book(s): 1. John Rittinghouse and James Ransome, "Cloud Computing, Implementation, Management and Security", CRC Press. 2. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "Mastering Cloud Computing", McGraw Hill Education.				
Reference(s): 1. David E.Y. Sarna, "Implementing and Developing Cloud Applications", CRC Press. 2. Anthony T Velte, Toby J Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach", Tata McGraw-Hill. Web resources: https://presiuniv.knimbus.com/user#/home				

<table>
<tr><td>Course Code:
CSE2273</td><td colspan="2">Course Title: Cloud computing
Type of Course : Lab</td><td>L- T-P- C</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td>Version No.</td><td colspan="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7"></td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">This course is designed to give hands-on experience with cloud platforms, services, and deployment models. Students will learn to set up, configure, and manage cloud environments using platforms like AWS, Microsoft Azure, and Google Cloud. The course covers virtualization, containerization, serverless computing, cloud storage, security, and scalability. Through practical assignments, students will develop skills in deploying cloud applications, managing cloud resources, automating cloud workflows, and implementing cost-effective cloud solutions.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is to Understand Cloud Infrastructure, deploy and manage virtual machines, implement cloud storage, develop and deploy cloud applications, optimize cost and performance.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course the students shall be able to:
CO1· Deploy and Manage Cloud Resources.
CO2. Develop and Deploy Cloud-based Applications
CO3. Optimize Performance and Cost in the Cloud
CO4. Implement Security and Automation in Cloud Environments</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Introduction to Cloud and Virtualization</td><td>Assignment</td><td colspan="3">Virtualization</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="8">Lab Assignment 1: Setting Up Virtual Machines on Cloud
Create a Virtual Machine (VM) on AWS/Azure/GCP
Configure OS, storage, and network settings
Connect to the VM using SSH/RDP
Install web server (Apache/Nginx) and deploy a static webpage
Lab Assignment 2: Containerization Using Docker
Install Docker on a local or cloud VM
Create and run a Docker container
Build a custom Docker image with a simple Python/Node.js application
Push the image to Docker Hub and deploy it on a new VM</td></tr>
<tr><td>Module 2</td><td>High Throughput and Data Intensive</td><td>Assignment</td><td colspan="3">Virtualization</td><td colspan="2">10 Sessions</td></tr>
</table>

	Computing			

Lab Assignment 1: Setting Up a Distributed Computing Environment

- Launch a **Hadoop or Spark cluster** on AWS EMR / Azure HDInsight / Google Dataproc
- Configure **HDFS (Hadoop Distributed File System)** for big data storage
- Run a **basic MapReduce job** on sample data

Lab Assignment 2: Data Preprocessing with Cloud Storage

- Store large datasets in **Amazon S3 / Azure Blob Storage / Google Cloud Storage**
- Use **Apache Spark or Hadoop** to read, clean, and process data
- Convert datasets into **Parquet or Avro formats** for efficient storage

Lab Assignment 3: Batch Processing with Apache Spark

- Load large datasets (e.g., logs, tweets, transaction data) into **Spark DataFrame**
- Perform **ETL (Extract, Transform, Load) operations** on the data
- Use **SparkSQL** for querying large datasets

Lab Assignment 4: Real-Time Data Processing with Spark Streaming

- Set up **Kafka / AWS Kinesis / Google Pub/Sub** for real-time data ingestion
- Process streaming data using **Spark Streaming**
- Perform windowed aggregations and visualize real-time trends

Lab Assignment 5: Cloud-Based Machine Learning with Big Data

- Use **Google BigQuery ML / AWS SageMaker / Azure Machine Learning** for model training
- Train a **linear regression or classification model** on a large dataset
- Deploy the trained model as an API for real-time predictions

Lab Assignment 6: Running Parallel Machine Learning Workloads

- Implement **distributed ML training** using **Spark MLlib or TensorFlow on Cloud TPUs**
- Train models on a large dataset and optimize performance using distributed execution

Lab Assignment 7: Auto-Scaling and Load Balancing for Data Processing

- Deploy a **serverless Spark job** using **AWS Glue / Azure Synapse**
- Implement **auto-scaling for high-throughput jobs**
- Measure performance improvements using cloud monitoring tools

Lab Assignment 8: Cost Optimization for High-Throughput Data Processing

- Analyze **cloud cost reports** for data-intensive workloads
- Optimize cloud storage and compute resources for cost-efficiency
- Compare **on-demand vs. reserved vs. spot instances** for cost savings

Module 3	Cloud Security and Standards	Assignment	Virtualization	**9 Sessions**

Lab Assignment 9: Configuring Identity and Access Management (IAM)

- Set up **IAM roles and policies** in AWS / Azure / Google Cloud
- Create and assign users with **least privilege access**
- Implement **Multi-Factor Authentication (MFA)** for added security
- Audit IAM policies using **AWS IAM Access Analyzer / Azure Security Center**

Lab Assignment 10: Setting Up Single Sign-On (SSO) and Role-Based Access Control (RBAC)

- Configure **AWS Cognito / Azure Active Directory / Google IAM** for authentication
- Implement **Role-Based Access Control (RBAC)** for users and groups
- Integrate **OAuth 2.0 / OpenID Connect (OIDC) / SAML** for secure authentication

Lab Assignment 11: Encrypting Data at Rest and in Transit

- Encrypt **cloud storage (S3, Blob, Cloud Storage)** using **KMS (Key Management Service)**
- Set up **TLS/SSL certificates** for secure web traffic encryption
- Enable **database encryption (AWS RDS, Azure SQL, GCP Cloud SQL)**

Lab Assignment 12: Implementing Compliance & Governance in Cloud

- Enable **GDPR, HIPAA, ISO 27001 compliance tools** in cloud platforms
- Use **AWS Config / Azure Policy / GCP Security Command Center** to enforce compliance
- Conduct **security audits and generate compliance reports**

Lab Assignment 13: Implementing Cloud Monitoring & Threat Detection

- Configure **AWS CloudTrail / Azure Monitor / GCP Operations Suite** for activity logging
- Set up **intrusion detection systems (IDS) & anomaly detection**
- Analyze security logs using **Amazon GuardDuty / Azure Sentinel / Chronicle Security**

Lab Assignment 14: Automating Security Incident Response

- Deploy a **Serverless Lambda / Azure Logic App** to automatically respond to security incidents
- Implement **automated alerts** for suspicious activity
- Test a **denial-of-service (DDoS) simulation** and implement mitigation strategies

Module 4	Cloud Platforms	Assignment	Virtualization	**9 Sessions**

Lab Assignment 15: Getting Started with Cloud Platforms

- Create a **free-tier account** on AWS, Azure, or Google Cloud
- Navigate the **Cloud Console, CLI, and SDKs**
- Explore and configure **dashboard, billing, and IAM settings**

Lab Assignment 16: Launching a Virtual Machine (VM) on Cloud

- Deploy a **VM instance** using **AWS EC2, Azure Virtual Machines, or Google Compute Engine**

Configure **OS, storage, networking, and security groups**

Connect to the instance using **SSH (Linux) or RDP (Windows)**

Lab Assignment 17: Cloud Storage and File Management

- Create **Object Storage (AWS S3 / Azure Blob Storage / Google Cloud Storage)**
  Upload, download, and set access permissions for files
  Implement **Lifecycle Policies and Versioning**

Lab Assignment 18: Cloud Database Management

- Deploy a **Relational Database (AWS RDS / Azure SQL Database / Cloud SQL)**
  Connect and query the database using **MySQL/PostgreSQL clients**
  Set up database **backups and automatic scaling**

Lab Assignment 19: Configuring Virtual Networks in Cloud

- Set up a **Virtual Private Cloud (VPC) / Azure Virtual Network / GCP VPC**
  Configure **subnets, firewalls, and security groups**
  Test **network communication between two VMs**

Lab Assignment 20: Deploying a Web Application on Cloud

Deploy a Python/Node.js/Java web app using:

- **AWS Elastic Beanstalk**
- **Azure App Service**
- **Google App Engine**
  Connect the app to Cloud Database (RDS, CosmosDB, Firestore)
  Monitor application performance and logs

Targeted Application & Tools that can be used:

Text Book(s):

1. John Rittinghouse and James Ransome, "Cloud Computing, Implementation, Management and Security", CRC Press.
2. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "Mastering Cloud Computing", McGraw Hill Education.

Reference(s):

1. David E.Y. Sarna, "Implementing and Developing Cloud Applications", CRC Press.
2. Anthony T Velte, Toby J Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach", Tata McGraw-Hill. Web resources: https://presiuniv.knimbus.com/user#/home

<table>
<tr><td>Course Code:
CSE2505</td><td colspan="2">Course Title: Mobile Application Development
Type of Course: Theory</td><td>L- T-P- C</td><td>2</td><td>0</td><td>0</td><td>2</td></tr>
<tr><td>Version No.</td><td colspan="7">2.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">CSE2255</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">The course deals with the basics of android platform and application life cycle. The goal of the course is to develop mobile applications with Android containing at least one of the following phone material components: GPS, accelerometer or phone camera, use simple GUI applications and work with database to store data locally or in a server. Topics include user interface design; user interface building; input methods; data handling; network techniques and URL loading; GPS and motion sensing. Android application framework and deployment. Power management, Screen resolution, Touch interface, Store data on the device.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is to familiarize the learners with the concepts of Mobile Applications and Development as mentioned above and attain Employability Skills through Experiential Learning Techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course the students shall be able to:
1. Discuss the fundamentals of mobile application development and its architecture. (Comprehension)
2. Illustrate mobile applications with appropriate android view. (Application)
3. Demonstrate the use of services, broadcast receiver, Notifications and content provider.(Application)
4. Apply data persistence techniques, to perform CRUD operations. (Application) 5. Use advanced concepts for mobile application development. (Application)</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Introduction and Architecture of Android</td><td>Assignment</td><td colspan="3">Simulation/Data Analysis</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="8">Topics:
Android: History and features, Architecture, Development Tools, Android Debug Bridge (ADB), and Life cycle.</td></tr>
<tr><td>Module 2</td><td>User Interfaces, Intent and Fragments</td><td>Term paper/Assignment</td><td colspan="3">Simulation/Data Analysis</td><td colspan="2">15 Sessions</td></tr>
<tr><td colspan="8">Topics:
Views, Layout, Menu, Intent and Fragments.</td></tr>
<tr><td>Module 3</td><td>Components of Android</td><td>Term paper/Assignment</td><td colspan="3">Simulation/Data Analysis</td><td colspan="2">15 Sessions</td></tr>
</table>

Topics: Activities, Services, Broadcast receivers, Content providers, User Navigation				
Module 4	Notifications and Data Persistence	Term paper/Assignment	Simulation/Data Analysis	15 Sessions
Topics: Notification, Shared Preferences, SQLite database, Android Room with a View, Firebase.				
Module 5	Advance App Development	Term paper/Assignment	Simulation/Data Analysis	15 Sessions
Topics: Graphics and Animation, App Widgets, Sensors, Performance, Location, Places, Mapping, Custom Views, Canvas.				
Targeted Application & Tools that can be used: Applications: Native Android Applications Native iOS Applications Cross Platform mobile Apps Mobile web Applications				
Text Book(s): T1. Pradeep kothari "Android Application Development - Black Book", dreamtechpress T2. Barry Burd (Author), "Android Application Development" ALL – IN – ONE FOR Dummies T3. Jeff Mcherter (Author),Scott Gowell (Author), "Professional mobile Application Development" paperback, Wrox - Wiley India Private Limited T4. Wei-Meng Lee (Author) "Beginning Android Application Development" Wrox – Wiley India Private Limited				
Reference(s): 1. Bill Phillips, Chris Stewart, and Kristin Marsicano (Author) "Android Programming" 3rd edition, 2017.The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 5. The Big Nerd Ranch Guide, by" 2. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd, 2014. 3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O'Reilly SPD Publishers, 2015. 4. J F DiMarzio, "Beginning Android Programming with Android Studio", 4th Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978-8126565580 5. Anubhav Pradhan, Anil V Deshpande, " Composing Mobile Apps" using Android, Wiley 2014, ISBN: 978-81-265-4660-2 6. Reto Meier "Professional Android Application Development" E-Resources: https://puniversity.informaticsglobal.com/login Or http://182.72.188.193/				

Course Code: **CSE2506**	**Course Title:** Mobile Application Development Lab **Type of Course:** Lab		**L- T-P- C**	0	**0**	4	2
Version No.	2.0						
Course Pre-requisites	CSE2256						
Anti-requisites	**NIL**						
Course Description	The course provides hands-on experience in designing, developing, and deploying mobile applications for Android and iOS platforms. Students will work with native development frameworks such as Android Studio (Java/Kotlin) and Xcode (Swift), as well as explore cross-platform tools like Flutter or React Native.						
Course Objective	The objective of the course is to develop Native and Cross-Platform Mobile Applications, design Interactive and Responsive User Interfaces, integrate Backend Services and APIs, implement State Management and Performance Optimization, ensure Mobile App Security and Data Protection						
Course Outcomes	On successful completion of the course the students shall be able to: 1. Develop Functional Mobile Applications 2. Design and Implement Interactive UIs 3. Integrate Cloud Services and APIs 4. Integrate Backend Systems and Data Management 5. Deploy, Publish, and Maintain advanced Mobile Application						
Course Content:							
Module 1	Introduction and Architecture of Android	Assignment	Simulation/Data Analysis	10 Sessions			
1.a. Design an app to read user inputs using edit text and display the result of arithmetic operations using toast message. 1.b. Create an android app to calculate the current age of yourself, select your DOB using date picker. 2. Design an app to input your personal information. Use an autocomplete text view to select your place of birth.							
Module 2	User Interfaces, Intent and Fragments	Term paper/Assignment	Simulation/Data Analysis	15 Sessions			
3. a. Design an app to select elective course using spinner view and on click of the display button, toast your ID and selected elective course. 3. b. Design a restaurant menu app to print the total amount of orders.							

<table>
<tr><td>Module 3</td><td>Components of Android</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>15 Sessions</td></tr>
<tr><td colspan="5">4. Develop an android app that uses intent to maintain the following scenario.
Check the eligibility criteria for voting. Input the Aadhar no., Name & age in the first activity. If the age is above 18, display the voter's detail in the second activity. Else, display, “You are not eligible to vote” in the second Activity.
5. Demonstrate the use of fragment with list of buttons representing various colors, and on click of these buttons, the appropriate color is filled in the next fragment. Create an Android application to input the vitals of a person (temperature, BP). If the vitals are abnormal, give proper notification to the user.
6. Create an android app to for movie ticket booking. Save the user name of the customer using shared preferences. After completion of booking, retrieve the username from the shared preferences and print the ticket details.</td></tr>
<tr><td>Module 4</td><td>Notifications and Data Persistence</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>15 Sessions</td></tr>
<tr><td colspan="5">7. Create an android application to manage the details of students' database using SQLite.Use necessary UI components, which perform the operations such as insertion, modification, removal and view.Presidency University needs an APP for Admission eligibility checking for students, for that you need to take the following information from the Student: registration ID, physics, chemistry and mathematics marks (PCM), fees is allotted as below criteria.
PCM (Total marks %) Fee concession
90 above 80 %
70 to 89 60 %
Below 69 % no concession
On click on the button “Registration” details should be stored in the database using SQLite. Create button DISPLAY ALL (full students list) on click on the button it should display the students list per the fee
concession.
8. A company need to design an app that plays soft music automatically in the background. Create an app to achieve this functionality.
9. Create an android application such that your view object in the Activity can be Animated with fade-in effect. Create an appropriate XML file named fade-in and write the application to perform the property animation.</td></tr>
<tr><td>Module 5</td><td>Advance App Development</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>15 Sessions</td></tr>
<tr><td colspan="5">10. Demonstrate how to send SMS and email.
11. Create an android application to transfer a file using WiFi. Create an android application “Where am I” with an Activity that uses the GPS Location provider to find the device’s last known location.</td></tr>
<tr><td colspan="5">Targeted Application & Tools that can be used:
Applications:</td></tr>
</table>

- Native Android Applications (Java/Kotlin)
    - Android Mobile Apps built for Android smartphones and tablets using Java or Kotlin programming languages.
    - Target audience: Android users.
- Native iOS Applications (Swift)
    - iOS Mobile Apps designed for iPhone and iPad using Swift.
    - Target audience: iOS users (Apple ecosystem).
- Cross-Platform Mobile Apps (Flutter, React Native)
    - Cross-platform apps designed to run on both Android and iOS from a single codebase using frameworks like Flutter or React Native.
    - Target audience: Users on both Android and iOS platforms.
- Mobile Web Applications (Progressive Web Apps - PWA)
    - Mobile-optimized web applications using HTML5, CSS3, and JavaScript that run in a browser with native-like functionality (offline support, push notifications).
    - Target audience: Users accessing apps via mobile browsers.

Development Tools and Frameworks

- Integrated Development Environments (IDEs)
    - Android Studio (for Android): The official IDE for Android development, supporting Java, Kotlin, and Android SDK.
    - Xcode (for iOS): The official IDE for iOS development with Swift and Objective-C, providing a comprehensive suite of development tools for iPhone/iPad applications.
    - Visual Studio Code (VS Code): Lightweight IDE for working with Flutter, React Native, and web development projects.
- Cross-Platform Development Frameworks
    - Flutter: Open-source UI framework by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.
    - React Native: Open-source framework developed by Facebook for building cross-platform apps with JavaScript and React.
- Backend & Cloud Tools
    - Firebase: Google's backend-as-a-service (BaaS) platform offering authentication, real-time databases, cloud storage, and push notifications for mobile apps.
    - AWS Amplify: Cloud platform for backend services (API, storage, authentication) and mobile deployment.
    - SQLite / Realm: Local storage solutions for mobile apps to manage data storage and retrieval on-device.
- Mobile App Testing and Debugging Tools
    - Android Emulator (for Android): A virtual device to run and test Android apps without needing physical devices.
    - Xcode Simulator (for iOS): A tool to simulate different iOS devices and test apps during development.
    - Appium: Open-source tool for automated testing across native, hybrid, and mobile web applications.
- Version Control and Collaboration
    - Git: Version control system for managing code changes and collaborating with teams.

    - GitHub / GitLab / Bitbucket: Online platforms for hosting Git repositories, collaboration, and version control management.
- Mobile App Deployment Tools
    - Google Play Console: For managing Android app publishing, distribution, and monitoring.
    - Apple App Store Connect: For managing iOS app submissions, reviews, and releases on the Apple App Store.
- UI/UX Design Tools
    - Figma / Adobe XD: Tools for UI/UX design and wireframing to create the visual elements of mobile applications before development.
    - Sketch: Vector-based design tool for iOS UI design and prototyping

Text Book(s):

T1. Pradeep kothari "Android Application Development - Black Book", dreamtechpress
T2. Barry Burd (Author), "Android Application Development" ALL – IN – ONE FOR Dummies
T3. Jeff Mcherter (Author),Scott Gowell (Author), "Professional mobile Application Development" paperback, Wrox - Wiley India Private Limited
T4. Wei-Meng Lee (Author) "Beginning Android Application Development" Wrox – Wiley India Private Limited

Reference(s):

1. Bill Phillips, Chris Stewart, and Kristin Marsicano (Author) "Android Programming" 3rd edition, 2017.The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 5. The Big Nerd Ranch Guide, by"
2. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd, 2014.
3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O'Reilly SPD
Publishers, 2015.
4. J F DiMarzio, "Beginning Android Programming with Android Studio", 4th Edition, Wiley India Pvt
Ltd, 2016. ISBN-13: 978-8126565580
5. Anubhav Pradhan, Anil V Deshpande, " Composing Mobile Apps" using Android, Wiley 2014, ISBN: 978-81-265-4660-2
6. Reto Meier "Professional Android Application Development"
E-Resources: https://puniversity.informaticsglobal.com/login Or http://182.72.188.193/

urse Code: **E2274**	**urse Title:** Competitive Programming and Problem Solving **pe of Course: Program Core**	-P-C	0	0		
rsion No.						
urse Pre-requisites	L					
ti-requisites	L					
urse Description	The **Competitive Programming and Problem Solving** course equips students with efficient problem-solving skills for coding competitions and real-world challenges. Starting with brute-force solutions, students learn to optimize time and space complexity using advanced techniques like dynamic programming, greedy algorithms, and backtracking. Hands-on practice on platforms like CodeChef and Codeforces helps tackle problems involving number theory, data structures, and algorithmic paradigms. By understanding CP constraints and fostering a strategic mindset, students gain the confidence to excel in competitions, technical interviews, and practical applications.					
urse Out Comes	On successful completion of the course the students shall be able to: CO1 : Understanding the issues of online platforms and Competitive Programming (CP) and developing brute force coding for commonly asked CP problems. CO2 : Analyzing the space and time complexity of brute force solutions and designing efficient solutions. CO3 : Evaluating the applicability of suitable algorithmic approaches to solve relevant CP problems. CO4: Creating efficient solutions of CP problems using the learnt algorithmic approaches.					
urse Objective	The objective of the course is to familiarize the learners with the concepts of **Competitive Programming and Problem Solving** and attain **Skill Development** through **Experiential Learning** techniques.					

dule 1: Introduction to Competitive Programming

erview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforces etc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL

dule 2: Number Theory for Problem-Solving

e of Number Theory for problem-solving: reducing time/space complexity of brute force coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient coding for Permutation Combination; XORing based and pattern-based solutions.

dule 3: Optimizing Time & Space Using Sequential Storage

ding for Optimizing time and Space using Sequential Storage: two pointer approach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string matching algorithms; Kadane's algo, stacks, priority-queues and hashing based efficient coding; median based problems and alternate solutions.

dule 4: Non-Linear Data Structures

plying Non-Linear Data Structures for real-life problems: design of efficient solutions for problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path algos for CP problems with reduced time/space complexity.

dule 5: Problem Solving using Advanced Topics

Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

t of Laboratory Tasks:

1. You are given the finishing times of 'N' runners in a marathon. Write a program to find the runner who finished in the third position. **Focus:** Basic data structures (arrays), sorting algorithms (e.g., insertion sort, selection sort), and basic input/output.
2. In the same marathon, you are given the finishing times of 'N' runners and their bib numbers. Write a program to efficiently find the top 10 runners and their corresponding bib numbers. **Focus:** Efficient sorting algorithms (e.g., merge sort, quick sort), data structures like priority queues, and optimizing for large datasets.
3. A library maintains a list of books with their unique IDs. Write a program to check if a given book ID is present in the library. Focus: Searching algorithms (linear search), basic data structures (arrays or lists).
4. The library wants to implement a system to quickly find books by their titles. Suggest an efficient data structure (e.g., a hash table or a trie) and explain how to implement it to achieve fast book lookups. Focus: Understanding the trade-offs between different data structures, choosing the most appropriate data structure for a specific problem, and implementing efficient search operations.
5. An online store sells products with different prices. Write a program to calculate the total cost of a given list of products. **Focus:** Basic arithmetic operations, working with arrays or lists to store product prices.
6. The online store offers discounts based on the total purchase amount. Design an algorithm to efficiently calculate the final cost of an order, considering different discount rules (e.g., percentage discounts, fixed amount discounts, tiered discounts). Focus: Algorithmic design, conditional statements, handling complex scenarios with multiple rules, and potentially using dynamic programming techniques for optimization.
7. You are given two integers, 'a' and 'm'. Calculate 'a' raised to the power 'm' modulo a large prime number 'p'. **Focus:** Basic modular arithmetic operations (modular exponentiation), understanding the modulo operator.
8. In a secure communication system, you need to efficiently compute the modular exponentiation for very large values of 'm'. Implement and analyze the efficiency of the binary exponentiation algorithm for this task. **Focus:** Efficient algorithms for modular exponentiation (binary exponentiation), time complexity analysis, and understanding the importance of efficient algorithms in cryptography.
9. You have a deck of 'N' cards. Calculate the total number of possible hands of size 'K' that can be drawn from the deck. **Focus:** Basic combinatorics (combinations), factorial calculations.
10. In a card game, you need to calculate the probability of drawing certain combinations of cards (e.g., a pair, a three-of-a-kind) from a shuffled deck. Design an efficient algorithm to calculate these probabilities. **Focus:** Advanced combinatorics (permutations and combinations with repetitions), probability calculations, and optimizing calculations to avoid overflows.
11. You are given a network of devices represented as a graph. Determine if there is a path between two given devices in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search).
12. In a secure network, you need to detect and isolate compromised devices. Design an algorithm that efficiently identifies devices that exhibit anomalous behavior (e.g., unusual traffic patterns) using XOR-based techniques for data comparison and pattern matching. **Focus:** Applying XOR operations for data comparison and pattern recognition, understanding the properties of XOR (e.g., commutative, associative), and designing algorithms for network anomaly detection.
13. You are given an array representing the speeds of cars on a highway. Find the minimum time required for all cars to pass a certain point. **Focus:** Basic array traversal, finding the minimum element in an array.
14. In a more realistic scenario, cars have different lengths. Implement a two-pointer approach to simulate the movement of cars and determine the minimum time for all cars to pass a given point. **Focus:** Two-pointer technique, simulating real-world scenarios with arrays, optimizing time complexity.
15. Given a string, find the number of occurrences of a specific substring within the string. **Focus:** Basic string manipulation, string matching (brute-force approach).
16. Implement the KMP (Knuth-Morris-Pratt) string matching algorithm to efficiently find all occurrences of a given pattern within a large text document. **Focus:** Advanced string matching algorithms, understanding the concept of the "next" array in KMP, optimizing for large input sizes.

17. An online auction platform receives bids for different items. Implement a data structure (e.g., a priority queue) to efficiently track the highest bid for each item. **Focus:** Priority queues, insertion and extraction operations on priority queues, basic implementation of a priority queue using an array or a suitable library.
18. The auction platform needs to handle a large number of bids concurrently. Design and implement a system that efficiently processes bids, updates the highest bid for each item, and handles potential race conditions. **Focus:** Concurrent data structures and algorithms, thread safety, handling race conditions, optimizing for high-throughput scenarios.
19. A social network can be represented as a graph where users are nodes, and connections between users are edges. Write an algorithm to find if two given users are connected in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search), basic graph representation (adjacency list or adjacency matrix).
20. In a large social network, efficiently finding the shortest path between two users is crucial. Implement Dijkstra's algorithm to find the shortest paths between users in the network, considering edge weights (e.g., representing the strength of connections). **Focus:** Shortest path algorithms (Dijkstra's algorithm), graph algorithms with weighted edges, optimizing for large graphs.
21. A file system can be modeled as a tree structure. Implement a function to traverse the file system and print the names of all files and directories. **Focus:** Tree traversal algorithms (depth-first search or breadth-first search), basic tree representation (using nodes and pointers).
22. Design and implement a file system that supports efficient operations like creating directories, deleting files, and finding files based on their names or paths. Consider using a combination of tree structures and hash tables for efficient indexing and searching. **Focus:** Designing and implementing file system structures, using multiple data structures together, optimizing for common file system operations.
23. An online shopping cart can be represented as a tree, where each node represents an item or a category of items. Write an algorithm to calculate the total price of all items in the shopping cart. **Focus:** Tree traversal, calculating sums within a tree structure.
24. Implement a system that allows customers to apply discounts and coupons to their shopping carts. Consider using a combination of trees and other data structures (e.g., hash tables) to efficiently apply discounts and calculate the final price. **Focus:** Applying discounts and promotions to tree-like structures, efficient implementation of discount rules, optimizing for complex pricing scenarios.
25. In a social network, users can form groups. Given a list of friendships, determine if all users in a specific group are connected (directly or indirectly) through friendships. **Focus:** Disjoint set union (DSU) data structure, basic connectivity checks.
26. Design an efficient algorithm to find the minimum number of new friendships needed to connect all users in the social network into a single, connected component. **Focus:** Applying DSU for finding connected components, greedy algorithms, optimization for minimizing connections.
27. A treasure hunt involves a series of clues leading to the final treasure. Given a list of possible paths and their associated costs, find the cheapest path to reach the treasure. **Focus:** Greedy algorithms (e.g., Dijkstra's algorithm for shortest paths), basic graph representation.
28. In a more complex treasure hunt, there are time constraints associated with each path. Design an algorithm to find the fastest path to the treasure while considering both path costs and time constraints.

Focus: Combining greedy approaches with other techniques (e.g., priority queues), handling multiple constraints, optimizing for time-critical scenarios.

29. In a simplified chess game with only rooks, determine the minimum number of moves required for a rook to reach a specific target square on an empty board. **Focus:** Breadth-first search (BFS) on a graph (the chessboard), basic graph traversal.
30. In a more realistic chess game with multiple pieces and obstacles, implement a minimax algorithm with alpha-beta pruning to determine the best move for a player. **Focus:** Game tree search, minimax algorithm, optimization techniques like alpha-beta pruning, handling complex game states.

rgeted Application & Tools that can be used:

1. C or C++ Compiler (g++): The standard compiler for CP. Familiarize students with compilation flags

(e.g., -O2 for optimization).

2. IDE (Integrated Development Environment): Code:: Blocks, Visual Studio, CLion, or similar IDEs. These provide debugging capabilities, code completion, and other helpful features.
3. Online Judges (CodeChef, Codeforces, LeetCode, HackerRank): Essential for practicing and submitting solutions.
4. Debugger (gdb): Crucial for understanding code execution and finding bugs. Origin, excel and Mat lab soft wares for programming and data analysis.
5. Number Theory Libraries: Some libraries provide pre-built functions for number theory operations (though often it's better to implement them yourself for learning).
6. Wolfram Alpha: A useful tool for verifying number theory calculations and exploring concepts.
7. **String Libraries:** Familiarize students with the string manipulation functions available in C++.
8. **Graph Visualization Tools:** Tools like Graphviz can be helpful for visualizing graphs and understanding graph algorithms.
9. **DP Debugging Techniques:** Practice debugging DP solutions, as they can be complex. Visualizing the DP table can be helpful.

xt Books:

1 **Guide to Competitive Programming: Learning and Improving Algorithms Through Contests" (3rd Edition),** ***Antti Laaksonen, springer, 2024***

2 **"Data Structures and Algorithms in Java: A Project-Based Approach" –** ***Dan S. Myers, Cambridge University Press***

Reference Books:

1. **Data Structures and Algorithmic Thinking with Python/C++/Java",** ***Narasimha Karumanchi, 5th Edition, Career Monk, 2017.***
2. **Introduction to Algorithms, Thomas H. Cormen (Author), Charles E. Leiserson (Author), Ronald L. Rivest , fourth edition April 2022**

Web Resources

1. **https://nptel.ac.in/courses/106106231**

ject work/Assignment: Mention the Type of Project /Assignment proposed for this course

Assessment Type

- Midterm exam
- Assignment (review of digital/ e-resource from PU link given in references section -mandatory to submit screen shot accessing digital resource.)
- Quiz
- End Term Exam
- Self-Learning

Course ode: APT4005	**Course Title: Aptitude For Employability** **Type of Course: Practical Only**	**L- T-P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Students should have the basic concepts of Quantitative aptitude, Verbal ability along with its applications in real life problems.					
Anti-requisites	Nil					
Course Description	This course is designed to enable the students to enhance their skills in quantitative aptitude and verbal ability skills.					
Course Objective	The objective of the course is to familiarize the learners with concepts in Quantitative Aptitude and Verbal ability through problem solving techniques suitable for their career development.					
Course Outcomes	On successful completion of the course the students shall be able to: CO1] **Recall** all the basic mathematical concepts CO2] **Identify** the principle concept needed in a question CO3] **Solve** the quantitative and logical ability questions with the appropriate concept.					
Course Content:						

Module 1	Quantitative Ability	Lab-10hrs		Platform Assessment-10hrs	**20 Hours**
Topics: Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss, Time Speed and Distance, Simple Interest and Compound Interest, Probability, Permutation and Combination.					
Module 2	Verbal Ability	Lab-5hrs		Platform Assessment-5hrs	**10 Hours**
Topics: - Parts of Speech, Subject Verb Agreement, Spotting Error, Cloze Test, Verbal Analogies, Reading Comprehension, Idioms & Phrases, Para Jumbles					

Targeted Application & Tools that can be used:
Application area: Placement activities and Competitive examinations. Tools: LMS

Evaluation	**Continuous Evaluation** Topic wise evaluation

Text Book
1. Fast track objective by Rajesh Verma
2. R S Aggarwal
3. S.P Bakshi

References
1. www.indiabix.com
2. www.testbook.com
3. www.youtube.com/c/TheAptitudeGuy/videos

Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.

Course Code: PPS3018	**Course Title: Preparedness for Interview** Type of Course: Practical Only Course	L- T- P- C	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.					
Anti-requisites	NIL					
Course Description	This course is designed to enable students to understand soft skills concepts to be corporate ready. The modules are set to improve self-confidence, communicate effectively and Prepare for the Interview to assist in employability. It helps the students to get a glimpse of the acceptable corporate readiness and equip them with the fundamental necessities of being able to confidently deal with the highly competitive corporate environment and helps in crafting different types of resumes. The pedagogy used will be group discussions, flipped classrooms, continuous feedback, role-play and mentoring.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of "**Preparing for Interview**" and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.					
Course Out Comes	**On successful completion of this course the students shall be able to:** CO1: **Develop** professional Resumes CO2: **Illustrate** Resumes effectively CO3: **Apply** skills and knowledge learnt for active and effective Group Discussions and Interview					

Course Content:

Module 1	**Resume Building**	**Classroom activity**		10 Hours

Topics: Resume structure, use of templates, Do's and Don'ts, ATS methods, Cover Letter and Video Resume. **Activity:** Real world scenarios

Module 2	**Group Discussion**	**Mock G D**		9 Hours

Topics: -Group discussion as a placement process, GD techniques like Keyword. SPELT & POV of affected parties. Do & Don't of GD, Case-lets and topics for GD, practice session and evaluation
Activity:- Real world scenarios

Module 3	**Personal Interview**	**Grooming checks + Evaluation + Mock Interview+ Role Play**		9 Hours

Topics: Placement process, Different interview rounds, HR interviews, Interview questions and desired answers, Different types of interviews, Do's and Don'ts.
Activity: - Role Play & Real-world scenario

Module 4	**Recap/Revision /Feedback Session**	**Practice sessions**		2 Hours

Targeted Application & Tools that can be used:
1. TED Talks
2. You Tube Links
3. Role Play activities

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

Continuous Individual Assessment

The Topics related to Skill Development:
Art Of Presentation and Group Discussion for **Skill Development** through **Participative Learning** Techniques. This is attained through assessment Component mentioned in course handout.

Course Code: CSE 7100	**Course Title:** Mini Project **Type of Course:**	**L- T-P- C**	0	0	0	4
Version No.	**1.0**					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learning techniques.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Identify the engineering problems related to local, regional, national or global needs. (Understand) 2. Apply appropriate techniques or modern tools for solving the intended problem. (Apply) 3. Design the experiments as per the standards and specifications. 4. (Analyze) 5. Interpret the events and results for meaningful conclusions. (Evaluate) 6. Appraise project findings and communicate effectively through scholarly publications. (Create)					

Course Code: PPS3018	**Course Title: Preparedness for Interview** Type of Course: Practical Only Course	L- T- P- C	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.					
Anti-requisites	NIL					
Course Description	This course is designed to enable students to understand soft skills concepts to be corporate ready. The modules are set to improve self-confidence, communicate effectively and Prepare for the Interview to assist in employability. It helps the students to get a glimpse of the acceptable corporate readiness and equip them with the fundamental necessities of being able to confidently deal with the highly competitive corporate environment and helps in crafting different types of resumes. The pedagogy used will be group discussions, flipped classrooms, continuous feedback, role-play and mentoring.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of "**Preparing for Interview**" and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.					
Course Out Comes	**On successful completion of this course the students shall be able to:** CO1: **Develop** professional Resumes CO2: **Illustrate** Resumes effectively CO3: **Apply** skills and knowledge learnt for active and effective Group Discussions and Interview					
Course Content:						
Module 1	**Resume Building**	**Classroom activity**		10 Hours		
Topics: Resume structure, use of templates, Do's and Don'ts, ATS methods, Cover Letter and Video Resume. **Activity:** Real world scenarios						
Module 2	**Group Discussion**	**Mock G D**		9 Hours		
Topics: -Group discussion as a placement process, GD techniques like Keyword. SPELT & POV of affected parties. Do & Don't of GD, Case-lets and topics for GD, practice session and evaluation **Activity:-** Real world scenarios						
Module 3	**Personal Interview**	**Grooming checks + Evaluation + Mock Interview+ Role Play**		9 Hours		
Topics: Placement process, Different interview rounds, HR interviews, Interview questions and desired answers, Different types of interviews, Do's and Don'ts. **Activity: -** Role Play & Real-world scenario						
Module 4	**Recap/Revision /Feedback Session**	**Practice sessions**		2 Hours		
Targeted Application & Tools that can be used: 1. TED Talks 2. You Tube Links 3. Role Play activities						
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course						
Continuous Individual Assessment						
The Topics related to Skill Development: Art Of Presentation and Group Discussion for **Skill Development** through **Participative Learning** Techniques. This is attained through assessment Component mentioned in course handout.						

<table>
<tr><td>Course Code: CSE 7300</td><td>Course Title: Capstone Project
Type of Course:</td><td>L- T-P- C</td><td>0</td><td>0</td><td>0</td><td>10</td></tr>
<tr><td>Version No.</td><td colspan="6">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td colspan="6">Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.</td></tr>
<tr><td>Course Objectives</td><td colspan="6">The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="6">On successful completion of this course the students shall be able to:
1. Identify problems based on societal /research needs. (Understand)
2. Apply Knowledge and skill to solve societal problems in a group. (Apply)
3. Develop interpersonal skills to work as member of a group or leader. (Apply)
4. Analyze the inferences from available results through theoretical /
5. Experimental / Simulations. (Analyze)
6. Analyze the impact of solutions in societal and environmental context for sustainable development. (Analyze)
7. Improve in written and oral communication. (Create)
8. Demonstrate capabilities of self-learning in a group, which leads to lifelong learning. (Understand)</td></tr>
</table>

Course Code: CSE3500	**Course Title: Intelligent Systems with Machine Learning** **Type of Course: PEC – Theory & Lab Integrated**	**L- T-P- C**	2	0	2	3

Version No.	1.0
Course Pre-requisites	CSE2267
Anti-requisites	**NIL**
Course Description	Machine Learning algorithms are the key to develop intelligent systems such as Apple's Siri, Google's self-driving cars etc. This course introduces the concepts of the core machine learning techniques such as Regression learning, Bayesian learning, Ensemble learning, Perceptron learning, Unsupervised learning, Competitive learning, learning from Gaussian mixture models and learning to detect outliers. Course lectures covers both the theoretical foundations as well as the essential algorithms for the various learning methods. Lab sessions complement the lectures and enable the students in developing intelligent systems for real life problems.
Course Objectives	This course is designed to improve the learners '<u>EMPLOYABILITY SKILLS</u>' by using <u>EXPERIENTIAL LEARNING</u> techniques. The supervised hands-on laboratory exercises, assessments and the group projects facilitate this learning process.
Course Out Comes	On successful completion of the course the students shall be able to: 1] Apply advanced supervised machine learning methods for predictive modeling. [Application] 2] Produce machine learning models with better predictive performance using meta learning algorithms [Application] 3] Create predictive models using Perceptron learning algorithms[Application] 4] Employ advanced unsupervised learning algorithms for clustering, competitive learning and outlier detection[Application] 5] Implement machine learning based intelligent models using Python libraries. [Application]
Course Content:	

Module 1	Supervised Learning	Assignment		Programming using Keras/Sklearn	**No. of Classes L – 7 P – 12**
	Topics: An overview of Machine Learning(ML); ML workflow; types of ML; Types of features, Feature Engineering -Data Imputation Methods; Regression – introduction; simple linear regression, loss functions; Polynomial Regression; Logistic Regression; Softmax Regression with cross entropy as cost function**; Bayesian Learning** – Bayes Theorem, estimating conditional probabilities for categorical and continuous features, Naïve Bayes for supervised learning; Bayesian Belief networks; Support Vector Machines – soft margin and kernel tricks.				
Module 2	Ensemble Learning	Assignment		Programming using Keras/Sklearn	**No. of Classes L-3 P-4**
	Topics: **Ensemble Learning** – using subset of instances – Bagging, Pasting, using subset of features –random patches and random subspaces method; Voting Classifier, Random Forest; Boosting – AdaBoost, Gradient Boosting, Extremely Randomized Trees, Stacking.				
Module 3	Perceptron Learning	Assignment /Quiz		Programming using Keras/Sklearn	**No. of Classes L-7 P -2**
	Topics: **Perceptron Learning** – from biological to artificial neurons, Perceptrons, Linear Threshold Units, logical computations with Perceptrons, common activation functions – sigmoid, tanh, relu and softmax, common loss functions, multi-layer Perceptrons and the Backpropagation algorithm using Gradient Descent.				
Module 4	Unsupervised Learning	Assignment		Programming using Keras/Sklearn	**No. of Classes L-6 P -6**

	Topics: **Unsupervised Learning** – simple k Means clustering- simple and mini-batch**;** updating centroids incrementally; finding the optimal number of clusters using Elbow method ; Silhoutte coefficient,drawbacks of kMeans,kMeans++ **;** Divisive hierarchical clustering – bisecting k-means, clustering using Minimum Spanning Tree (MST) **Competitive Learning** - Clustering using Kohenen's Self Organising Maps (SOM), **Density Based Spatial Clustering – DBSCAN**; clustering using Gaussian Mixture Models (GMM) with EM algorithm ; Outlier Detection methods – **Isolation Forest, Local Outlier Factor(LOF)**
	List of Laboratory Tasks: **Experiment N0 1: Methods for handling missing values** **Level 1:** Given a data set from UCI repository, implement the different ways of handling missing values in it using Scikit-learn library of Python **Level 2:** Implement one of these methods using a custom defined function in Python. **Experiment No. 2: Data Visualization** **Level 1** Perform Exploratory Data Analysis for a given data set by creating Scatter Plot, Pair Plot, Count Plot using Matplotlib and Seaborn **Level 2** Create Heat Maps, WordCloud **Experiment No. 3: Regression learning** **Level 1** Given a data set from UCI repository, implement the simple linear regression algorithm and estimate the models parameters and the performance metrics. Plot the learning curves. **Level 2** Implement the polynomial regression algorithm. Compare the learning curves of Polynomial and Linear Regression. **Experiment No.4: Logistic regression** **Level 1** Write custom code for generating the logistic/sigmoid plot for a given input **Level 2** Given a data set from UCI repository, implement the Logistic regression algorithm. Estimate the class probabilities for a given test data set. Plot and analyze the decision boundaries. **Experiment No.5: Bayesian Learning** **Level 1** Given a data set from UCI repository, implement a classification model using the Bayesian algorithm **Experiment No.6: Support Vector Machine(SVM)** **Level 1** Given data sets from UCI repository, implement a linear SVM and a non-linear SVM based classification model. **Experiment No. 7: Ensemble Learning** **Level 1** : Implement Ensemble Learning algorithms such as Bagging, Pasting and Out-of Bag Evaluation **Level 2** : Random Patches and Random Subspace Method **Experiment No. 8: Ensemble Learning** **Level 1** : AdaBoost and Gradient Boosting, Stacking **Experiment No. 9: Perceptron Learning** **Level 1** : Implement the Perceptron Classifier **Level 2** : – An Image Classifier Using the Sequential API of Keras **Experiment No. 10: Unsupervised Learning** **Level 1** : K-means – simple and mini-batch. Finding the optimal number of clusters using Elbow method and Silhoutte Coefficient . Compare the inertia of both as k increases. Tuning the hyperparameter 'k' using GridSearchCV. **Level 2** : – Using clustering for Image segmentation and Preprocessing. Kmeans++ **Experiment No. 11: Density Based Clustering**

	Level 1 Implement DBSCAN – clustering using the local density estimation. Perform hard and soft clustering for new instances. **Experiment No. 12: Outlier Detection** **Level 1** Outlier Detection using Isolation Forest and Local Outlier Factor
	Targeted Application & Tools that can be used : 1. Execution of the ML algorithms will be done using the Google's cloud service namely "Colab", available at https://colab.research.google.com/ or Jupyter Notebook. 2. The data sets will be from the bench marking repositories such as UCI machine learning repository available at : https://archive.ics.uci.edu/ml/index.php 3. Laboratory tasks will be implemented using the libraries available in Python such as Scikit learn, matplotlib, seaborn, perceptron and the deep learning framework namely Keras.
	Project work/Assignment: Mention the Type of Project /Assignment proposed for this course
	Students can be assigned a mini project to develop a machine learning application for real-life problems in various domains such as health care, business intelligence, environmental modeling, etc.
	Text Book There are a number of useful textbooks for the course, but each cover only a part of the course syllabus. Following is an indicative list of textbooks. 1. Aurélien Géron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow", Oreilly, Second Edition, 2019. 2. Andreas C Muller, Sarah Guido, "Introduction to Machine Learning with Python :A Guide for Data Scientists", Oreilly, First Edition, 2018 3. Giuseppe Bonaccorso, "Machine Learning Algorithms: A reference guide to popular algorithms from data science and machine learning", Packt Publishing, 2017.
	References In references apart from the books and web links, mention a few standards &Hand books relevant to the Laboratory tasks used by the professionals. 1. Tan P. N., Steinbach M & Kumar V. "***Introduction to Data Mining"***, Pearson Education, 2016. 2. **https://towardsdatascience.com/machine-learning/home** 3. MITopencourseware:https://ocw.mit.edu/courses/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/resources/lecture-11-introduction-to-machine-learning/ 4. https://onlinecourses.nptel.ac.in/noc21_cs85/preview

Course Code: CSE3501	**Course Title: Advanced Deep Learning Techniques** **Type of Course: PEC – Theory & Lab Integrated**	**L- T-P- C**	**3**	**0**	**0**	**3**
Version No.						
Course Pre-requisites	CSE2267					
Anti-requisites						
Course Description	This course introduces students to the concepts of deep neural networks and state of the art approaches to develop deep learning models. In this course students will be given an exposure to the details of neural networks as well as deep learning architectures and to develop end-to-end models for such tasks. It will help to design and develop an application-specific deep learning models and also provide the practical knowledge handling and analyzing end user realistic applications. Topics include Fundamental concepts of deep neural networks, Convolutional Neural Networks, Recurrent Network structures, Deep Unsupervised Learning, Generative Adversarial Networks and applications in various problem domains.					
Course Objective	This course is designed to improve the learners EMPLOYABILITY SKILLS by using EXPERIENTIAL LEARNING techniques.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Learn the Fundamental Principles of Deep Learning. (Remember). 2. Identify the Deep Learning Algorithms for learning tasks in various related domains (Apply). 3. To understand and apply deep generative models. (Understand). 4. Apply deep learning architectures to image and audio data. (Apply)					
Course Content:						

Module 1	**Introduction to Deep Learning and Neural Networks**	Assignment		**13[7L+6P] Sessions**
Topics: Fundamentals of Deep Learning, Perceptron, Multilayer Perceptron, Optimizing Perceptions using Activation Functions, Loss Functions, Gradient Descent. Feedforward Neural Network, Training Neural Network with Back-propagation, Hyper parameters, Regularization, Dropouts, Batch Normalization, Practical Issues in Neural Network Training -The Problem of Overfitting, The Vanishing and Exploding Gradient Problems				
Module 2	**Common Deep Learning Architectures:**	Assignment		**18[8L+10P] Sessions**
Topics: Convolutional Neural Network, Transfer learning Techniques, Variants of CNN: DenseNet, ResNet Sequence Modelling: Recurrent Neural Network and its variants - Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)				
Module 3	**Deep Generative Models**	Assignment		**16[8L+8P] Sessions**
Topics: Generative Adversarial Networks, Kohonen Networks, Autoencoders, Boltzmann Machine, Restricted Boltzmann Machine, Deep Belief Network				
Module-4	**Advanced Deep Learning Architectures**	Assignment		**13[7L+6P] Sessions**
Topics: Hopfield Network, Probabilistic Neural Network, Deep Reinforcement Learning - The Basic Framework of Reinforcement Learning Deep Learning applications: Image segmentation, Object detection,Speech Recognition, Video Analytics				

Project work/Assignment:

1. **Assignment 1 on (Module 1 and Module 2)**
2. **Assignment 2 on (Module 3 and Module 4)**

List of Laboratory Tasks:

Lab 1: Working with Deep Learning Frameworks
Objective: Explore various Deep Learning Frameworks
Tasks: Identify deep learning frameworks (Keras, Tensorflow, Matplotlib, etc)
Activity: Practice with various methods available in DL Frameworks to develop a Model.

Lab 2: Build a Basic Artificial Neural Network
Objective: Create a ANN with DL frameworks.
Task: Identify suitable ANN Layers using Keras and Tensorflow.
Activity: Design a basic Artificial Neural Networks using Keras with TensorFlow (pima-indians-diabetes)

Lab 3 and Lab 4: Build a MultiLayer Perceptron
Objective: Create a MLP for classification task.
Task: Identify suitable model for house price prediction.
Activity: Design a MLP for implementing classification and fine-tuning using House price.csv

Lab 5: Build a Convolutional Neural Network
Objective: Create a CNN model.
Task: Build CNN architecture for Dog-Cat classification problem.
Activity: Implement a Convolution Neural Network (CNN) for dog/cat classification problem using keras

Lab 6 and Lab 7: Build a Time-Series Model
Objective: Create a RNN and LSTM Model
Task: Build RNN/LSTM Model for predicting time series data.
Activity Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU notes

Lab 8: Build a Gated Recurrent Unit architecture.
Objective: Create a Time Series Model.
Task: Build GRU Architecture for predicting time series data.
Activity: Implement a GRU architecture for language translations.

Lab 9 and Lab 10: Build a Transfer Learning Model.
Objective: Create a Seq2Seq Model
Task: Create Hugging-face API using Transfer learning model.
Activity: Implement Transfer Learning models for classification problems Exploring Hugging-face API

Lab 11: Build an Auto-Encoder model
Objective: Create an Unsupervised Deep Learning Model.
Task: Create AutoEncoder network Output Translations.
Activity: implement an Encoder-Decoder Recurrent neural network model for Neural Machine Translation.

Lab 12: Build Generative Adversarial Networks.
Objective: Create an Unsupervised Deep Learning Model.
Task: Design GAN Architecture for Image generations.
Activity: Design a Age Prediction model by Applying Generative Adversarial

REFERENCE MATERIALS:

TEXTBOOKS

1. François Chollet, “Deep Learning with Python”, 2nd Edition, Manning Publications, 2022
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2017.

REFERENCES
1. Amlan Chakrabarti Amit Kumar Das, Saptarsi Goswami, Pabitra Mitra , “Deep Learning”, Pearson Publication, 2021.
2. David Foster, “Generative Deep Learning” O’Reilly Publishers, 2020.
3. John D Kellehar, “Deep Learning”, MIT Press, 2020.

JOURNALS/MAGAZINES

1. IEEE Transactions on Neural Networks and Learning Systems
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385
2. IEEE Transactions on Pattern Analysis and Machine Intelligence
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34http://ijaerd.com/papers/special_papers/IT032.pdf
3. International Journal of Intelligent Systems https://onlinelibrary.wiley.com/journal/1098111x

SWAYAM/NPTEL/MOOCs:

4. Swayam Nptel – Deep Learning – IIT Ropar https://onlinecourses.nptel.ac.in/noc21_cs35/preview
5. Coursera – Neural Networks and Deep Learning Andrew Ng
6. Coursera - Neural Networks for Machine Learning by Geoffrey Hinton in Coursera

<table>
<tr><td>Course Code:
CSE3502</td><td colspan="3">Course Title: Computational Optimization for Intelligent Systems
Type of Course: PEC – Theory & Lab Integrated</td><td>L-T- P- C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">CSE2267</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course introduces a range of machine learning models and optimization tools that are used to apply these models in practice. The course will delve into the underlying principles of optimization tools, often used as a black box, providing an understanding of the trade-offs between numerical accuracy and theoretical and empirical complexity. For students with some optimization background, this course will introduce a variety of applications arising in machine learning and statistics, as well as novel optimization methods targeting these applications.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Optimization Techniques for Machine Learning and attain Skill Development through Participative Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="8">On successful completion of this course the students shall be able to:
1. Demonstrate simple examples to illustrate how Machine Learning is applied in real-world scenarios. [Understand].
2. Implement Machine Learning models (e.g., decision trees, linear regression, neural networks) using tools or programming languages. [Apply].
3. Determine the suitability of convex optimization in solving problems like portfolio optimization, machine learning, or network design. [Apply].
4. Solve convex optimization problems with real or simulated data, such as minimizing a cost function or optimizing resource allocation. [Apply].</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1:</td><td>Fundamentals of Machine learning</td><td>Quiz</td><td></td><td colspan="3">Knowledge based Quiz</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="9">Topics: Machine Learning Paradigm, Empirical Risk Minimization, Structural Risk Minimization, Learning Guarantees, Introduction of VC-Dimension, Dimensionality Reduction Techniques (e.g., PCA, t-SNE basics).</td></tr>
<tr><td>Module 2:</td><td>Machine learning models</td><td>Quiz</td><td></td><td colspan="3">Comprehension based Quiz</td><td colspan="2">12 Sessions</td></tr>
<tr><td colspan="9">Topics: Logistic Regression, Support Vector Machines (SVMs) - primal and dual forms, kernels, Sparse Regression (e.g., Lasso, Ridge), Low Dimensional Embedding, Low Rank Matrix Factorization, Sparse PCA, Multiple Kernel Learning, Loss Functions (e.g., Mean Squared Error, Absolute Error), Entropy, Cross-Entropy Loss.</td></tr>
<tr><td>Module 3</td><td>Convex optimization models</td><td>Assignment</td><td></td><td colspan="3">Batch-wise Assignments</td><td colspan="2">13 Sessions</td></tr>
<tr><td colspan="9">Topics: Linear Optimization (Linear Programming), Convex Quadratic Optimization, Second Order Cone Optimization (SOCP), Semi-definite Optimization (SDP) - basic concepts, Convex Composite Optimization.</td></tr>
<tr><td>Module 4:</td><td>Methods for convex optimization</td><td>Assignment and Presentation</td><td></td><td colspan="3">Batch-wise Assignment and Presentations</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="9">Topics: Gradient Descent and its variants (Batch, Mini-batch, Stochastic Gradient Descent), Newton Method, Interior Point Methods (basics), Active Set Methods (for quadratic programming), Proximal Methods, Accelerated Gradient Methods (e.g., Nesterov's accelerated gradient), Coordinate Descent, Cutting Plane Methods, Stochastic Gradient Descent (SGD) and its variants (e.g., Adam, RMSprop).</td></tr>
<tr><td colspan="9">Targeted Application & Tools that can be used:
Use of Google Colab (Python with libraries like NumPy, SciPy, scikit-learn, TensorFlow/PyTorch, CVXPY/Pyomo).</td></tr>
<tr><td colspan="9">Lab Tasks:</td></tr>
</table>

Tools: Google Colab, Python with libraries: NumPy, SciPy, scikit-learn, Matplotlib, Seaborn, Pandas, CVXPY (for convex optimization), TensorFlow/PyTorch (for neural networks).

Experiment 1: Introduction to Python for ML & Optimization

- **Low-Level:** Basic Python data structures, NumPy array operations, Matplotlib for basic plotting.
- **Higher-Level:** Implement a simple vectorized operation (e.g., dot product) without explicit loops. Analyze the performance difference.

Experiment 2: Linear Regression with Gradient Descent

- **Low-Level:** Implement simple linear regression from scratch using batch gradient descent. Plot the loss function over iterations.
- **Higher-Level:** Implement linear regression using stochastic gradient descent (SGD) and mini-batch gradient descent. Compare their convergence behavior and plot the results.

Experiment 3: Logistic Regression for Classification

- **Low-Level:** Implement logistic regression from scratch for a binary classification problem (e.g., Iris dataset - two classes). Use a sigmoid activation function.
- **Higher-Level:** Extend logistic regression to multiclass classification (e.g., One-vs-Rest or Softmax regression). Evaluate performance using appropriate metrics.

Experiment 4: Support Vector Machines (SVM) Fundamentals

- **Low-Level:** Use scikit-learn's SVC to train a linear SVM on a linearly separable dataset. Visualize the decision boundary.
- **Higher-Level:** Experiment with different kernels (polynomial, RBF) for non-linearly separable data. Tune hyperparameters (C, gamma) and analyze their impact on the decision boundary and generalization.

Experiment 5: Dimensionality Reduction using PCA

- **Low-Level:** Apply PCA on a dataset (e.g., MNIST digits) to reduce dimensionality. Visualize the principal components.
- **Higher-Level:** Reconstruct data from reduced dimensions and calculate reconstruction error. Compare the performance of a classifier trained on original vs. PCA-reduced data.

Experiment 6: Introduction to Convex Optimization with CVXPY

- **Low-Level:** Solve a simple linear programming problem (e.g., resource allocation) using CVXPY.
- **Higher-Level:** Solve a basic convex quadratic programming problem (e.g., least squares with L2 regularization) using CVXPY.

Experiment 7: Linear Programming for Portfolio Optimization

- **Low-Level:** Formulate and solve a simple portfolio optimization problem (e.g., maximizing return for a given risk) as a linear program using CVXPY.
- **Higher-Level:** Introduce additional constraints (e.g., maximum allocation per asset, minimum number of assets) and solve the more complex problem.

Experiment 8: Understanding Loss Functions and Regularization

- **Low-Level:** Implement and visualize different loss functions (MSE, MAE, Cross-Entropy) for regression and classification.
- **Higher-Level:** Apply L1 (Lasso) and L2 (Ridge) regularization to linear regression. Analyze the impact of regularization strength on model coefficients and generalization error.

Experiment 9: Gradient Descent Variants for Complex Functions

- **Low-Level:** Implement and compare the convergence of Batch Gradient Descent, Mini-batch Gradient Descent, and Stochastic Gradient Descent on a non-convex function (e.g., Rosenbrock function).
- **Higher-Level:** Experiment with adaptive learning rate optimizers like Adam or RMSprop

from scratch (or using a library) and compare their performance.

Experiment 10: Introduction to Neural Networks and Backpropagation

- **Low-Level:** Build a simple feedforward neural network for binary classification using TensorFlow/PyTorch. Understand the basic concept of backpropagation.
- **Higher-Level:** Implement a multi-layer perceptron (MLP) for a multiclass classification problem. Experiment with different activation functions and analyze their impact.

Experiment 11: Sparse Regression Techniques

- **Low-Level:** Apply Lasso regression using scikit-learn on a dataset with many features. Observe how Lasso performs feature selection.
- **Higher-Level:** Compare Lasso with Ridge regression on a high-dimensional dataset. Discuss when to use each technique.

Experiment 12: Low-Rank Matrix Factorization for Recommender Systems

- **Low-Level:** Implement a basic matrix factorization algorithm (e.g., Singular Value Decomposition - SVD) for a small rating matrix.
- **Higher-Level:** Apply a collaborative filtering approach using matrix factorization for a simple recommender system. Evaluate prediction accuracy.

Experiment 13: Convex Composite Optimization - Proximal Methods

- **Low-Level:** Understand the concept of proximal operators for simple functions (e.g., L1 norm).
- **Higher-Level:** Implement a basic Proximal Gradient Method (e.g., FISTA) for solving a regularized linear regression problem (e.g., Lasso).

Experiment 14: Second Order Cone Programming (SOCP) Applications

- **Low-Level:** Formulate and solve a simple SOCP problem (e.g., minimum enclosing ball problem) using CVXPY.
- **Higher-Level:** Explore an application of SOCP in machine learning, such as robust linear regression or robust PCA (conceptual understanding and small-scale implementation if feasible).

Experiment 15: Case Study: Optimization in Real-World ML Scenario

- **Low-Level:** Choose a simple real-world dataset (e.g., house price prediction, sentiment analysis) and apply a suitable machine learning model. Optimize its performance using one of the learned optimization methods.
- **Higher-Level:** Analyze a more complex dataset and apply multiple optimization techniques. Compare their effectiveness, training time, and generalization performance. Document the findings and insights.

Generate another set of 10 thought provoking / case study / industry questions for practice

Here are 10 thought-provoking / case study / industry questions for practice, building upon the "Computational Optimization for Intelligent Systems" syllabus:

1. **Case Study: Supply Chain Optimization with Uncertainty:** An e-commerce company faces fluctuating demand and unreliable supplier lead times. They need to optimize their inventory levels and transportation routes to minimize costs and maximize customer satisfaction.
   - How would you formulate this problem using a combination of machine learning (to predict demand/lead times) and convex optimization techniques?
   - Which specific optimization models (e.g., linear programming, robust optimization, stochastic programming) would be most suitable, and why?
   - Discuss the challenges of integrating predictions from ML models directly into optimization frameworks, and propose strategies to handle forecast uncertainty.
2. **Ethical Implications of Algorithmic Bias in Optimization:** A major bank uses a machine learning model, whose training involves extensive optimization, to decide loan approvals.

The model, while appearing accurate overall, shows a statistically significant bias against certain demographic groups.

- How could the optimization process (e.g., choice of loss function, regularization) potentially contribute to or exacerbate this bias?
- Propose methods, rooted in computational optimization, to mitigate algorithmic bias while maintaining acceptable model performance. Consider techniques like fair optimization or constrained optimization.

3. **Industry Application: Optimizing Energy Consumption in Data Centers:** A large cloud service provider wants to minimize the energy consumption of its data centers while ensuring service level agreements (SLAs) are met. This involves optimizing server utilization, cooling systems, and power distribution.
   - Describe how you would model this problem using a combination of machine learning (to predict workload/temperature) and a suitable optimization framework.
   - Which *types* of convex optimization problems (e.g., linear, quadratic, SOCP) might arise in this scenario, and for what specific sub-problems?
   - Discuss the trade-offs between energy efficiency and computational performance, and how optimization can help balance these.
4. **Hardware Acceleration for Optimization Algorithms:** Modern machine learning relies heavily on parallel processing and specialized hardware (GPUs, TPUs) for training large models.
   - How do the design principles of optimization algorithms (e.g., gradient descent, stochastic gradient methods) lend themselves to parallelization?
   - Discuss how the "Methods for Convex Optimization" (Module 4) can be adapted or re-designed to take full advantage of parallel computing architectures. What are the limitations?
5. **Explainable AI (XAI) and Optimization:** As ML models become more complex, the demand for explainability increases. Can optimization itself contribute to making models more interpretable?
   - Discuss how concepts from convex optimization, such as sparsity (L1 regularization), can directly contribute to model interpretability.
   - Can optimization techniques be used *after* a model is trained to extract "explanations" or identify important features? If so, provide an example.
6. **Beyond Convexity: When Standard Methods Fail:** While the course focuses on convex optimization, many real-world ML problems are non-convex (e.g., training deep neural networks).
   - What are the fundamental challenges and pitfalls of applying "Methods for Convex Optimization" (Module 4) directly to highly non-convex problems?
   - Briefly outline techniques or heuristics used to tackle non-convex optimization in machine learning (even if not explicitly covered in this course). How do they relate to the concepts learned?
7. **Real-time Optimization in Autonomous Systems:** An autonomous vehicle needs to make real-time decisions (e.g., path planning, obstacle avoidance) based on sensor data and predictive models. These decisions often involve complex optimization problems.
   - Discuss the unique computational and time constraints when applying optimization techniques in real-time autonomous systems.
   - How would the choice of optimization method (from Module 4) be influenced by these constraints? Which methods would be favored, and which would be less suitable?
8. **Case Study: Personalized Medicine and Treatment Optimization:** Imagine a system that

uses a patient's genetic data, medical history, and real-time health metrics to recommend personalized treatment plans. This involves optimizing drug dosages, lifestyle interventions, and therapy schedules.

- How can machine learning models be used to predict treatment efficacy, and how would these predictions be incorporated into an optimization framework?
- What challenges arise from a data perspective (e.g., small sample sizes for rare conditions, heterogeneity of patient responses) when applying optimization in personalized medicine?

9. **The Interplay of Data Quality and Optimization Performance:** "Garbage in, garbage out" is a common adage in ML.
    - How does the quality, cleanliness, and representativeness of the input data affect the performance and convergence of the optimization algorithms used to train ML models?
    - Discuss how data preprocessing techniques (which themselves might involve optimization, e.g., for feature scaling or imputation) can indirectly impact the success of the optimization phase.
10. **Future Trends: Quantum Computing and Optimization:** While still nascent, quantum computing holds promise for solving certain optimization problems that are intractable for classical computers.
    - Briefly explain how a quantum approach might differ from classical optimization methods for a problem like the Traveling Salesperson Problem (an NP-hard problem).
    - Speculate on how advancements in quantum optimization could potentially revolutionize specific areas of machine learning in the future, even if practical applications are still distant.

Project work/Assignment:
Survey on Methods for convex optimization in Machine Learning.
Survey on Machine learning models where optimization plays a crucial role.

Text Book

T1. Charu C. Aggarwal, “*Linear Algebra and Optimization for Machine Learning*”, Springer, 2020.
T2. Sra Suvrit, Nowozin Sebastian, and Wright Stephen J, “*Optimization for Machine Learning*”, The MIT Press,2012.

References

R1.Guanghui Lan, “*First-order and Stochastic Optimization Methods for Machine Learning*”, Springer Cham, 2020.

Web References

W1. https://sm-nitk.vlabs.ac.in/
W2. https://nptel.ac.in/courses/

Topics relevant to SKILL DEVELOPMENT: Concepts of Convex optimization models and Methods for convex optimization f**or Skill Development through Problem Solving methodologies. This is attained through assessment component mentioned in course handout.**

Course Code: CSE3503	Course Title: Reinforcement Learning for AI Systems Type of Course: PEC – Theory & Lab Integrated			L-T-P-C	2	0	2	3
Version No.	1.0							
Course Pre-requisites	CSE2267							
Anti-requisites	NIL							
Course Description	This course provides a comprehensive introduction to reinforcement learning (RL), a highly active research sub-field of machine learning. RL is fundamentally concerned with developing intelligent agents that learn optimal policies for decision-making in stochastic environments through interaction and experience. We will explore how these agents learn to predict future states and take actions that maximize cumulative rewards over time. Applications of RL are vast and impactful, ranging from classical control problems like power plant optimization and robotics to complex domains such as game playing (e.g., AlphaGo), inventory control, autonomous systems, and even modeling biological learning processes. The course will cover both theoretical underpinnings and practical applications, primarily following the classic textbook by Sutton & Barto (2nd Edition) and supplementing it with relevant research papers and contemporary materials.							
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Reinforcement Learning** and attain **Skill Development** through **Problem Solving** Methodologies.							
Course Out Comes	On successful completion of the course the students shall be able to: 1. **Understand** basic and advanced reinforcement learning concepts, algorithms, and their underlying mathematical foundations. [Knowledge] 2. **Identify** suitable real-world problems and formulate them as reinforcement learning tasks, selecting appropriate learning techniques. [Apply] 3. **Analyze** the strengths and limitations of various reinforcement learning techniques in different application domains. [Analyze] 4. **Design and Implement** computational experiments for reinforcement learning problems, evaluate the results, and draw meaningful conclusions. [Create & Evaluate]							
Course Content:								

Module 1	Introduction and Foundations	Assignment		Programming	No. of Classes:10
Topics: **Topics:** Course logistics and overview. Historical context and evolution of Reinforcement Learning research. Connections to other related fields (e.g., optimal control, operations research, game theory, psychology) and different branches of machine learning (supervised, unsupervised learning). **Probability Primer Brush-up:** Axioms of probability, concepts of random variables (discrete, continuous), Probability Mass Functions (PMF), Probability Density Functions (PDFs), Cumulative Distribution Functions (CDFs). Expectation, Variance. Concepts of joint and multiple random variables, joint, conditional, and marginal distributions. Covariance, correlation, and statistical independence. Introduction to stochastic processes. Understanding the probabilistic backbone of RL.					
Module 2	Markov Decision Processes (MDPs)	Assignment		Programming	No. of Classes:10
Topics: Introduction to RL terminology: Agent, Environment, States, Actions, Rewards, Policy, Value Function. Markov property and its significance. Markov Chains, Markov Reward Processes (MRP). **Bellman Equations for MRPs:** Introduction to Bellman equations for MRPs, derivation, and proof of existence of a unique solution. **Introduction to Markov Decision Processes (MDPs):** Formal definition of MDP, State-Value function V(s), Action-Value function Q(s,a). Bellman					

Expectation Equations for V(s) and Q(s,a). Optimality of Value Functions and Policies. Bellman Optimality Equations, Formalizing decision-making problems.					
Module 3	Prediction and Control by Dynamic Programming & Monte Carlo Methods	Assignment		Programming	**No. of Classes:10**
Topics: Dynamic Programming (DP) for MDPs: Overview of DP for planning in MDPs. Definition and formulation of the planning problem. Principle of Optimality. Iterative Policy Evaluation. Policy Iteration algorithm. Value Iteration algorithm. **Theoretical Foundations of DP:** Banach Fixed Point Theorem. Proof of contraction mapping property of Bellman Expectation and Optimality Operators. Proof of convergence of Policy Evaluation and Value Iteration algorithms. DP extensions and efficiency considerations. **Monte Carlo (MC) Methods for Model-Free Prediction and Control:** Overview of Monte Carlo methods for model-free RL. First-Visit Monte Carlo vs. Every-Visit Monte Carlo. Monte Carlo Control (e.g., On-policy Monte Carlo control for estimating optimal policies). On-policy vs. Off-policy learning. Introduction to Importance Sampling for off-policy MC, Implementing fundamental planning and model-free prediction algorithms.					
Module 4	Temporal-Difference (TD) Methods and Policy Gradients	Assignment		Programming	**No. of Classes:10**
Topics: Incremental Monte Carlo Methods for Model-Free Prediction: Introduction to Temporal Difference (TD) Learning. TD(0) for prediction. TD(1) and TD(λ) for eligibility traces. K-step estimators. Unified view of DP, MC, and TD evaluation methods. **TD Control Methods:** SARSA (State-Action-Reward-State-Action) algorithm for on-policy control. Q-Learning algorithm for off-policy control. Variants and improvements (e.g., Double Q-Learning). **Policy Gradient Methods:** Getting started with policy gradient methods. The Log-derivative trick. Naive REINFORCE algorithm. Bias and Variance in Reinforcement Learning algorithms. Techniques for reducing variance in policy gradient estimates: baselines, advantage function. Introduction to Actor-Critic Methods (A2C/A3C conceptual overview), Developing advanced model-free control strategies and understanding foundational deep RL concepts.					
Targeted Application & Tools that can be used: • Reinforcement Learning as a framework for computational neuroscience to model decision-making processes. • Applications of RL in various industries: Resource Management, Traffic Light Control, Robotics, Web System Configuration, Finance, Healthcare, Gaming, Autonomous Systems, etc. • **Tools:** Python (Jupyter Notebook, Google Colaboratory, Spyder), Libraries: NumPy, OpenAI Gym, Stable Baselines3, Ray RLib, PyTorch / TensorFlow (for Deep RL).					
Lab Tasks: **Tools:** Google Colab, Python, OpenAI Gym, NumPy, Matplotlib, PyTorch / TensorFlow (for Deep RL components), Stable Baselines3 (for higher-level experiments/benchmarking).					
Experiment 1: Setting up the RL Environment & Basic Python for RL • **Low-Level:** Install necessary libraries (Gym, NumPy, Matplotlib). Explore basic Gym environments (e.g., 'CartPole-v1', 'FrozenLake-v1'). Understand state and action spaces. • **Higher-Level:** Implement a simple random agent for 'FrozenLake-v1' or 'CartPole-v1'. Calculate and report the average reward over multiple episodes. **Experiment 2: Markov Chains and Markov Reward Processes** • **Low-Level:** Define a small Markov Chain (e.g., 3 states) and calculate its stationary distribution manually and programmatically. • **Higher-Level:** Extend to a Markov Reward Process. Calculate the value function for a given discount factor using matrix inversion or iterative methods. **Experiment 3: Policy Evaluation using Iterative Methods (DP)**					

- **Low-Level:** Implement iterative policy evaluation for a small gridworld environment (defined manually) with a given policy.
- **Higher-Level:** Visualize the value function after convergence. Compare the convergence speed with different thresholds.

Experiment 4: Policy Iteration Algorithm

- **Low-Level:** Implement the policy iteration algorithm for a small gridworld. Trace the changes in policy and value function across iterations.
- **Higher-Level:** Experiment with different initial policies and observe their impact on convergence.

Experiment 5: Value Iteration Algorithm

- **Low-Level:** Implement the value iteration algorithm for a small gridworld. Observe the convergence of the optimal value function.
- **Higher-Level:** Extract the optimal policy from the converged value function. Compare the optimal policy with that found by Policy Iteration.

Experiment 6: Monte Carlo Prediction (First-Visit & Every-Visit)

- **Low-Level:** Implement First-Visit Monte Carlo prediction to estimate the value function for a given policy in 'FrozenLake-v1'.
- **Higher-Level:** Implement Every-Visit Monte Carlo prediction. Compare the estimated values and convergence properties of both methods.

Experiment 7: Monte Carlo Control (On-Policy: MC Exploring Starts)

- **Low-Level:** Implement Monte Carlo control with Exploring Starts to find an optimal policy for a small gridworld or 'Blackjack-v1' environment.
- **Higher-Level:** Analyze the learning curve (average reward over episodes). Discuss the practical limitations of Exploring Starts.

Experiment 8: TD(0) Prediction

- **Low-Level:** Implement TD(0) to estimate the value function for a given policy in 'FrozenLake-v1'.
- **Higher-Level:** Compare the convergence of TD(0) with Monte Carlo prediction. Discuss the advantages of TD learning.

Experiment 9: SARSA for On-Policy Control

- **Low-Level:** Implement the SARSA algorithm to learn an optimal policy for 'FrozenLake-v1'.
- **Higher-Level:** Experiment with different epsilon-greedy exploration strategies and learning rates. Analyze the trade-off between exploration and exploitation.

Experiment 10: Q-Learning for Off-Policy Control

- **Low-Level:** Implement the Q-Learning algorithm to learn an optimal policy for 'FrozenLake-v1'.
- **Higher-Level:** Compare the learned policy and convergence behavior of Q-Learning with SARSA. Discuss the implications of on-policy vs. off-policy learning.

Experiment 11: Introduction to Function Approximation (Linear/Neural Network)

- **Low-Level:** Implement linear function approximation for a simple value function (e.g., using tile coding or polynomial features) in an environment with a large state space.
- **Higher-Level:** Introduce a small neural network (e.g., 1-2 hidden layers) to approximate the Q-function. Begin to understand Deep Q-Networks conceptually.

Experiment 12: Deep Q-Networks (DQN) - Conceptual and Basic Implementation

- **Low-Level:** Understand the architecture of a DQN. Use a pre-built DQN agent from `stable-baselines3` or a simple custom DQN for 'CartPole-v1'.
- **Higher-Level:** Experiment with replay buffers and target networks (conceptually and by observing their use in the library). Analyze the impact of these techniques on stability.

Experiment 13: Policy Gradient Methods (REINFORCE)

- **Low-Level:** Implement the REINFORCE algorithm for 'CartPole-v1'. Focus on the log-derivative trick and calculating gradients.
- **Higher-Level:** Introduce a baseline to reduce variance in the REINFORCE algorithm. Compare the learning stability with and without a baseline.

Experiment 14: Actor-Critic Methods (Conceptual & Simple Implementation)

- **Low-Level:** Understand the basic idea behind Actor-Critic methods (separate policy and value networks).
- **Higher-Level:** Implement a very basic Actor-Critic agent for 'CartPole-v1' or a similar environment. (Focus on understanding the interaction between actor and critic updates).

Experiment 15: Real-World Environment Simulation and Agent Evaluation

- **Low-Level:** Choose a simple Gym-like environment that simulates a real-world problem (e.g., 'Taxi-v3', 'CliffWalking-v0'). Apply one or two learned RL algorithms (e.g., Q-Learning, SARSA) and evaluate their performance.
- **Higher-Level:** Analyze the strengths and weaknesses of the chosen algorithms for the specific problem. Discuss potential improvements or more advanced algorithms that could be applied. This could involve exploring a more complex environment or adding partial observability.

10 Real-World Industry Case Studies for Practice

These case studies are designed to encourage critical thinking, problem formulation, and selection of appropriate RL techniques for practical scenarios. Students should be encouraged to research, discuss, and propose solutions, possibly even prototyping small parts if time permits.

1. **Autonomous Warehouse Logistics:**
    - **Scenario:** A large e-commerce warehouse uses autonomous robots to move inventory. How can RL be used to optimize robot path planning, task scheduling, and collision avoidance to maximize throughput and minimize energy consumption?
    - **Questions:** What would be the states, actions, and rewards for an individual robot? How would you handle coordination in a multi-robot system? Which RL algorithms (e.g., Q-Learning, PPO, multi-agent RL) would be suitable, and why?
2. **Financial Trading Strategy Optimization:**
    - **Scenario:** A hedge fund wants to develop an automated trading agent that learns optimal buy/sell/hold decisions for a specific stock or portfolio based on market data.
    - **Questions:** How can market data (price, volume, news sentiment) be formulated as states? What are the challenges in defining rewards (e.g., immediate profit vs. long-term portfolio value)? Discuss the ethical and risk management implications of using RL in high-stakes financial environments. Which RL algorithms are more suited for continuous action spaces or complex state representations?
3. **Smart Grid Energy Management:**
    - **Scenario:** Optimize energy distribution in a smart grid, balancing renewable energy sources, consumer demand, and battery storage to minimize cost and ensure grid stability.
    - **Questions:** How would you define the state of the grid (e.g., energy levels, demand, forecasts)? What actions can the RL agent take? What are the long-term rewards? Discuss the complexities of handling continuous variables and the need for robust control.
4. **Personalized Healthcare and Treatment Planning:**
    - **Scenario:** Develop an AI system that learns to recommend personalized treatment plans for chronic diseases (e.g., diabetes, hypertension) based on patient data, response to treatment, and health outcomes.

  - **Questions:** How can medical states be represented? What are the actions (e.g., medication dosage, lifestyle advice)? What are the ethical challenges of using RL for human health decisions? Discuss the importance of incorporating medical expert knowledge into the reward function or policy constraints.

5. **Game AI for Non-Player Characters (NPCs):**
   - **Scenario:** Design intelligent NPCs in a complex video game (e.g., a strategy game or open-world RPG) that exhibit adaptive, believable behavior without being explicitly scripted.
   - **Questions:** How can the game state and NPC actions be mapped to an MDP? What kind of reward function encourages desired behaviors (e.g., exploration, combat strategy, social interaction)? Discuss the use of hierarchical RL for complex tasks.
6. **Optimizing A/B Testing in Digital Marketing:**
   - **Scenario:** A marketing team wants to continuously optimize website layouts, ad creatives, or email subject lines by dynamically adjusting content presented to users based on their real-time engagement, moving beyond traditional static A/B testing.
   - **Questions:** How can this be framed as a contextual bandit problem or a full RL problem? What are the states (user demographics, past interactions), actions (content variations), and rewards (click-through rate, conversion)? Discuss the importance of exploration vs. exploitation in this context.
7. **Traffic Management for Ride-Sharing Services:**
   - **Scenario:** A ride-sharing company wants to optimize the dispatching of drivers and dynamic pricing to reduce passenger wait times, minimize driver idle time, and maximize company revenue across a city.
   - **Questions:** What would be the spatial and temporal states? What actions can the system take (e.g., driver re-positioning, surge pricing)? How would you define the reward function to balance multiple objectives? Consider multi-agent RL approaches for large fleets.
8. **Industrial Process Control and Optimization:**
   - **Scenario:** Optimize parameters (e.g., temperature, pressure, flow rate) in a complex manufacturing process (e.g., chemical plant, steel production) to maximize yield, minimize waste, and ensure product quality.
   - **Questions:** How can sensor readings form the state space? What are the adjustable parameters as actions? What are the safety constraints and how can they be incorporated? Discuss the challenge of exploring potentially dangerous states in a real-world industrial setting.
9. **Cybersecurity: Adaptive Threat Detection and Response:**
   - **Scenario:** Develop an AI agent that can learn to identify novel cyber threats and take proactive defense actions (e.g., isolate a compromised system, block an IP address) in a dynamic network environment.
   - **Questions:** How would you represent the network state and ongoing attacks? What actions are available to the defense agent? What are the rewards for successful defense and penalties for false positives or delayed responses? Discuss the challenge of dealing with an adversarial environment.
10. **Resource Allocation in Cloud Computing:**
    - **Scenario:** A cloud provider needs to dynamically allocate virtual machine (VM) resources (CPU, RAM, network bandwidth) to multiple user applications to meet performance guarantees while minimizing energy consumption and maximizing server utilization.
    - **Questions:** How would the state reflect the current resource usage and application

demands? What actions correspond to resource adjustments? What are the conflicting objectives that need to be balanced in the reward function? Consider how multi-agent RL might be applied if each application or server is an agent.

Project work/Assignment:

This part is written for general readers. At the same time, it will be of greater value for readers with some knowledge about RL.

Resources management in computer clusters

Designing algorithms to allocate limited resources to different tasks is challenging and requires human-generated heuristics. The paper "Resource Management with Deep Reinforcement Learning" [2] showed how to use RL to automatically learn to allocate and schedule computer resources to waiting jobs, with the objective to minimize the average job slowdown.

State space was formulated as the current resources allocation and the resources profile of jobs. For action space, they used a trick to allow the agent to choose more than one action at each time step. Reward was the sum of (-1/duration of the job) over all the jobs in the system. Then they combined REINFORCE algorithm and baseline value to calculate the policy gradients and find the best policy parameters that give the probability distribution of actions to minimize the objective.

Traffic Light Control

Researchers tried to design a traffic light controller to solve the congestion problem. Tested only on simulated environment though, their methods showed superior results than traditional methods and shed a light on the potential uses of multi-agent RL in designing traffic system.

Five agents were put in the five-intersection traffic network, with a RL agent at the central intersection to control traffic signalling. The state was defined as eight-dimensional vector with each element representing the relative traffic flow of each lane. Eight choices were available to the agent, each representing a phase combination, and the reward function was defined as reduction in delay compared with previous time step. The authors used DQN to learn the Q value of the {state, action} pairs.

Robotics

There are tremendous works on applying RL in Robotics. Readers are referred to for a survey of RL in Robotics. In particular, trained a robot to learn policies to map raw video images to robot's actions. The RGB images were fed to a CNN and outputs were the motor torques. The RL component was the guided policy search to generate training data that came from its own state distribution.

Web System Configuration

There are more than 100 configurable parameters in a web system and the process of tuning the parameters requires a skilled operator and numerous trail-and-error tests. The paper "A Reinforcement Learning Approach to Online Web System Auto-configuration" showed the first attempt in the domain on how to do autonomic reconfiguration of parameters in multi-tier web systems in VM-based dynamic environments.

The reconfiguration process can be formulated as a finite MDP. The state space was the system configuration, action space was {increase, decrease, keep} for each parameter, and reward was defined as the difference between the given targeted response time and measured response time. The authors used the model-free Q-learning algorithm to do the task.

Text Book

1. "Reinforcement Learning: An Introduction", Richard S. Sutton and Andrew G. Barto, 2nd Edition
2. "Probability, Statistics, and Random Processes for Electrical Engineering", 3rd Edition, Alberto Leon-Garcia
3. "Machine Learning: A Probabilistic Perspective", Kevin P. Murphy

References

1. Richard S. Sutton and Andrew G. Barto, "Reinforcement learning: An introduction", Second Edition, MIT Press, 2019.
2. Li, Yuxi. "Deep reinforcement learning." arXiv preprint arXiv:1810.06339 (2018).
3. Wiering, Marco, and Martijn Van Otterlo. "Reinforcement learning." Adaptation, learning, and optimization 12 (2012):

E-Resources

1. NPTEL course – https://onlinecourses.nptel.ac.in/noc19_cs55/preview
2. NPTEL course archive – https://archive.nptel.ac.in/courses/106/106/106106143/
3. Additional NPTEL content – https://www.digimat.in/nptel/courses/video/106106143/L35.html
4. OpenAI Spinning Up in Deep RL: https://spinningup.openai.com/en/latest/

Topics relevant to "SKILL DEVELOPMENT": Real-time Data Analysis using Reinforcement Learning for Skill Development through Problem Solving techniques. This is attained through assessment components mentioned in the course handout, including laboratory experiments and the final project.

<table>
<tr><td>Course Code: CSE3504</td><td colspan="3">Course Title: Computational Linguistics and Natural Language Processing
Type of Course: PEC – Theory & Lab Integrated</td><td>L- T-P- C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">CSE2267</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course provides a foundational and practical introduction to Natural Language Processing (NLP), the fascinating field at the intersection of computer science, artificial intelligence, and linguistics. NLP focuses on enabling computers to understand, interpret, and generate human language. Students will learn the fundamental theories and cutting-edge techniques required to extract meaningful information from unstructured text and build intelligent language-aware systems. The course emphasizes a hands-on approach, integrating theoretical concepts with practical implementation using popular NLP libraries and tools.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of this course is to familiarize learners with the core concepts and state-of-the-art techniques of Natural Language Processing, and to foster skill development through participative learning techniques, including practical implementations and problem-solving.</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of the course the students shall be able to:
1. Understand the fundamental concepts, theories, and challenges in Natural Language Processing. [Knowledge]
2. Apply appropriate preprocessing, representation, and modeling techniques to solve various NLP tasks using programming languages and libraries. [Application]
3. Utilize word embeddings and deep learning architectures to build effective NLP applications. [Application]
4. Analyze and Evaluate the performance of different NLP models for tasks such as machine translation, sentiment analysis, and question answering. [Analysis & Evaluation]</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction to NLP and Linguistic Fundamentals</td><td colspan="2">Quizzes, Small Programming exercises.</td><td colspan="5">7 Sessions</td></tr>
<tr><td colspan="9">Topics:
Introduction to NLP: What is NLP? History and evolution of NLP. Relationship with AI, ML, and Linguistics. Text Analytics & NLP Tasks: Overview of various NLP tasks (e.g., text classification, sentiment analysis, machine translation, information extraction, summarization, question answering). Basic Text Processing: Corpus and dataset understanding. Tokenization (word, sentence), Normalization (case folding, stemming, lemmatization). Text Preprocessing Challenges: Handling noise, stop words, special characters. Introduction to Edit Distance: Levenshtein distance, applications in spelling correction. Brief Overview of Core NLP Concepts: Introduction to word embeddings (concept only), Part-of-Speech (PoS) tagging, Chunking, Parsing, Machine Translation (high-level).</td></tr>
<tr><td>Module 2</td><td>Word and Text Representations & Traditional ML for NLP</td><td>Quizzes</td><td>Assignments</td><td colspan="5">8 Sessions</td></tr>
</table>

Topics:
Traditional ML for Classification: Review of Logistic Regression and Naïve Bayes Classifiers. Their application in text classification. **Count-based Text Representations:** Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF). **Vector Semantics and Word Embeddings:** Distributed representations, Introduction to Word2Vec (Skip-gram, CBOW), GloVe., Properties of word embeddings (semantic and syntactic relationships). **Introduction to Neural Networks:** Perceptrons, Multi-Layer Perceptrons (MLPs), activation functions, backpropagation (brief review of concepts as students are exposed to ML). **Neural Language Models (NNLM):** Basic architecture for predicting next word. **Introduction to Deep Learning Architectures for Sequence Processing:** Recurrent Neural Networks (RNNs): Basic RNN architecture, limitations (vanishing/exploding gradients), Long Short-Term Memory (LSTM): Architecture, handling long-term dependencies, Convolutional Neural Networks (CNNs) for text: 1D convolutions, pooling for feature extraction.

Module 3	**Advanced Tagging and Syntactic Parsing**	Quizzes	Assignments	**12 Sessions**

Topics:
Part-of-Speech (PoS) Tagging: Rule-based and statistical PoS tagging, Hidden Markov Models (HMMs) for PoS tagging: Forward-backward algorithm, Viterbi algorithm, Using NLTK and SpaCy for PoS tagging, **Named Entity Recognition (NER):** Definition and importance of NER, Rule-based and statistical approaches (e.g., Conditional Random Fields - CRFs concept), Relationship between NER tagging and PoS tagging, Using NLTK and SpaCy for NER ,**Syntactic Parsing:** Introduction to Formal Grammars: Context-Free Grammars (CFGs), Constituency Parsing: Treebanks, CKY algorithm (brief), Dependency Parsing: Introduction to dependency relations ,Using SpaCy for parsing. **Sequence Labeling with Deep Learning:** Introduction to Bi-LSTM-CRF architectures for PoS and NER (conceptual).

Module 4	**NLP Applications and Beyond**	Quizzes		**9 Sessions**

Topics:
Lexical Resource Creation: Introduction to WordNet, FrameNet. **Sentiment Analysis:** Lexicon-based methods, Machine learning approaches (supervised classification), Deep learning approaches (e.g., using LSTMs, CNNs for sentiment), Aspect-based sentiment analysis (conceptual). **Machine Translation:** Rule-based and Statistical Machine Translation (SMT) (brief overview), Neural Machine Translation (NMT): Encoder-Decoder architecture, Attention Mechanism, Sequence-to-Sequence (Seq2Seq) models, **Word Sense Disambiguation (WSD):** Definition and challenges. Rule-based and supervised approaches, **Question Answering (QA) Systems:** Information Retrieval-based QA. Knowledge Graph-based QA. Reading Comprehension QA (e.g., SQuAD dataset, attention mechanisms in QA). **Introduction to Transformer Architecture:** Conceptual understanding of self-attention and its impact on modern NLP (BERT, GPT - high-level overview).

Targeted Application & Tools that can be used:

1. **Python Libraries:** NLTK, SpaCy, scikit-learn, Pandas, NumPy, Matplotlib, Seaborn, PyTorch / TensorFlow (for deep learning).
2. **Java:** Stanford CoreNLP (for specific demonstrations/comparisons, though Python will be primary).
3. **Development Environment:** Google Colab, Jupyter Notebook, local Python environments.

Tools: Google Colab, Python, NLTK, SpaCy, scikit-learn, Matplotlib, Seaborn, Pandas, PyTorch / TensorFlow.

Experiment 1: Basic Text Preprocessing with NLTK

- **Low-Level:** Tokenization (word/sentence), lowercasing, stop word removal, stemming

(Porter/Snowball).

- **Higher-Level:** Implement lemmatization using WordNet. Compare stemming and lemmatization on a small corpus and analyze their impact on word forms.

Experiment 2: Regular Expressions for Text Pattern Matching

- **Low-Level:** Use regular expressions to extract specific patterns (e.g., phone numbers, email addresses) from raw text.
- **Higher-Level:** Build a simple rule-based tokenizer or a regex-based pattern extractor for a slightly complex linguistic phenomenon (e.g., identifying dates or currency values).

Experiment 3: Feature Engineering - Bag-of-Words and TF-IDF

- **Low-Level:** Convert a collection of text documents into Bag-of-Words (BoW) and TF-IDF representations using `CountVectorizer` and `TfidfVectorizer` from `scikit-learn`.
- **Higher-Level:** Analyze the sparsity of the resulting matrices. Experiment with `ngram_range` and `max_features` parameters and observe their impact on feature space size.

Experiment 4: Text Classification with Naïve Bayes and Logistic Regression

- **Low-Level:** Implement a text classifier using Naïve Bayes on a simple text dataset (e.g., spam detection). Evaluate with accuracy.
- **Higher-Level:** Implement a Logistic Regression classifier on the same dataset. Compare performance using metrics like precision, recall, F1-score, and confusion matrix.

Experiment 5: Exploring Word Embeddings (Word2Vec/GloVe)

- **Low-Level:** Load pre-trained Word2Vec or GloVe embeddings using `gensim` or `spaCy`. Find word similarities and analogies (e.g., king - man + woman = ?).
- **Higher-Level:** Visualize a small set of word embeddings using t-SNE or PCA. Discuss semantic relationships observed.

Experiment 6: Implementing a Simple Neural Network for Text Classification

- **Low-Level:** Build a simple feedforward neural network (MLP) using PyTorch/TensorFlow for binary text classification (e.g., positive/negative sentiment) using BoW or TF-IDF features.
- **Higher-Level:** Experiment with different hidden layer sizes and activation functions. Analyze the impact of hyperparameter tuning.

Experiment 7: Part-of-Speech (PoS) Tagging with NLTK and SpaCy

- **Low-Level:** Use NLTK's default PoS tagger and SpaCy's default model to tag sentences. Understand the PoS tagset.
- **Higher-Level:** Evaluate the accuracy of the taggers on a small annotated corpus. Identify common tagging errors and discuss their linguistic reasons.

Experiment 8: Hidden Markov Model (HMM) for PoS Tagging

- **Low-Level:** Implement a basic HMM for PoS tagging. Train it on a small pre-tagged corpus (e.g., from NLTK) and perform tagging on new sentences.
- **Higher-Level:** Analyze the Viterbi algorithm's role in finding the most likely tag sequence. Discuss the limitations of HMMs for PoS tagging.

Experiment 9: Named Entity Recognition (NER)

- **Low-Level:** Use SpaCy's built-in NER model to identify named entities in various texts. Extract entity types and their spans.
- **Higher-Level:** Identify cases where the default NER model fails. Discuss strategies for custom NER (e.g., rule-based, training custom models).

Experiment 10: Syntactic Parsing (Constituency and Dependency)

- **Low-Level:** Use NLTK's `RegexpParser` for simple chunking (e.g., NP chunking). Use SpaCy to visualize dependency parse trees for sentences.
- **Higher-Level:** Analyze the differences between constituency and dependency parses. Discuss their applications in information extraction.

Experiment 11: Sentiment Analysis with Deep Learning (LSTM/CNN)

- **Low-Level:** Build a sentiment classification model using a simple LSTM or 1D CNN layer with pre-trained word embeddings on a movie review dataset.
- **Higher-Level:** Experiment with different network architectures (e.g., adding more layers, bidirectional LSTMs). Analyze the impact of embedding choice and network complexity on performance.

Experiment 12: Sequence-to-Sequence (Seq2Seq) Model for Simple Tasks

- **Low-Level:** Implement a basic Encoder-Decoder Seq2Seq model (without attention) for a toy problem, like converting numbers to words ("123" -> "one two three").
- **Higher-Level:** Discuss the vanishing gradient problem in longer sequences and the conceptual need for attention mechanisms.

Experiment 13: Introduction to Neural Machine Translation (Conceptual and Library Use)

- **Low-Level:** Understand the high-level architecture of NMT with attention. Use a pre-trained NMT model (e.g., from Hugging Face Transformers) to perform translation between two languages.
- **Higher-Level:** Discuss the quality of the translation. Analyze how attention weights might highlight relevant words during translation.

Experiment 14: Question Answering (QA) using Pre-trained Models

- **Low-Level:** Use a pre-trained QA model (e.g., BERT-based model from Hugging Face) on a given context passage to answer questions.
- **Higher-Level:** Explore the types of questions the model can answer and its limitations. Discuss how the model identifies the answer span in the text.

Experiment 15: Building a Simple NLP Application (Mini-Project Kickoff)

- **Low-Level:** Choose a small NLP problem (e.g., a simple chatbot based on rules, a basic text summarizer for short texts, or a spam email classifier). Define the scope and necessary components.
- **Higher-Level:** Develop a preliminary pipeline for the chosen application. Present the problem definition, chosen approach, and initial data preparation steps. This lab can serve as a foundation for the final project.

10 Real-World Industry Case Studies for Practice in Labs

These case studies require students to apply their knowledge to realistic scenarios, encouraging critical thinking, problem formulation, and technology selection. They can be used for group discussions, mini-presentations, or as inspiration for project work.

1. **Customer Support Chatbot for E-commerce:**
   - **Scenario:** An e-commerce company wants to automate responses to common customer queries (e.g., "Where is my order?", "How to return an item?") using a chatbot.
   - **Questions:** How would you identify customer intent? What NLP tasks are involved (e.g., intent classification, entity extraction)? Discuss the challenges of handling ambiguous queries and escalating to human agents. Which libraries/models would you use?
2. **Automated Legal Document Review:**
   - **Scenario:** A law firm needs to rapidly review thousands of legal contracts to extract specific clauses, identify key entities (parties, dates), and flag inconsistencies.
   - **Questions:** What NLP techniques are most relevant for information extraction from highly structured but text-heavy documents? How would you handle domain-specific terminology? Discuss the challenges of precision and recall in a high-stakes environment like legal review.

3. **News Article Categorization and Sentiment Analysis:**
   - **Scenario:** A media monitoring company needs to automatically categorize incoming news articles into topics (e.g., "Politics", "Economy", "Sports") and determine the overall sentiment towards specific entities or events.
   - **Questions:** Which text representation and classification models are suitable for this task? How would you handle potential biases in sentiment analysis, especially for complex or nuanced topics? Discuss the importance of a well-curated training dataset.
4. **Healthcare: Electronic Health Record (EHR) Analysis:**
   - **Scenario:** Extracting structured information (e.g., diagnoses, medications, symptoms) from unstructured clinical notes in Electronic Health Records to aid medical research or clinical decision support.
   - **Questions:** What are the challenges of processing clinical language (e.g., abbreviations, medical jargon, informal writing)? Which NLP tasks (NER, relation extraction) are critical? Discuss privacy and data anonymization concerns.
5. **Social Media Brand Monitoring:**
   - **Scenario:** A brand wants to monitor social media (Twitter, Instagram comments) to understand public perception, identify trending topics related to their products, and detect customer complaints.
   - **Questions:** How would you handle noisy, informal, and short-form text? What are the challenges of sarcasm and irony detection? Discuss techniques for topic modeling and trend analysis.
6. **Resume Parsing for Recruitment:**
   - **Scenario:** A recruitment agency receives thousands of resumes daily and needs to automatically extract key information like skills, work experience, education, and contact details to match candidates with job descriptions.
   - **Questions:** What NLP tasks are essential for resume parsing? How would you handle variations in resume formats and writing styles? Discuss the challenge of extracting structured data from free-form text.
7. **Machine Translation for Customer Communications:**
   - **Scenario:** A global company needs to translate customer service emails and chat messages in real-time between multiple languages to provide seamless international support.
   - **Questions:** What are the main challenges of real-time machine translation (e.g., latency, domain-specific terminology, cultural nuances)? Discuss the importance of post-editing and quality evaluation.
8. **Automated Essay Scoring/Feedback:**
   - **Scenario:** An educational platform wants to develop an AI system to provide automated feedback or score essays written by students, focusing on aspects like coherence, grammar, and argument strength.
   - **Questions:** What NLP features would be indicative of essay quality? How would you measure coherence and logical flow? Discuss the ethical considerations of using AI for grading and the importance of human oversight.
9. **Patent Search and Analysis:**
   - **Scenario:** A research and development firm needs to efficiently search vast databases of patent documents, identify existing technologies, and analyze competitive landscapes.
   - **Questions:** How can NLP assist in semantic search beyond keyword matching? What techniques are useful for identifying technological trends and relationships between patents? Discuss the challenges of highly technical and formal language.

10. **Content Moderation for Online Platforms:**
    - **Scenario:** A social media platform needs to automatically detect and flag harmful content (e.g., hate speech, cyberbullying, spam) to maintain a safe online environment.
    - **Questions:** What are the linguistic complexities of identifying harmful content (e.g., implicit hate, evolving slang)? Discuss the trade-off between false positives and false negatives. How can human-in-the-loop systems be integrated with NLP moderation?

Project work/Assignment:

Group Assignments (Modules 2 & 3): Students will work in groups to implement solutions to specific NLP problems, such as:

- Building a sentiment classifier for a specific domain.
- Developing a custom PoS tagger or NER system for a specialized corpus.
- Implementing a simple N-gram language model or a basic word embedding model from scratch.

Mini-Project (End of Course): A culminating group project where students select an NLP application (e.g., a simple chatbot, text summarizer, machine translation system, or advanced sentiment analysis) and implement a solution using the learned techniques and modern NLP libraries. This will involve problem definition, data collection/preparation, model selection, implementation, evaluation, and presentation.

Assignment:

Students will have to do group assignments for Modules 2 & 3. As a part of their assignments, they will have to implement the solution to particular problems.

Text Book

T1. Daniel Jurafsky, and James Martin. “Speech and Language Processing” (3rd edition draft, available online at https://web.stanford.edu/~jurafsky/slp3/)

T2. Christopher D. Manning and Hinrich Schütze, “Foundations of Statistical Natural Language Processing”, 1st Edition, MIT Press, 1999.

References:

R1. Yoav Goldberg, "Neural Network Methods in Natural Language Processing", Morgan & Claypool Publishers, 2017. (Excellent for neural NLP focus)

R2. Pawan Goyal, “Natural Language Processing”. NPTEL. * E-Book Link for R2: https://drive.google.com/file/d/10nbwAJd-dv6htOOZVBgAvLd1WscI0RqC/view

Web Resources:

1. Stanford NLP Group: https://nlp.stanford.edu/
2. NLTK Documentation: https://www.nltk.org/
3. SpaCy Documentation: https://spacy.io/
4. Hugging Face Transformers: https://huggingface.co/docs/transformers/index (for advanced deep learning models)

NPTEL Course: https://onlinecourses.nptel.ac.in/noc22_cs98/course

Topics relevant to “SKILL DEVELOPMENT”: Assignment implementations in software, batch-wise presentations for developing Skill Development through Participative Learning techniques. This is attained through assessment components mentioned in the course handout, including laboratory experiments and the final project. Emphasis on practical coding, debugging, model evaluation, and clear communication of results.

Course Code: CSE3405	Course Title: Synergistic Neural Fuzzy Computing Type of Course: PEC – Theory & Lab Integrated			L-T-P-C	2	0	2	3
Version No.	1.0							
Course Pre-requisites	CSE2267							
Anti-requisites	NIL							
Course Description	This course provides a foundational understanding of Neural Networks and Fuzzy Logic, two powerful paradigms in artificial intelligence that often complement each other in "synergistic" or hybrid intelligent systems. Neural Networks draw inspiration from the human brain's structure, enabling computational systems to learn from data, recognize patterns, and solve complex problems in AI, machine learning, and deep learning. Fuzzy Logic, on the other hand, offers a method of reasoning that mimics human intuition, allowing for decision-making under uncertainty by embracing intermediate possibilities between absolute true and false. This course will cover the fundamental concepts of both Neural Networks and Fuzzy Logic theory, laying the groundwork for designing adaptive and intelligent systems capable of handling real-world complexity and imprecision.							
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Neural Networks and Fuzzy Logic** and attain **Skill Development** through **Participative Learning** techniques. The objective of this course is to familiarize learners with the core concepts of Neural **Networks and Fuzzy Logic**, and to foster **skill development** through **participative learning techniques**, emphasizing practical implementation and **problem-solving in hybrid intelligent systems.**							
Course Outcomes	**On successful completion of this course the students shall be able to:** 1. **Define** the fundamental concepts, architectures, and learning principles of Neural Networks. [Knowledge] 2. **Explain** the ideas behind common learning algorithms in Neural Networks, such as Perceptron, Least Mean Squares, and Back-propagation. [Knowledge] 3. **Discuss** the concepts of Fuzzy Sets, Fuzzy Operations, and Fuzzy Relations, distinguishing them from crisp set theory. [Comprehension] 4. **Demonstrate** the application of Fuzzy Logic concepts, including fuzzification, inference, and defuzzification, in designing Fuzzy Logic Controllers for various problems. [Application] 5. **Design and Implement** basic neural network models and fuzzy logic systems using appropriate tools and libraries. [Application]							
Course Content:								
Module 1	**Introduction to Neural Networks**	Quiz		Single Layer Perceptron	**9Classes**			

Topics:

Introduction to Neural Networks (NN): History and evolution of neural networks. Comparison between Artificial and Biological Neural Networks. Relationship with Artificial Intelligence, Machine Learning, and Deep Learning. **Neurons and Basic Models:** Structure of biological neurons. Models of single neurons (McCulloch-Pitts model). Different neural network models (feedforward, recurrent - conceptual overview). **Single Layer Perceptron:** Architecture, decision boundaries. The Perceptron learning algorithm. Convergence theorem. **Least Mean Square (LMS) Algorithm:** Derivation, Widrow-Hoff rule. Learning curves, impact of learning rates.

Module 2	**Multilayer Perceptron and Advanced Neural Networks**	Quiz		Multilayer Perceptron	**10 Classes**

Topics:
Multilayer Perceptron (MLP): Limitations of single-layer perceptrons (e.g., the XOR problem). Architecture of MLPs. **Back-propagation Algorithm:** Derivation of the back-propagation algorithm for training MLPs. **Heuristics for Improving Back-propagation:** Momentum, learning rate adaptation, regularization techniques (L1, L2 - brief overview). **Activation Functions:** Sigmoid, ReLU, Tanh. **Radial-Basis Function (RBF) Networks:** Introduction to RBF networks, interpolation, regularization. Learning strategies for RBF networks. **Kohonen Self-Organizing Maps (SOM):** Concepts of self-organization, competitive learning. The SOM algorithm and its application in data visualization and clustering. **Learning Vector Quantization (LVQ):** Supervised learning based on SOM.

Module 3	**Fuzzy Sets, Operations and Relations**	Quiz		Fuzzy Operations	**10Classes**

Topics:
Crisp Sets - An Overview: Review of classical set theory concepts (union, intersection, complement, cardinality). **Fuzzy Sets - Definition and Examples:** Introduction to fuzzy sets, membership functions. Comparison with crisp sets. Linguistic variables. **Properties of Fuzzy Sets:** Alpha (α) - Cuts and their properties. Support, Core, Height, Normality. **Representations of Fuzzy Sets:** Set-theoretic, graphical, and mathematical representations. **Extension Principle of Fuzzy Sets:** Application of fuzzy sets to fuzzy numbers and fuzzy arithmetic. **Fuzzy Operations:** Operations on Fuzzy Sets: Fuzzy Complements (e.g., standard complement), Fuzzy Intersections (e.g., min, product), Fuzzy Unions (e.g., max, probabilistic sum). Combinations of operations. **Aggregation Operations:** Weighted average, ordered weighted averaging (OWA) operators. **Fuzzy Relations:** Binary Fuzzy Relations, Composition of Fuzzy Relations (max-min, max-product). **Fuzzy Equivalence and Compatibility Relations:** Properties and applications.

Module 4	**Fuzzy Logic and Fuzzy Logic Controller**	Assignment		Developing Fuzzy Logic Controller	**10Classes**

Topics:
Fuzzy Logic: Comparison of Classical Logic, Multivalued Logic, and Fuzzy Logic. Fuzzy Propositions, Fuzzy Quantifiers, Linguistic Hedges (e.g., "very," "somewhat"). Inference from Conditional Fuzzy Propositions (Fuzzy Implication). Conditional and Qualified Propositions, and Quantified Propositions. **Fuzzy Controllers:** Overview of Fuzzy Logic Controllers (FLCs) and their architecture. **Fuzzification Module:** Converting crisp inputs to fuzzy values. **Fuzzy Rule Base:** Design of fuzzy IF-THEN rules. **Fuzzy Inference Engine:** Methods of inference (e.g., Mamdani, Sugeno). **Defuzzification Module:** Converting fuzzy outputs to crisp values (e.g., Centroid, Mean of Maxima). **Applications and Examples:** Detailed example of an FLC (e.g., inverted pendulum, washing machine control). **Neuro-Fuzzy Systems (Conceptual):** Brief introduction to Adaptive Neuro-Fuzzy Inference Systems (ANFIS) as a hybrid approach (concept only, no deep implementation).

Targeted Application & Tools that can be used:

1. **Python Libraries:** NumPy, Matplotlib, SciPy, scikit-learn, TensorFlow/Keras, PyTorch, `scikit-fuzzy` (or `Fuzzy-Logic` for basic fuzzy implementations).
2. **MATLAB:** Neural Network Toolbox, Fuzzy Logic Toolbox (for comparison and visual demonstration, if resources permit, but Python will be primary).

3. **Development Environment:** Google Colab, Jupyter Notebook, local Python environments.

Labs:

Tools: Google Colab, Python (NumPy, Matplotlib, scikit-learn, TensorFlow/Keras, scikit-fuzzy or custom fuzzy logic implementation).

Experiment 1: Simulating a Single Neuron and Perceptron Learning
- **Low-Level:** Implement a simple McCulloch-Pitts neuron. Simulate its behavior for different inputs and weights.
- **Higher-Level:** Implement the Perceptron learning algorithm from scratch to classify linearly separable data (e.g., AND gate, OR gate). Visualize the decision boundary.

Experiment 2: Least Mean Square (LMS) Algorithm for Regression
- **Low-Level:** Implement the LMS algorithm from scratch for a simple linear regression problem. Plot the error curve.
- **Higher-Level:** Experiment with different learning rates and initial weights. Analyze the impact on convergence speed and final error.

Experiment 3: Multilayer Perceptron (MLP) for XOR Problem
- **Low-Level:** Implement a basic MLP with one hidden layer using NumPy to solve the XOR problem. Manually define weights and test the network.
- **Higher-Level:** Implement the back-propagation algorithm from scratch to train the MLP for the XOR problem. Observe the learning process.

Experiment 4: MLP for Classification using Keras/TensorFlow
- **Low-Level:** Use Keras/TensorFlow to build and train an MLP for a simple classification dataset (e.g., Iris or MNIST small subset).
- **Higher-Level:** Experiment with different activation functions (sigmoid, ReLU, tanh) and optimizers (SGD, Adam). Analyze their impact on training speed and accuracy.

Experiment 5: Exploring Radial Basis Function (RBF) Networks
- **Low-Level:** Understand the concept of RBF. Implement a simple RBF network with fixed centers and widths to perform function approximation.
- **Higher-Level:** Discuss how to choose centers and widths (e.g., using k-means for centers). Compare its performance with an MLP on a similar problem.

Experiment 6: Self-Organizing Maps (SOM) for Clustering
- **Low-Level:** Implement a basic SOM algorithm for a 2D input dataset. Visualize the weight vectors after training.
- **Higher-Level:** Use MiniSom or a similar library to apply SOM for clustering on a slightly larger dataset. Interpret the resulting feature map.

Experiment 7: Fuzzy Set Definition and Operations
- **Low-Level:** Define various fuzzy sets with different membership functions (e.g., triangular, trapezoidal, Gaussian) for a given linguistic variable (e.g., "temperature").
- **Higher-Level:** Implement and visualize fuzzy operations (complement, union using max, intersection using min) on defined fuzzy sets.

Experiment 8: Alpha-Cuts and Extension Principle
- **Low-Level:** For a given fuzzy set, calculate and visualize its α-cuts for different α values.
- **Higher-Level:** Demonstrate the Extension Principle by applying a simple mathematical function (e.g., addition, multiplication) to fuzzy numbers.

Experiment 9: Fuzzy Relations and Compositions
- **Low-Level:** Define two binary fuzzy relations. Calculate and visualize their composition using max-min and max-product rules.
- **Higher-Level:** Discuss the properties of fuzzy equivalence and compatibility relations with examples.

Experiment 10: Fuzzification Module Implementation

- **Low-Level:** Implement a fuzzification module for a crisp input (e.g., sensor reading) to convert it into fuzzy values using defined membership functions.
- **Higher-Level:** Experiment with different shapes of membership functions and their impact on the fuzzified output.

Experiment 11: Fuzzy Rule Base and Inference Engine (Mamdani)

- **Low-Level:** Define a simple fuzzy rule base for a 2-input, 1-output system (e.g., fan speed control based on temperature and humidity).
- **Higher-Level:** Implement a Mamdani fuzzy inference engine to aggregate the fuzzy outputs from the rules. Visualize the aggregated fuzzy output.

Experiment 12: Defuzzification Methods

- **Low-Level:** Implement common defuzzification methods (e.g., Centroid of Area, Mean of Maxima, Bisector of Area) for a given fuzzy output set.
- **Higher-Level:** Compare the results of different defuzzification methods and discuss their implications.

Experiment 13: Building a Complete Fuzzy Logic Controller

- **Low-Level:** Integrate fuzzification, rule base, inference engine, and defuzzification to build a complete Fuzzy Logic Controller for a simple system (e.g., a "tip calculator" based on service quality and food quality).
- **Higher-Level:** Test the FLC with various inputs and analyze its control surface.

Experiment 14: Case Study: Fuzzy Control of a Simple System (e.g., Washing Machine)

- **Low-Level:** Design a fuzzy logic controller for a simplified washing machine (inputs: dirtiness, grease; output: wash time).
- **Higher-Level:** Implement and simulate the FLC. Analyze its performance and compare with a crisp control strategy.

Experiment 15: Introduction to Neuro-Fuzzy Systems (Conceptual & Library Use)

- **Low-Level:** Understand the conceptual architecture of ANFIS (Adaptive Neuro-Fuzzy Inference System).
- **Higher-Level:** Use a library if available (or conceptually discuss) how ANFIS combines neural network learning with fuzzy logic to adapt fuzzy rules and membership functions from data. Discuss potential applications where such a hybrid approach is beneficial.

10 Real-World Industry Case Studies for Practice in Labs

These case studies are designed to encourage critical thinking, problem formulation, and the application of synergistic neural fuzzy computing principles to practical scenarios. They can be used for group discussions, mini-presentations, or as inspiration for project work.

1. **Smart HVAC (Heating, Ventilation, and Air Conditioning) System:**
   - **Scenario:** Design an intelligent HVAC system for a building that adapts to varying occupancy, outdoor temperature, and user preferences to optimize energy consumption and maintain comfort.
   - **Questions:** How can fuzzy logic handle the imprecise human comfort notions and external weather conditions? How could a neural network learn optimal set points or predict energy demand based on historical data? Discuss how these two could synergize.
2. **Autonomous Vehicle Cruise Control:**
   - **Scenario:** Develop an adaptive cruise control system for autonomous vehicles that maintains safe distances and smooth acceleration/braking in varying traffic conditions (e.g., heavy traffic, highway driving).
   - **Questions:** Where would fuzzy logic excel (e.g., handling "close" or "far" distances,

"slow" or "fast" acceleration)? How could a neural network predict optimal speed adjustments based on sensory input from cameras/radars?

3. **Industrial Process Control (e.g., Chemical Reactor Temperature):**
   - **Scenario:** Control the temperature of a chemical reactor where inputs (e.g., reactant flow rate, coolant flow) are precise, but the desired output temperature has a tolerance range, and the process dynamics are complex or uncertain.
   - **Questions:** How can fuzzy logic handle the imprecise "desired temperature" range and rule-based control? Could a neural network model the non-linear dynamics of the reactor to provide better predictions for the fuzzy controller?
4. **Medical Diagnosis Support System:**
   - **Scenario:** Create a system to assist doctors in diagnosing diseases where symptoms can be vague, and patient data might be incomplete or imprecise.
   - **Questions:** How can fuzzy logic represent symptoms (e.g., "mild fever," "severe pain") and rule-based diagnostic criteria? Could a neural network learn complex patterns from patient medical records to identify high-risk cases or suggest likely diagnoses based on symptoms?
5. **Washing Machine Intelligent Control:**
   - **Scenario:** Design an advanced washing machine that automatically adjusts wash cycle parameters (water level, wash time, detergent amount) based on the "dirtiness" and "load size" of the clothes.
   - **Questions:** How can fuzzy logic handle the qualitative inputs like "very dirty" or "medium load"? How would you define fuzzy rules? Can a simple neural network learn to classify load type or dirtiness from sensor data?
6. **Robotics: Gripper Force Control:**
   - **Scenario:** Develop a robotic gripper that can grasp objects of varying fragility and textures with appropriate force to avoid crushing delicate items or dropping heavy ones.
   - **Questions:** How can fuzzy logic handle the imprecise concept of "fragile" or "slippery" and adjust gripping force? Could a neural network learn the optimal gripping force based on tactile sensor data for different object types?
7. **Smart Home Energy Management:**
   - **Scenario:** Optimize energy consumption in a smart home by controlling appliances, lighting, and climate based on human presence, time of day, weather forecasts, and user preferences.
   - **Questions:** How can fuzzy logic interpret "comfortable," "occupied," "daytime"? Could a neural network predict energy usage patterns based on historical data and provide input to the fuzzy controller for optimal scheduling?
8. **Image Processing: Edge Detection with Uncertainty:**
   - **Scenario:** Enhance edge detection in noisy or ambiguous images, where traditional thresholding might produce fragmented or false edges.
   - **Questions:** How can fuzzy logic define the "degree of edge-ness" for pixels rather than a crisp binary decision? Could a neural network learn to identify relevant features in image patches that inform the fuzzy edge detection process?
9. **Financial Risk Assessment:**
   - **Scenario:** Develop a system to assess the creditworthiness of loan applicants, considering both quantitative financial data and qualitative factors like "job stability" or "repayment history" (which can be subjective).
   - **Questions:** How can fuzzy logic represent these qualitative factors and combine them with quantitative data using fuzzy rules? Could a neural network learn complex non-

linear relationships between financial indicators and loan default risk, potentially providing input to the fuzzy system?

10. **Water Quality Management in Aquaculture:**
    - **Scenario:** Design an automated system for managing water quality (pH, oxygen levels, temperature) in aquaculture tanks to ensure optimal conditions for aquatic life, reacting to sensor readings and predicting potential issues.
    - **Questions:** How can fuzzy logic control pumps, aerators, and heaters based on imprecise "optimal ranges" and "critical levels"? Could a neural network forecast future water quality parameters based on current conditions and historical trends, providing anticipatory input for the fuzzy controller?

Project work/Assignment:

Group Assignments (Modules 2 & 4): Students will work in groups to implement solutions to specific problems. These may include:

- Building and training an MLP for a classification or regression task.
- Implementing a Self-Organizing Map for clustering or dimensionality reduction.
- Designing and simulating a Fuzzy Logic Controller for a real-world scenario (e.g., controlling a simple robot, smart climate control).

Mini-Project (End of Course): A culminating group project where students explore a problem that can benefit from a **hybrid Neural-Fuzzy approach** (even if conceptual) or an in-depth application of either Neural Networks or Fuzzy Logic to a complex control or decision-making problem. This will involve problem definition, data preparation/feature selection, model selection, implementation, evaluation, and presentation.

Textbook(s):

1. Haykin, Simon. "*Neural networks and learning machines*", 3/E. Pearson Education India, 2011. https://www.pearson.com/en-us/subject-catalog/p/Haykin-Neural-Networks-and-Learning-Machines-3rd-Edition/P200000003278/9780133002553
2. George J. Klir and Bo Yuan, "*Fuzzy Sets and Fuzzy Logic- Theory and Applications*", Prentice Hall of India, 2015. https://www.worldcat.org/title/fuzzy-sets-and-fuzzy-logic-theory-and-applications/oclc/505215200

References:

1. Shivanandam, Deepa S, "*Principles of Soft computing*", N Wiley India, 3rd Edition, 2018.https://www.wileyindia.com/principles-of-soft-computing-3ed.html
2. Timothy J. Ross, "*Fuzzy Logic with Engineering Applications*", Third Edition, Wiley, 2011. https://onlinelibrary.wiley.com/doi/book/10.1002/9781119994374
3. Kumar S., "*Neural Networks - A Classroom Approach*", Tata McGraw Hill, 2nd Edition 2017.https://www.worldcat.org/title/neural-networks-a-classroom-approach/oclc/56955342
4. Fakhreddine O. Karray, and Clarence W. De Silva. "*Soft computing and intelligent systems design: theory, tools, and applications*". Pearson Education, 2009.

Weblinks

1. For Haykin's book: https://www.pearson.com/en-us/subject-catalog/p/Haykin-Neural-Networks-and-Learning-Machines-3rd-Edition/P200000003278/9780133002553
2. For Klir & Yuan's book: https://www.worldcat.org/title/fuzzy-sets-and-fuzzy-logic-theory-and-applications/oclc/505215200
3. For Sivanandam & Deepa: https://www.wileyindia.com/principles-of-soft-computing-3ed.html
4. For Ross: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119994374
5. For Karray & De Silva: https://www.pearson.com/en-gb/search.html?q=Karray%20Soft-Computing-and-Intelligent-Systems-Design-Theory-Tools-and-Applications

Topics relevant to "Skill Development": Assignment implementations in software, batch-wise presentations are used for Skill Development through Participative Learning techniques. This is attained through assessment components mentioned in the course handout, including laboratory experiments and the final project. Emphasis on practical coding, designing intelligent systems, evaluating performance, and clear communication of results.

Course Code: CSE3506	**Course Title: Introduction to Bioinformatics** **Type of Course: PEC – Theory**	**L- T- P- C**	3	0	0	3

Version No.	2.0
Course Pre-requisites	NIL
Anti-requisites	NIL
Course Description	This course is designed to provide foundational knowledge in Bioinformatics, focusing on the principles and applications essential for analyzing biological data. Students will gain an understanding of fundamental molecular biology concepts related to DNA and Protein sequences, their structures, and the databases used to store them. The course covers crucial techniques such as pairwise sequence comparison, scoring matrix calculation, and various sequence alignment algorithms (e.g., Needleman-Wunsch, Smith-Waterman). Furthermore, it delves into motif discovery, an overview of structural bioinformatics, and an introduction to genome sequencing. The integrated lab component will provide hands-on experience with widely used bioinformatics tools.
Course Objective	The objective of this course is to familiarize learners with the core concepts of Introduction to Bioinformatics and to enhance their **Employability Skills** through **Participative Learning** techniques.
Course Outcomes	**CO1**: Understand the fundamental concepts of DNA and Protein sequences and structures. [Knowledge] **CO2**: Explain various bioinformatics file formats and the principles of sequence alignments.[Comprehension] **CO3**: Apply established techniques for sequence similarity searching and motif discovery for the analysis of DNA and Protein Sequences.[Application]
Course Content:	

Module 1	**Fundamentals of Bioinformatics**	Quiz	Comprehension based Quizzes and assignments;	**9 Classes**

Topics: Introduction to Molecular Biology: Cell, DNA, RNA, Transcription, Translation, Protein Folding, Gene Structure; Introduction to Bioinformatics: Definition, Components and fields of bioinformatics, Omics technologies (Genomics, Proteomics, Metabolomics); Basic principles of structural/functional analysis of biological molecules; Biological Data Acquisition; Types of DNA sequences (Genomic DNA, Mitochondrial DNA); DNA Sequencing tools and methods; Protein Sequencing and Structure Determination methods; Finding Reverse Complement of a DNA sequence.

Module 2	**Genome databases and Sequence Similarity**	Quizzes and assignments	Comprehension based Quizzes and assignments	**8 Classes**

Topics: Types and Classification of Genome Databases (e.g., GenBank, UniProt, PDB); DNA and Protein Sequence Retrieval Systems; Various DNA and Protein Sequence File Formats (e.g., FASTA, GenBank, GFF, PDB, SAM/BAM); Files for Multiple Sequence Alignment (e.g., ClustalW output); Files for Structural Data (e.g., PDB, CIF); Frequent words and k-mers in Text; String Reconstruction problem; Sequence Similarity Searching (Homology, Orthology, Paralogy); Sequence Similarity Searching Tools (NCBI BLAST: BLASTN, BLASTP, PSI-BLAST, Specialized BLASTs; FASTA); Significance of Sequence Alignments; Alignment Scores and Gap Penalties (match, mismatch, gap opening, gap extension).

Module 3	**DNA sequence analysis and applications**	Quizzes and assignments	Comprehension based Quizzes and assignments	**10 Classes**

Topics: Sequence Similarity Searches and Alignment Tools revisited; Finding Optimal Global Alignment using Needleman-Wunsch algorithm; Finding Optimal Local Alignment using Smith-Waterman algorithm; Heuristic Methods of Sequence Alignment (BLAST, FASTA revisited); Pair-wise and Multiple Sequence Alignments (e.g., ClustalW, T-Coffee, MAFFT); DNA Sequence Analysis; Motif in Protein Sequence; Motif Discovery using Gibbs sampling and Expectation-Maximization (EM) algorithm; Motif Finding tools; Gene Prediction Models: Introduction to Hidden Markov Models (HMM) in Bioinformatics, Generalized Hidden Markov Model (GHMM), Bayesian Methods for gene prediction. Overview of Structural Bioinformatics and Protein Structure Prediction (Homology Modeling, Threading, de novo prediction).

Targeted Application & Tools that can be used:
Targeted Application: Analyzing biological sequences (DNA, RNA, Protein), understanding evolutionary relationships, gene prediction, and protein function inference. This fundamental course is crucial for anyone entering fields requiring biological data analysis.
Professionally Used Software/Tools: BLAST, FASTA, ClustalW, MEGA, NCBI Databases (GenBank, UniProt, PDB), Expasy tools, Python with Biopython library.

Project work/Assignment:

Approach: Each batch of students (self-selected, up to 4 members per batch) will be allocated case studies/assignments.
Focus: Problem Solving, Data Analysis, and Application of Bioinformatics Tools. Projects will involve practical implementation and analysis using learned techniques.

Textbook(s):
T1. David W. Mount, Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, 2004.
T2. Arthur Lesk, Introduction to Bioinformatics, Fifth Edition, Oxford University Press, 2019.

References
R1. S. C. Rastogi, N. Mendiratta, P. Rastogi, Bioinformatics Methods and Applications, Fourth Edition, Prentice Hall India.
R2. Phillip Compeau & Pavel Pevzner, Bioinformatics Algorithms - An Active Learning Approach, 2nd Edition, Vol. I & II, Active Learning Publishers, 2015.

WEB-References
- https://puniversity.informaticsglobal.com:2229/login.aspx
- NCBI Handbooks and Resources: https://www.ncbi.nlm.nih.gov/books/
- ExPASy Bioinformatics Resource Portal: https://www.expasy.org/
- Biopython Documentation: https://biopython.org/

- EBI Training Resources: https://www.ebi.ac.uk/training/online/

Topics Relevant to Skill Development

Employability Skills:

- Participative Learning Techniques: Achieved through quizzes, comprehension-based assignments, and collaborative project work.
- Batch-wise Presentations: Students will present on selected topics such as:
  - String Reconstruction Problem
  - Sequence Similarity Searching (e.g., BLAST variants)
  - Alignment Scores and Gap Penalties
  - Protein Sequencing and its implications
  - Gene Prediction Models, especially Hidden Markov Models (HMM)
  - Finding similarities by performing Pairwise and Multiple Sequence Alignments
  - Evaluating Phylogenetic Trees (implied as an application of sequence alignment results).

urse Code: **E3407**	**urse Title: Algorithms in Computational Biology** **pe of Course: Program Core** **eory & Lab Integrated**	**T-P-C**	2	0	2	3

rsion No.		)
urse Pre-requisites		CSE3406 – Introduction to Bioinformatics
ti-requisites		L
urse Description		This course introduces fundamental algorithms and computational techniques extensively used to solve complex problems in modern biology. Building upon an understanding of bioinformatics concepts, students will delve into core algorithmic strategies such as dynamic programming, string matching, graph algorithms, and statistical methods. The curriculum covers their application to critical biological challenges like sequence analysis, genome assembly, motif discovery, and protein structure prediction. Through a blend of theoretical instruction and hands-on laboratory exercises, students will learn to apply algorithmic thinking to real-world biological datasets, analyze computational complexity, and evaluate the performance of different algorithmic solutions.
urse Objective		The objective of this course is to familiarize the learners with the concepts of Algorithms in Computational Biology and to attain Skill Development through Experiential Learning techniques.
urse tComes		On successful completion of the course the students shall be able to: **CO1:** Define key concepts in computational biology and the fundamental principles of algorithms. (Remember) **CO2:** Explain the significance and challenges of applying algorithms in analyzing diverse biological data. (Understand) **CO3:** Apply basic algorithmic strategies and advanced techniques like dynamic programming to solve core biological problems (e.g., sequence alignment, gene prediction). (Apply) **CO4:** Explore and implement algorithms for different types of biological data, considering their specific computational needs and evaluating solution performance.(Apply)
urse Content:		

dule 1	**Introduction**	signment			**5L+6P Sessions**
	<u>Topics:</u> History and principles of algorithms in a biological context; Types of algorithms				

	(deterministic, randomized, approximation); Introduction to computational complexity (Big O notation, time and space complexity); Relevance of efficiency in biological data processing; Introduction to common biological problems requiring algorithmic solutions.					
odule 2	gorithms Issues and Problems	signment			+8P Sessions	
	Topics: Asymptotic analysis of algorithms revisited (Omega, Theta notations, analyzing recursive algorithms); Introduction to NP-complete problems and their significance in bioinformatics; Polynomial reducibility; Classic combinatorial problems with biological relevance: Traveling Salesman Problem (TSP) as a model for sequence reassembly, Consecutive Integer Problem (CIP) and its applications; Review of fundamental sorting algorithms; Fibonacci problem and recursive vs. iterative solutions					
odule 3	gorithmic Approaches	signment			L+8PSessions	
	Topics: Algorithmic Paradigms: Linear Search, Exhaustive Search, Divide and Conquer; Introduction to Branch and Bound techniques; Iterative Algorithms (Expectation Maximization (EM) algorithm with biological examples); Hidden Markov Models (HMMs) and related algorithms: Forward and Backward algorithms for probability calculation, Viterbi algorithm for most probable path (decoding); Introduction to Discriminative Learning concepts in biological data; String Matching Algorithms: Naive approach, Knuth-Morris-Pratt (KMP) algorithm, Boyer-Moore algorithm; Graph Algorithms in bioinformatics (e.g., de Bruijn graphs for genome assembly, shortest path for metabolic pathways); Maximum Likelihood Algorithms for phylogenetic inference.					
odule 4	namic Programming & Methods	signment			+8P Sessions	
	pics: Principles of Dynamic Programming (optimal substructure, overlapping subproblems); Applications of Dynamic Programming in sequence alignment (Needleman-Wunsch, Smith-Waterman algorithms – revisited algorithmically); Heuristics tools and their underlying principles (BLAST, FASTA, ClustalW); Probabilistic/Statistical Methods (e.g., Gibbs sampling for motif discovery); Models of Evolution and relevant algorithms (e.g., Jukes-Cantor, Kimura 2-parameter models, phylogenetic tree reconstruction algorithms – algorithmic perspective); Specific biological problems: Partial and Double Digest Problems; Graph algorithms for DNA sequence assembly (e.g., Overlap-Layout-Consensus paradigm, introduction to CASP3, Phrap, Phred); Protein Structure Prediction algorithms (e.g., Chou-Fasman algorithm, introduction to comparative modeling)					
	rgeted Application & Tools **rgeted Application:** Efficient analysis of large-scale biological datasets, including genomics, proteomics, and sequence data. Developing computational solutions for gene prediction, sequence alignment, evolutionary analysis, and structural biology. **ftware/Tools:** Python with relevant libraries (e.g., Biopython, NumPy, SciPy), command-line bioinformatics tools (BLAST, FASTA, ClustalW), visualization tools, text editors/IDEs.					

	Project Work/Assignment 3. Assignment 1: Covers concepts and problems from Module 1 and Module 2. 4. Assignment 2: Covers concepts and problems from Module 3 and Module 4. **Project Work:** Each batch of students will undertake a mini-project involving the design and implementation of an algorithm to solve a specific problem in computational biology (e.g., a simplified gene finder, a small sequence aligner, or a tool for motif detection).
	Text Book **T1**. Phillip Compeau & Pavel Pevzner, Bioinformatics Algorithms: An Active Learning Approach, Vol. 1, 2nd Ed., Active Learning Publishers, 2019. **T2**. Michael T. Goodrich & Roberto Tamassia, Algorithm Design and Applications, Wiley, 2015. **T3**. Jason Kinser, Computational Biology: A Hypertextbook, 2nd Ed., Jones & Bartlett Learning, 2021. **T4**. Gautam B. Singh, Fundamentals of Bioinformatics and Computational Biology, Springer, 2015.
	References **R1**. Zhumur Ghosh & Bibekanand Mallick, Bioinformatics: Principles and Applications, Oxford University Press, 2014. **R2**. Steven Skiena, The Algorithm Design Manual, Springer, 2nd Ed., 2008. **R3**. Pavel Pevzner, Computational Molecular Biology: An Algorithmic Approach, MIT Press, 2000. **R4**. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd Ed., MIT Press, 2009. **R5**. Arthur Lesk, Introduction to Bioinformatics, 5th Ed., Oxford University Press, 2019. **Web Resources** • NPTEL Course on Bioinformatics Algorithms: https://onlinecourses.nptel.ac.in/noc25_cs06/preview • MIT OpenCourseWare - Algorithms for Computational Biology: https://ocw.mit.edu/courses/6-096-algorithms-for-computational-biology-spring-2005/pages/lecture-notes/ • Rosalind (Online Platform for Bioinformatics Problems): http://rosalind.info/
	Module I: Introduction to Algorithms • Experiment 1: Implement Sorting and Searching Algorithms LEVEL 1 : Implement Bubble, Merge Sort, Linear and Binary Search LEVEL 2 : Compare time complexity using real biological data (e.g., gene lengths) • Experiment 2: Time Complexity and Recursion using Fibonacci Series LEVEL 1 : Compare recursive and dynamic programming approaches LEVEL 2 : Visualize time/memory usage with time and memory_profiler modules **Module II: Algorithmic Problem Solving** • Experiment 3: Solve the Travelling Salesman Problem (TSP) LEVEL 1 : Use brute force or greedy algorithms LEVEL 2 : Simulate sequencing fragment reassembly as a path problem • Experiment 4: Knapsack Problem in Bioinformatics LEVEL 1 : Apply knapsack logic to protein interaction weighting or resource allocation

	Module III: String Matching and Sequence Analysis • Experiment 5: LEVEL 1 : Naive Pattern Matching Algorithm on DNA Sequence LEVEL 2 : Identify motifs like start/stop codons • Experiment 6: KMP and Boyer-Moore Algorithms LEVEL 1 : Compare performance on large FASTA datasets • Experiment 7: Regular Expression Matching in Genomics Data LEVEL 1 : Search for specific motifs **Module IV: Dynamic Programming** • Experiment 8: BLAST Query using Biopython (Online) LEVEL 1 : Submit a BLAST query and parse top hits • Experiment 9: DNA Read Assembly Simulation
	Topics Relevant to Skill Development **Employability:** Proficiency in designing, implementing, and analyzing algorithms specifically for bioinformatics problems. Strong data analysis skills for biological datasets. Ability to compare and select appropriate algorithmic tools. **Professional Ethics:** Maintaining professional integrity in handling sensitive biological data, ensuring reproducibility of results, and adhering to ethical guidelines in computational research and development.

urse Code: ;E3408	**urse Title: Statistical Methods for BioInformatics** **pe of Course: Program Core** **eory & Lab Integrated**	**T-P-C**	2	0	2	3

rsion No.		)
urse Pre-requisites		CSE3406 – Introduction to Bioinformatics
ti-requisites		L
urse Description		This course provides a comprehensive introduction to statistical methods used in bio-informatics and biological research. Starting with foundational probability and statistics, the course progresses to advanced bio-informatics applications such as Markov models, Bayesian inference, and biological sequence analysis. Emphasis is placed on practical, hands-on learning through integrated lab experiments using real-world biological datasets and bio-informatics tools.
urse Object		The objective of the course is to familiarize the learners with the concepts of Statistical Methods for BioInformatics and attain Skill Development through Experiential Learning techniques.
urse Out Comes		On successful completion, the student will be able to: **CO1**: Understand the basic concepts of bioinformatics including major databases. (Understand) **CO2**: Evaluate tools and techniques to characterize and manage biological sequence data. (Apply) **CO3**: Apply core biostatistical techniques including sampling and descriptive statistics. (Apply) **CO4**: Analyze biological data using statistical hypothesis testing and inferential methods. (Apply)
urse Content:		

odule 1	Bioinformatics Fundamentals and Database Systems	signment			**6L+6P Sessions**
	Topics: Bioinformatics definition, history, scope and applications, Bioinformatics web portals: NCBI, EBI, ExPASy, Biological databases: Classification of databases - primary (Genbank), secondary (PIR) and tertiary or composite (KEGG) databases, Sequence databases - DNA sequence databases (ENA,DDBJ), Protein sequence databases (Swissprot, PROSITE) .				
odule 2	quence Alignment	signment			**+8P Sessions**

	Topics: Basics of sequence alignment - match, mismatch, gaps, gap penalties, scoring alignment, Types of sequence alignment - pairwise and multiple alignment, local and global alignment, Dot matrix comparison of sequences, Scoring matrices - PAM and BLOSUM, Pairwise sequence similarity search by BLAST and FASTA					
odule 3		sic Concepts In Biostatistics	signment			**L+8PSessions**
	Topics: Introduction to Biostatistics, kinds of data and variables - based on nature (numericaldiscrete and continuous, categorical-ordinal and nominal) - based on source (primary and secondary data), sample size, sampling methods and sampling errors, Data tabulation and representation methods: graphical methods– stem and leaf plot, line diagram, bar graphs, histogram, frequency polygon, frequency curves; diagrammatic method- pie diagram, Measures of central tendency- mean, median, mode; merits and demerits, Measures of dispersion- range, variance, standard deviation, standard error and coefficient of variation; merits and demerits, Correlation and regression analysis and their applications to biolog.					
odule 4		ostatistics-Applications	signment			**+8P Sessions**
	pics: roduction to Probability- definition; Normal distribution: definition and properties, Hypothesis testing- steps in testing for statistical hypothesis, null and alternative hypothesis, level of significance- type-1 and type-2 errors, Test of significance for large samples- Z-test for means and proportions, Test of significance for small samples- student's t-test(one sample and two samples), Chi-square test and its applications- goodness of fit (not based on distribution), test of independence , Analysis of variance (One-way ANOVA) and their applications to biology .					
	oject work/Assignment: 1. Assignment 1 on (Module 1 and Module 2) 2. Assignment 2 on (Module 3 and Module 4)					
	Text Book **T1** Arthur Lesk, Introduction to Bioinformatics, Oxford University Press, 5th Ed., 2019 **T2** Wayne W. Daniel & Chad L. Cross, Biostatistics: A Foundation for Analysis in the Health Sciences, Wiley, 11th Ed., 2019 **T3** Gautam B. Singh, Fundamentals of Bioinformatics and Computational Biology, Springer, 2015					
	References **R1** Zhumur Ghosh & Bibekanand Mallick, Bioinformatics: Principles and Applications, OUP, 2014 **R2** David W. Mount, Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor, 2nd Ed., 2004 **R3** Marcello Pagano & Kimberlee Gauvreau, Principles of Biostatistics, CRC Press, 2nd Ed., 2018 **R4** S. C. Gupta & V. K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand **Web Resources** • https://www.ncbi.nlm.nih.gov					

	• https://archive.nptel.ac.in/courses/102/101/102101056/ • https://onlinecourses.nptel.ac.in/noc25_bt06/preview • https://www.ebi.ac.uk • https://www.expasy.org .
	Module I: Bioinformatics & Biological Databases **Level 1 Experiments: (Foundational knowledge and basic data retrieval)** • Experiment 1: Introduction to NCBI – Searching for DNA & protein sequences (Fundamental for accessing primary biological data) • Experiment 2: Exploring the EBI and ExPASy portals (Introduction to other major bioinformatics resources) • Experiment 3: Retrieving gene information from GenBank and ENA (Core skill for genetic studies) **Level 2 Experiments: (More specific data interpretation and specialized database usage)** • Experiment 4: Identifying protein domains using PROSITE and SwissProt (Involves analyzing specific features of proteins, building on basic sequence retrieval) **Module II: Sequence Alignment** **Level 1 Experiments: (Basic alignment types and widely used tools)** • Experiment 5: Pairwise sequence alignment using EMBOSS Needle (Introduces the concept of alignment) • Experiment 6: Local sequence alignment using BLAST (Essential and most commonly used tool for sequence similarity searches) **Level 2 Experiments: (More complex alignment scenarios and advanced analysis)** • Experiment 7: Global sequence alignment using Clustal Omega (Handles multiple sequences, which is more complex than pairwise alignment) • Experiment 8: Constructing phylogenetic trees using MEGA or Phylogeny.fr (Involves interpreting evolutionary relationships, requiring a deeper understanding of sequence data) **Module III: Basic Biostatistics** **Level 1 Experiments: (Fundamental statistical concepts and descriptive statistics)** • Experiment 9: Data collection and classification of variables (Prerequisite for any statistical analysis) • Experiment 10: Creating bar charts, pie charts, histograms using MS Excel or Python (matplotlib/seaborn) (Basic data visualization) • Experiment 11: Calculating mean, median, mode, standard deviation using statistical software (R/SPSS/Excel) (Core descriptive statistics, essential building blocks) **Level 2 Experiments: (Module III primarily focuses on basic concepts, so there are no Level 2 experiments within this module in this breakdown. The more advanced statistical analyses are covered in Module IV.)** **Module IV: Statistical Analysis** **Level 1 Experiments: (Module IV is inherently about more advanced statistical analysis, so there are no Level 1 experiments within this module in this breakdown. All experiments here build on the basic biostatistics from Module III.)** **Level 2 Experiments: (Inferential statistics and hypothesis testing)** • Experiment 12: Performing correlation and regression analysis in R (Involves

	understanding relationships between variables, which is an inferential task) • Experiment 13: Hypothesis testing using t-test and Z-test in SPSS/R (Core inferential statistical tests) • Experiment 14: Conducting chi-square test for independence (Another important inferential test for categorical data) • Experiment 15: One-way ANOVA application on biological data (More advanced hypothesis testing for comparing multiple group means)
	Topics relevant to SKILL DEVELOPMENT: **Topics relevant to development of "Employability":** Training in bioinformatics tools, biological databases, sequence analysis **Topics relevant to "PROFESSIONAL ETHICS":** Data privacy in genomic research, ethical use of biological databases

<table>
<tr><td>Course Code:
CSE3409</td><td colspan="3">Course Title:
Emerging Technologies in Big Data</td><td>L-T- P- C</td><td>2 -0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">CSE3156-Database Management System,</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">The purpose of this course is to provide the fundamentals of Big Data technology, emphasizing the importance of choosing suitable tools for processing and analyzing big data to gain insights. Students will acquire the knowledge and skills to select and use appropriate big data tools to solve business problems. The associated laboratory provides an opportunity to implement the concepts and enhance critical thinking and analytical skills. With a good understanding of Big Data technology fundamentals, students will gain practical experience in implementing them, enabling them to be effective solution providers for applications involving huge volumes of data.</td></tr>
<tr><td>Course Objectives</td><td colspan="7">The objective of the course is to familiarize the learners with the concepts of Big Data Technologies and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course, the students shall be able to:
CO1: Apply Map-Reduce programming on given datasets to extract required insights. (Application)
CO2: Employ appropriate Hadoop Ecosystem tools such as Sqoop, Hbase, Hive, to perform data analytics for a given problem. (Application)
CO3: Use Spark tool to analyze the given dataset for a given problem. (Application)</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Introduction to Hadoop</td><td>Programming Assignment</td><td colspan="2">Data Collection and Analysis</td><td colspan="3">10 Classes</td></tr>
<tr><td colspan="8">Topics
Introduction to Big Data and its importance: Basics of Distributed File System, Four Vs (Volume, Velocity, Variety, Veracity), Drivers for Big data, Big data applications, Structured, unstructured, semi-structured and quasi structured data. Big data Challenges - Traditional versus big data approach, The Big Data Technology Landscape: No-SQL.The Hadoop: History of Hadoop, Hadoop use cases, The Design of HDFS, Blocks and replication management, Rack awareness, HDFS architecture, HDFS Federation, Name node and data node, Anatomy of File write, Anatomy of File read. Hadoop MapReduce Paradigm: Map and reduce tasks, Job Tracker and Task Tracker, MapReduce execution pipeline, Key-value pair, Shuffle and sort, Combiner and Partitioner, APIs used to Write/Read files into/from Hadoop, Need for Flume and Sqoop. Anatomy of YARN: Hadoop 2.0 Features, NameNode High Availability, YARN Architecture, Introduction to Schedulers, YARN scheduler policies (FIFO, Fair, and Capacity scheduler).</td></tr>
</table>

Programming Assignment: Data Collection and Analysis

Module 2	Hadoop Ecosystem Tools	Programming Assignment	Data Collection and Analysis	8 Classes

Topics
Introduction to SQOOP: SQOOP features, Sqoop Architecture, Sqoop Import All Tables, Sqoop Export All Tables, Sqoop Connectors, Sqoop Import from MySQL to HDFS, Sqoop vs. Flume. Hive: Apache Hive with Hive Installation, Hive Data Types, Hive Table partitioning, Hive DDL commands, Hive DML commands, Hive SORT BY vs. ORDER BY, Hive Joining tables, Hive bucketing. Hbase: Introduction to HBase and its working architecture - Commands for creation and listing of tables - disabled and is_disabled of table - enable and is_enabled of table - describing and dropping of table - Put and Get command - delete and delete all command - commands for scan, count, truncate of tables.
Programming Assignment: Data Collection and Analysis

Module 3	Spark	Programming Assignment	Data analysis	8 Classes

Topics
Introduction to Apache Spark: A unified Spark, Who uses Spark and for what?, A Brief History of Spark, Spark versions and releases, Storage layers for Spark. Programming with RDDs: RDD Basics, Creating RDDs, RDD Operations, Passing functions to Spark, Common Transformations and Actions, Persistence. Spark SQL: Linking with Spark SQL, Using Spark SQL in Applications, Loading and Saving Data, JDBC/ODBC Server, User-defined functions, Spark SQL Performance. Scala: The Basics, Control Structures and functions, Working with arrays, Maps and Tuples.
Programming Assignment: Data Analysis

List of Laboratory Tasks:

1. Hadoop Installation and HDFS Commands

Level 1: Install Hadoop in pseudo-cluster mode. Practice HDFS Shell Commands for Files and Folders (e.g., `ls`, `mkdir`, `put`, `cat`).

Level 2: Utilize HDFS Shell Commands for advanced Management tasks (e.g., `setrep` for replication, `df` for disk usage, `du` for directory space, `chown` for ownership).

2. Basic Word Count MapReduce Program

Level 1: Develop a MapReduce program to find the number of occurrences of each word in an input file(s).

Level 2: Enhance the MapReduce program to perform a word search count, specifically looking for the occurrences of predefined keywords in a file.

3. Map Reduce Program for Weather Data Analysis

Level 1: Write a Map-reduce program to analyze the NCDC weather dataset (available at: https://github.com/tomwhite/hadoopbook/tree/master/input/ncdc/all) to find the average, maximum, and minimum temperatures for each year.

Level 2: Design a programming assignment to analyze social media data for business analytics, focusing on extracting insights relevant to specific business problems (e.g., trending topics, basic sentiment analysis from a sample dataset).

4. MapReduce for Data Aggregation and Matrix Operations

Level 1: Implement a MapReduce program to find the number of products sold in each country using a given sample dataset.

Level 2: Develop a MapReduce program to perform matrix multiplication.

5. Hive Basic Operations

Level 1: Install Hive and practice basic Hive commands (e.g., `CREATE TABLE`, `ALTER TABLE`, `DROP TABLE`).

Level 2: Apply Hive DDL and DML commands to create, manage, and query a student or employee database, demonstrating practical data manipulation.

6. Hive Advanced Commands - Partitioning

Level 1: Work with advanced Hive commands related to **Static Partitioning** and **Dynamic Partitioning**.

Level 2: Building on the previous experiment, select and apply the most suitable partitioning technique (static vs. dynamic) for a given dataset and query pattern, explaining the performance benefits.

7. Hive Advanced Commands - Bucketing

Level 1: Explore advanced Hive commands focusing on **Bucketing**.

Level 2: Continue the previous experiment by applying bucketing techniques to demonstrate the difference between partitioning and bucketing, highlighting scenarios where bucketing is more effective.

8. Hadoop Ecosystem Tools - Sqoop and HBase Installation & Data Movement

Level 1: Install Hadoop ecosystem tools, specifically **Sqoop** and **HBase**.

Level 2: Use Sqoop to **import data from a relational database** (e.g., MySQL) into Hadoop (HDFS/Hive/HBase).

9. HBase Basic Commands

Level 1: Practice basic HBase commands including general commands (e.g., `status`, `version`, `list`) and DDL commands (e.g., `create`, `describe`, `disable`, `enable`, `drop` tables).
Level 2: Apply HBase commands to create and manage tables for an insurance database or

employee dataset, defining appropriate column families and verifying table structures.

10. HBase Advanced Commands - DML Operations

Level 1: Work with advanced HBase commands for Data Manipulation Language (DML) operations (e.g., `put` to insert, `get` to retrieve, `scan` to iterate).

Level 2: Continue the previous experiment to demonstrate **CRUD (Create, Read, Update, Delete)** operations comprehensively on an HBase table using `put`, `get`, `delete`, and `scan` commands.

11. Apache Spark Installation and RDD Basics

Level 1: Install, deploy, and configure Apache Spark.

Level 2: Use **RDDs** and the `flatMap` transformation to count word occurrences in a file and then filter and output a list of words whose count is strictly greater than 4 using Spark.

12. Spark Word Count and Log Analysis

Level 1: Write an Apache Spark program to count word occurrences in a given text file and display only those words starting with 'a', sorted in ascending order of count.

Level 2: Analyze Apache access logs. Given a `log.txt` file (e.g., `127.0.0.1 - Scott [10/Dec/2019:13:55:36 -0700] "GET /server-status HTTP/1.1" 200 2326`), write a Spark program to read the records and display the total number of successful requests (indicated by HTTP 200 status code).

13. Scala Programming and Spark Data Analytics

Level 1: Write a Scala program to determine if a chess king can move from a given start cell to a target cell in one move. The program should take four numbers (column and row for start and end cells) as input and output "YES" or "NO".

Level 2: Perform **data analytics using Apache Spark on the Amazon food dataset** to find all pairs of items frequently reviewed together. The Spark application should: * Transpose the original dataset to obtain a Pair RDD of the type: `<User ID, List of Product IDs reviewed by the User>`. * Count the frequencies of all product pairs reviewed together. * Write all pairs of products that appear more than once, along with their frequencies (sorted by frequency), to an output folder.

Targeted Application & Tools that can be used:

- Business Analytical Applications
- Social media Data Analysis
- Predictive Analytics
- Tools: Hadoop Framework tools like MapReduce, Hive, Hbase, Sqoop, Spark.

Text Book

T1 Seema Acharya, Subhashini Chellappan. 2015. Big Data and Analytics. Wiley Publication.
T2 Matei Zaharia, Bill Chambers. 2018. SPARK: The Definitive Guide. O'Reilly.

References

R1 Tom White. 2016. Hadoop: The Definitive Guide. O'Reilly.
R2 Cay S. Horstmann. 2017. Scala for the Impatient. Wesley.

Topics relevant to development of "Skill Development

Real-time application development using Hadoop Ecosystem tools through Experiential Learning as mentioned in the course handout.

<table>
<tr><td>Course Code:
CSE3410</td><td colspan="3">Course Title: Statistical Techniques for Data Science

Type of Course: Theory</td><td>L-T-P-C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course provides an in-depth introduction to statistics and machine learning theory, methods, and algorithms essential for data science. Topics covered include multiple regression, kernel learning, sparse regression, generalized linear models, supervised and unsupervised learning, deep learning concepts, covariance learning, factor models, and principal component analysis. The course emphasizes understanding the applicability and limitations of these methods through the lens of mathematical statistics and by working with real-world datasets.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Statistical Foundations for Data Science and attain Employability through Participative Learning techniques</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of the course, the students shall be able to:

1. CO1: Understand the rise and significance of Big Data and dimensionality in various fields such as Biological Sciences, Health Sciences, Computer and Information Sciences, Economics and Finance, Business and Program Evaluation, Earth Sciences, and Astronomy.[Understand]
2. CO2: Develop a strong foundation in multiple linear regression and the Gauss-Markov theorem, applying various regularization techniques.[Apply]
3. CO3: Apply methods for inference in linear regression, including handling random designs and partial linear regression.[Apply]
4. CO4: Apply the power method and learn about factor models, structured covariance learning, and various clustering and variable selection techniques.[Apply]</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction</td><td>Assignment</td><td colspan="2">Programming</td><td colspan="4">No. of Classes:10</td></tr>
<tr><td colspan="9">Topics: Introduction to Big Data and its importance: Rise of Big Data and Dimensionality in Biological Sciences, Health Sciences, Computer and Information Sciences, Economics and Finance, Business and Program Evaluation, Earth Sciences, and Astronomy. Impact of Big Data, Impact of Dimensionality, Computation of Noise Accumulation, Spurious Correlation. Statistical theory - Aim of High-dimensional Statistical Learning.
Assignment: Conceptual understanding and literature review.</td></tr>
<tr><td>Module 2</td><td>Multiple Linear Regression</td><td>Assignment</td><td colspan="2">Programming</td><td colspan="4">No. of</td></tr>
</table>

				Classes:12

Topics: Multiple Linear Regression, The Gauss-Markov Theorem. Statistical Tests - Weighted Least-Squares, Box-Cox Transformation. Model Building and Basis Expansions, Polynomial Regression, Spline Regression, Multiple Covariates. Ridge Regression - Bias-Variance Tradeoff - Penalized Least Squares - Bayesian Interpretation - Ridge Regression Solution Path - Kernel Ridge Regression. Exponential family, Elements of generalized linear models, Maximum likelihood, Computing MLE: Iteratively Reweighted Least Squares, Deviance and Analysis of Deviance, Regularization parameters, Refitted Cross-validation, Extensions to Nonparametric Modeling.
Assignment: Problem-solving on regression equations and properties.

Module 3	Inference in linear regression	Assignment	Programming	**No. of Classes:14**

Topics: Inference in linear regression - Debiasing of regularized regression estimators, Choices of weights, Inference for the noise level, Inference in generalized linear models. Desparsified Lasso, Decorrelated score estimator - Test of linear hypotheses, Numerical comparison. Asymptotic efficiency, Statistical efficiency and Fisher information. Linear regression with random design, Partial linear regression. Gaussian graphical models - Inference via penalized least squares, Sample size in regression and graphical models, General solutions, Local semi-LD decomposition, Data swap, Gradient approximation.
Assignment: Derivations and proofs for inference methods.

Module 4	Principal Component Analysis	Assignment	Programming	**No. of Classes:9**

Topics: Principal Component Analysis - Introduction to PCA, Power Method. Factor Models and Structured Covariance Learning, Factor model and high-dimensional PCA. Cluster Analysis - K-means clustering, Hierarchical clustering, Model-based clustering, Spectral clustering, Data-driven choices of the number of clusters, Variable Selection in Clustering, Sparse K-means clustering, Sparse model-based clustering, Sparse Mixture of Experts Model. Correlation Screening, Generalized and Rank Correlation Screening, Nonparametric Screening, Sure Screening and False Selection.
Assignment: Case studies on PCA and clustering applications.

Targeted Application & Tools that can be used:

- Predictive Modeling in various scientific and business domains.
- Data Exploration and Insight Generation.
- Feature Engineering and Selection.
- Clustering and Segmentation.
- Tools: Python (with libraries like NumPy, SciPy, Pandas, Matplotlib, Seaborn, Scikit-learn), Torch (for deep learning concepts if introduced more deeply in lectures), Google Colaboratory, Spyder, Jupyter Notebook.

Project work/Assignment
Students will undertake a mini-project applying statistical techniques to a real-world dataset (e.g., from Kaggle or UCI Machine Learning Repository) that involves data cleaning, model building, and interpretation.

List of Laboratory Tasks:

1. Data Loading and Basic Statistics * **Level 1:** Load a structured dataset (e.g., CSV, Excel) into a Python environment (using Pandas) and compute basic descriptive statistics (mean, median, mode, standard deviation, variance) for numerical columns. * **Level 2:** Handle missing values and outliers in a given dataset using various techniques (e.g., imputation, removal, transformation), and visualize their impact on data distribution.

2. Introduction to Linear Regression * **Level 1:** Implement a simple linear regression model from scratch or using a basic library (e.g., `scikit-learn`'s `LinearRegression`) to predict a continuous variable from a single predictor, visualizing the regression line. * **Level 2:** Evaluate the performance of the simple linear regression model using metrics like R-squared, MSE, RMSE, and MAE, and interpret these metrics in the context of the problem.

3. Multiple Linear Regression * **Level 1:** Build a multiple linear regression model using `scikit-learn` to predict a continuous variable using several independent features from a dataset. * **Level 2:** Analyze the coefficients of the multiple linear regression model, interpret their significance, and identify the most influential predictors based on p-values or feature importance.

4. Model Diagnostics and Assumptions * **Level 1:** Generate common diagnostic plots for a linear regression model (e.g., residuals vs. fitted, Q-Q plot of residuals, scale-location plot) to check linearity, homoscedasticity, and normality of residuals. * **Level 2:** Identify and address violations of linear regression assumptions (e.g., multicollinearity using VIF, non-normality using transformations like Box-Cox), and re-evaluate model performance.

5. Regularization Techniques (Ridge Regression) * **Level 1:** Implement Ridge Regression using `scikit-learn` on a given dataset and observe the effect of the regularization parameter (alpha) on coefficients. * **Level 2:** Perform cross-validation (e.g., K-fold cross-validation) to find the optimal `alpha` for Ridge Regression and compare its performance with ordinary least squares regression, especially on datasets with multicollinearity.

6. Generalized Linear Models (Logistic Regression) * **Level 1:** Implement a Logistic Regression model for binary classification on a real-world dataset (e.g., predicting customer churn, disease presence). * **Level 2:** Evaluate the Logistic Regression model using classification metrics (e.g., accuracy, precision, recall, F1-score, ROC-AUC) and interpret the odds ratios for the predictors.

7. Inference for Regression Coefficients * **Level 1:** Perform t-tests for individual regression coefficients to determine their statistical significance in a multiple linear regression model. * **Level 2:** Construct confidence intervals for regression coefficients and interpret them, discussing the implications for the predictive power and reliability of the features.

8. Hypothesis Testing for Linear Models * **Level 1:** Perform an F-test to assess the overall significance of a multiple linear regression model. * **Level 2:** Conduct a formal hypothesis test (e.g., using ANOVA) to compare nested linear models or to test specific linear hypotheses about multiple coefficients.

9. Principal Component Analysis (PCA) - Basic * **Level 1:** Apply PCA to a high-dimensional numerical dataset to reduce its dimensionality, explaining the concept of principal components and explained variance. * **Level 2:** Visualize the data in the reduced PCA space (e.g., 2D or 3D scatter plots) and interpret the loadings of the principal components to understand feature contributions.

10. PCA - Advanced and Applications * **Level 1:** Determine the optimal number of principal components to retain using scree plots or cumulative explained variance. * **Level 2:** Use PCA for noise reduction or as a preprocessing step for another machine learning algorithm (e.g., clustering or classification), comparing the performance before and after PCA.

11. K-Means Clustering * **Level 1:** Implement K-means clustering on a dataset, specifying the number of clusters (k), and visualize the resulting clusters. * **Level 2:** Determine the optimal number of clusters (k) using methods like the Elbow Method or Silhouette Score, and evaluate the quality of the clustering.

12. Hierarchical Clustering * **Level 1:** Apply hierarchical clustering (agglomerative or divisive) to a dataset and visualize the results using a dendrogram. * **Level 2:** Experiment with different linkage methods (e.g., single, complete, average, ward) in hierarchical clustering and analyze how they affect the dendrogram structure and cluster formation.

13. Model-Based Clustering (Gaussian Mixture Models) * **Level 1:** Implement Gaussian Mixture Models (GMM) for clustering and compare its results with K-means on a dataset with varying cluster shapes. * **Level 2:** Use information criteria (e.g., BIC or AIC) to select the optimal number of components for a GMM and discuss the advantages of GMM over K-means for certain data distributions.

14. Variable Selection in Clustering / Correlation Screening * **Level 1:** Apply basic correlation analysis to identify highly correlated features in a dataset. * **Level 2:** Implement a simple variable screening method (e.g., Sure Independence Screening or filtering based on a statistical test) to select relevant features before applying clustering, and assess the impact on cluster quality.

15. Advanced Statistical Concepts in Practice * **Level 1:** Implement a basic test for linearity (e.g., plotting residuals against predictors or using a simple F-test for non-linear terms) or check for homoscedasticity using a residual plot. * **Level 2:** Explore the concept of "spurious correlation" by analyzing a randomly generated dataset or a real-world dataset where non-causal correlations might appear, and discuss methods to mitigate misinterpretation. (This connects back to Module 1's "Spurious Correlation" topic).

Text Book

T1: Fan, J., Li, R., Zhang, C.-H., and Zou, H. (2020). Statistical Foundations of Data Science. CRC Press.

T2: Wainwright, M. J. (2019). High-dimensional statistics: A non-asymptotic viewpoint. Cambridge University Press.

References

R1: James, G., Witten, D., Hastie, T.J., Tibshirani, R. and Friedman, J. (2013). An Introduction to Statistical Learning with Applications in R. Springer, New York.

R2: Hastie, T.J., Tibshirani, R. and Friedman, J. (2009). The elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed). Springer, New York.

R3: Buehlmann, P. and van de Geer, S. (2011). Statistics for High-Dimensional Data: Methods, Theory and Applications. Springer, New York.

Book Link:

- Fan, J., Li, R., Zhang, C.-H., and Zou, H. Statistical Foundations of Data Science. CRC Press.

E-book Link:

- W. N. Venables, D. M. Smith and the R Core Team, https://cran.r-project.org/doc/manuals/R-intro.pdf, October, 2022.

Web Resources:

- https://www.youtube.com/playlist?list=PLOU2XLYxmsIK9qQfztXeybpHvru-TrqAP (General YouTube resources for data science)
- https://presiuniv.knimbus.com/user#/ (University specific resource portal)

Topics relevant to "EMPLOYABILITY SKILLS":

Asymptotic efficiency, Statistical efficiency, and Fisher information.

Linear regression with random design, Partial linear regression. These topics are crucial for developing Employability Skills through Participative Learning techniques, as they enable students to understand the theoretical underpinnings and practical considerations for robust statistical modeling in real-world data science applications. This is attained through the assessment components mentioned in the course handout, including programming assignments and project work.

Catalogue prepared by	
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

<table>
<tr><td>Course Code: CSE3411</td><td colspan="3">Course Title: Predictive Analytics and Applications
Type of Course: Program Core</td><td>L- T-P- C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">MATXXXX – Probability and Statistics</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course is conceptual in nature, focusing on modern data analytic concepts. Students will gain the skills necessary for analyzing and synthesizing datasets to support effective decision-making in various organizational contexts. The course emphasizes the practical application of predictive analytics techniques to real-world business scenarios.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is skill development of students by using Learning techniques.</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of the course, the students shall be able to:
• CO1: Define the nature of analytics and its applications. (Remember)
• CO2: Summarize the concepts of predictive analytics and data mining. (Understand)
• CO3: Construct analytical tools in business scenarios to achieve competitive advantage. (Apply)
• CO4: Build real-world insights using decision trees and time series analysis methods in a dynamic business environment. (Apply)</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction to Predictive Analytics</td><td>Self-Learning</td><td colspan="2">Applications of analytics</td><td colspan="4">7 Sessions</td></tr>
<tr><td colspan="9">Topics: Analytics - Definition, importance, Analytics in decision making, Applications, Challenges, Experts' perception on analytics; Popularity in Analytics; Predictive analytics in business Scenarios - case studies.
Learning Activities: Self-Learning, Discussions on applications of analytics.</td></tr>
<tr><td>Module 2</td><td>Principles and Techniques</td><td>Case analysis</td><td colspan="2"></td><td colspan="4">8 Sessions</td></tr>
</table>

Topics: Predictive modeling: Propensity models, cluster models, collaborative filtering, applications and limitations. Statistical analysis: Univariate Statistical analysis, Multivariate Statistical analysis.
Learning Activities: Case analysis, Hands-on exercises with basic statistical analysis.

Module 3	**Model Selection**	Participative Learning & Case Analysis		**7 Sessions**

Topics: Preparing to model the data: supervised versus unsupervised methods, statistical and data mining methodology, cross-validation, overfitting, bias-variance trade-off, balancing the training dataset, establishing baseline performance.
Topics (Model Types): Measuring Performance in Regression Models - Linear Regression and Its Cousins - Non-Linear Regression Models - Regression Trees and Rule-Based Models. Measuring Performance in Classification Models - Discriminant Analysis and Other Linear Classification Models - Non-Linear Classification Models.
Learning Activities: Participative Learning & Case Analysis, Model building and evaluation exercises.

Module 4	**Time Series Analysis**	Discussion & Presentation		8 Sessions

Topics: Time series Model: ARMA, ARIMA, ARFIMA. Temporal mining - Box-Jenkins method, temporal reasoning, temporal constraint networks.
Learning Activities: Discussion & Presentation on time series forecasting applications.

Targeted Application & Tools that can be used:

- Business Forecasting (Sales, Demand, Stock Prices)
- Customer Behavior Prediction (Churn, Propensity to Buy)
- Risk Assessment (Credit Risk, Fraud Detection)
- Recommendation Systems
- Market Basket Analysis
- Tools: Python (Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Statsmodels), R (tidyverse, caret, forecast, ggplot2), Jupyter Notebook, RStudio.

Text Book
T1 Jeffrey Strickland, Predictive Analytics Using R, Simulation Educators, Colorado Springs, 2015.
T2 Max Kuhn and Kjell Johnson, Applied Predictive Modeling, 1st edition, Springer, 2013.

References

R1: Dinesh Kumar, U. (2021). *Business Analytics: The Science of Data-Driven Decision Making*.

R2: Albright, S. Christian and Winston, Wayne L. (2012). *Business Analytics - Data Analysis & Decision Making*, 5th Edition, Cengage Publication.

E-book Links:

1. Raman, R., Bhattacharya, S., & Pramod, D. (2018). *Predict employee attrition by using predictive analytics*. Benchmarking: An International Journal. https://www-emerald-com-

presiuniv.knimbus.com/insight/content/doi/10.1108/BIJ-03-2018-0083/full/html

2. Jing, Z., Luo, Y., Li, X., & Xu, X. (2022). *A multi-dimensional city data embedding model for improving predictive analytics and urban operations*. Industrial Management & Data Systems, (ahead-of-print). https://www-emerald-com-presiuniv.knimbus.com/insight/content/doi/10.1108/IMDS-01-2022-0020/full/html
3. Singh, R., Sharma, P., Foropon, C., & Belal, H. M. (2022). *The role of big data and predictive analytics in employee retention: a resource-based view*. International Journal of Manpower. https://www-emerald-com-presiuniv.knimbus.com/insight/content/doi/10.1108/IJM-03-2021-0197/full/html
4. Mishra, D., Luo, Z., Hazen, B., Hassini, E., & Foropon, C. (2018). *Organizational capabilities that enable big data and predictive analytics diffusion and organizational performance: A resource-based perspective*. Management Decision. https://www-emerald-com-presiuniv.knimbus.com/insight/content/doi/10.1108/MD-03-2018-0324/full/html

Web Resources:

- W1: https://www.sas.com/en_in/insights/analytics/predictive-analytics.html
- W2: https://www.techtarget.com/searchbusinessanalytics/definition/predictive-analytics
- W3: https://www.cio.com/article/228901/what-is-predictive-analytics-transforming-data-intofuture-insights.html
- W4: https://www.simplilearn.com/what-is-predictive-analytics-article
- W5: https://www.northeastern.edu/graduate/blog/predictive-analytics/
- W6: https://www.marketingevolution.com/knowledge-center/the-role-of-predictive-analytics-in-data-driven-marketing

Swayam & NPTEL Video Lecture Sessions on Predictive Analytics:

1. https://onlinecourses.swayam2.ac.in/imb20_mg19/preview
2. https://onlinecourses.nptel.ac.in/noc19_mg42/preview

Case References:

1. Predictive Analytics Industry Use Cases.
2. https://www.rapidinsight.com/blog/11-examples-ofpredictive-analytics/
3. Srinivasan Maheswaran (2017). *Predictive Analytics – Employee Attrition Case study*.

List of Laboratory Tasks:

Module 1: Introduction to Predictive Analytics (Practical Application)

1. **Exploring Analytics Applications**
    1. **Level 1:** Identify and list three distinct real-world applications of predictive analytics in different industries (e.g., healthcare, finance, retail) based on provided case studies or online resources. Briefly describe the problem solved and the type of prediction made.
    2. **Level 2:** Select one detailed case study of predictive analytics. Analyze the challenges faced during its implementation and propose potential solutions or

alternative approaches based on your understanding.

Module 2: Predictive Modeling Principles and Techniques

Data Loading and Basic Statistical Insights

1. **Level 1:** Load a sample business dataset (e.g., customer demographics, sales records) into a programming environment (e.g., Python with Pandas, R) and perform initial data inspection, including displaying data types, dimensions, and the first few rows.
2. **Level 2:** Perform initial data cleaning by identifying and handling common issues like **missing values** (e.g., imputation with mean/median, removal) and obvious **outliers** (e.g., using IQR method). Summarize the impact of these cleaning steps on the data's characteristics.

Descriptive and Univariate Statistical Analysis

1. **Level 1:** Calculate and interpret **univariate descriptive statistics** (e.g., mean, median, mode, standard deviation, quartiles) for key numerical variables in the business dataset (e.g., product prices, customer age, transaction amount).
2. **Level 2:** Generate insightful **visualizations** (e.g., histograms, box plots, density plots) for univariate analysis. Use these plots and statistical summaries to compare distributions of key variables across different categorical groups (e.g., sales by region).

Bivariate Statistical Analysis and Relationships

1. **Level 1:** Compute and interpret **correlation coefficients** (e.g., Pearson's r) between pairs of numerical variables (e.g., advertising spend vs. sales, customer income vs. purchase frequency) to understand their strength and direction.
2. **Level 2:** Create effective **scatter plots with regression lines** to visually represent relationships between two continuous variables. Discuss the potential implications of these observed relationships for building predictive models.

Introduction to Propensity Models (Logistic Regression)

1. **Level 1:** Build a basic **logistic regression model** to predict a binary outcome (e.g., customer churn: Yes/No, loan default: Yes/No) using a single predictor variable.
2. **Level 2:** Extend the logistic regression model to include multiple predictor variables. Interpret the resulting **coefficients** and their associated odds ratios. Evaluate the model's overall accuracy using simple classification metrics.

Customer Segmentation using Cluster Models (K-Means)

1. **Level 1:** Apply **K-Means clustering** to a customer dataset to identify natural groupings of customers based on their characteristics (e.g., purchasing behavior, demographics).
2. **Level 2:** Experiment with different numbers of clusters (K) and use evaluation

metrics (e.g., **Silhouette Score**, **Elbow Method**) to determine the optimal K. Profile the key characteristics of each identified customer cluster to understand their segments.

Introduction to Collaborative Filtering (Conceptual Application)

1. **Level 1:** Research and explain the fundamental concept of **user-based** or **item-based collaborative filtering**. Discuss its broad applications in recommendation systems (e.g., Netflix, Amazon) using a relevant conceptual case study.
2. **Level 2:** For a small, simplified dataset (e.g., a few users and items with ratings), describe or outline the steps to manually calculate user or item similarity and generate basic recommendations. (Full implementation is beyond scope, focus on algorithmic understanding).

Module 3: Model Selection and Evaluation

Understanding Cross-Validation and Overfitting

1. **Level 1:** Manually split a given dataset into training and testing sets. Train a simple regression or classification model on the training set and evaluate its performance on both the training and testing sets. Discuss the observation of any performance gap.
2. **Level 2:** Implement **K-fold cross-validation** for a chosen predictive model (e.g., linear regression, decision tree). Analyze how this technique helps in obtaining a more robust estimate of model performance and effectively mitigating overfitting compared to a simple train/test split.

Regression Tree Construction and Evaluation

1. **Level 1:** Build a **regression tree model** to predict a continuous variable (e.g., house price, sales volume) and visualize its basic tree structure.
2. **Level 2:** Interpret the decision rules within the regression tree. Apply **pruning techniques** (e.g., setting max_depth, min_samples_leaf) to optimize the tree and evaluate its performance using metrics like **MSE/RMSE** against a simple baseline model.

Classification Tree Construction and Evaluation

1. **Level 1:** Construct a **classification tree model** to predict a categorical outcome (e.g., customer segment, loan status) and visualize the resulting tree.
2. **Level 2:** Evaluate the classification tree using comprehensive metrics such as a **confusion matrix**, **precision, recall, and F1-score**. Discuss the trade-offs involved in different pruning strategies for classification trees in terms of bias-variance.

Linear Regression and its Extensions (Polynomial Regression)

1. **Level 1:** Implement a **multiple linear regression model** on a dataset and perform basic checks for its assumptions (e.g., linearity through residual plots).
2. **Level 2:** Apply **polynomial regression** to capture non-linear relationships in data.

Compare its fit and performance with simple linear regression, and discuss the challenges of overfitting when using higher-order polynomials

Non-Linear Classification Models (Support Vector Machines)

1. **Level 1:** Implement a **Support Vector Machine (SVM)** with a linear kernel for a basic classification task on a given dataset.
2. **Level 2:** Experiment with different **kernels** (e.g., Radial Basis Function (RBF), polynomial) in SVMs and tune their hyperparameters (e.g., C, gamma) to optimize classification performance on a potentially complex, non-linearly separable dataset.

Module 4: Time Series Analysis

Time Series Data Preparation and Visualization

1. **Level 1:** Load a time series dataset (e.g., daily stock prices, monthly sales figures) and generate basic time series plots to visually identify trends, seasonality, and cyclical patterns.
2. **Level 2:** Decompose a time series into its trend, seasonal, and residual components (e.g., using additive or multiplicative models). Perform necessary **differencing** to achieve stationarity, demonstrating the impact on the time series characteristics.

ARMA/ARIMA Model Building and Forecasting

1. **Level 1:** Use **ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots** to identify appropriate **ARIMA model parameters** (p, d, q) for a given stationary time series.
2. **Level 2:** Fit an **ARIMA model** to a time series dataset, evaluate its fit using residual analysis (e.g., checking for white noise residuals), and generate short-term forecasts with confidence intervals.

Real-world Time Series Forecasting Application

1. **Level 1:** Apply a fitted ARIMA model (or a simpler exponential smoothing model like Holt-Winters) to forecast future values for a business metric (e.g., demand forecasting for a product, website traffic prediction) and visualize the actuals versus forecasts.
2. **Level 2:** Critically assess the **forecast accuracy** using appropriate metrics (e.g., MAE, RMSE, MAPE) for the business forecasting application. Discuss potential reasons for forecast errors and propose strategies to improve the model or collect additional data.

Topics relevant to development of "Skill Development": Application of Business Analytics to enhance customer satisfaction and firm's success. This is achieved through hands-on laboratory exercises focusing on practical problem-solving.

Topics relevant to development of "Environment and sustainability": Focus on Predictive analytics to minimize errors in decision-making, thereby optimizing resource allocation and

reducing waste, contributing to sustainable practices.

Course Code: CSE3412	Course Title: Data Mining Type of Course: Discipline Elective/ Theory Only Course		L- T-P- C	3	0	0	3
Version No.		2.0					
Course Pre-requisites		MAT1003 – Applied Statistics					
Anti-requisites		**NIL**					
Course Description		This course provides a comprehensive introduction to the field of **data mining**, covering its fundamental concepts, applications, and practical challenges. Students will learn about essential data pre-processing techniques, various data mining tasks such as **association rule mining**, **classification**, and **clustering**, along with methods for **outlier detection**. The course also delves into recent trends and advancements in the data mining landscape, equipping students with the knowledge to apply these techniques to real-world problems.					
Course Objective		The primary objective of this course is to familiarize learners with the core concepts of **Data Mining** and enhance their **employability** through practical **problem-solving methodologies**.					
Course Out Comes		On successful completion of the course, students shall be able to: • **CO1:** Apply various **pre-processing techniques** necessary for diverse data mining tasks.[Apply] • **CO2:** Understand the functionality and working principles of various **data mining algorithms**.[Understand] • **CO3:** Appreciate the **strengths and limitations** of different data mining models.[Evaluate] • **CO4:** Understand the **advances in data mining** and their applicability in real-life scenarios.[Apply]					
Course Content:							

Module 1	**Introduction to Data Mining**	Assignment		Data Collection	**5 Sessions**
	Topics: Introduction to Data Mining – Data Mining Goals – Stages of the Data Mining Process – Data Mining Techniques – Merits and Demerits. **Learning Activities**: Self-learning through readings and discussions on the scope and impact of data mining.				
Module 2	**Data preprocessing**	Quiz		Problem Solving	**9 Sessions**
	Topics: Types of Data (structured, unstructured, semi-structured, categorical, numerical) – Pre-processing Steps (data cleaning, integration, transformation, reduction) – Data Preprocessing Techniques (normalization, aggregation, sampling, dimensionality reduction) – Similarity and Dissimilarity Measures (Euclidean, Manhattan, Cosine, Jaccard). **Learning Activities:** Problem-solving exercises involving various data preprocessing				

scenarios.

Module 3	Data Mining – Frequent Patterns	Assignment		Problem Solving	7 Sessions

Topics: Market Basket Analysis – Concepts of Itemsets, Support, Confidence, and Lift – Generating Frequent Itemsets and Association Rules Efficiently – Apriori Algorithm – FP-Growth Algorithm.
Learning Activities: Problem-solving sessions on generating frequent itemsets and association rules from transactional data.

Module 4	Classification and clustering	Assignment		Problem Solving	11 Sessions

Topics (Classification): Introduction to Classification – Decision Tree Induction (ID3, C4.5) – Bayesian Classification (Naïve Bayes) – Classification by Back Propagation (Neural Networks basics) - Lazy Learners (k-Nearest Neighbors) – Model Evaluation and Selection Techniques (Confusion Matrix, Accuracy, Precision, Recall, F1-Score, Cross-validation) to Improve Classification Accuracy.
Topics (Clustering): Introduction to Clustering Analysis – Partitioning Methods (K-Means, K-Medoids) – Hierarchical Methods (Agglomerative, Divisive) – Density-Based Methods (DBSCAN).
Learning Activities: Practical exercises in applying various classification and clustering algorithms to datasets.

Module 5	Outlier detection & Data mining trends	Assignment		Problem Solving	5 Sessions

Topics: Anomaly Detection Preliminaries – Different Outlier Detection Techniques (statistical methods, distance-based, density-based, isolation forests) – Introduction to Web Mining – Introduction to Text Mining – Demonstration of Weka Tool for data mining tasks.
Learning Activities: Discussion of real-world outlier scenarios and exploration of emerging data mining areas.

Project work/Assignment:

Assignments

- From the dataset given, find the Entropy, Gain value of the attributes and also draw the decision tree using entropy for the given dataset.
- Transactional Data Base, D given below which contains set of items find the frequent item set using the Apriori Algorithm and generate the Association Rules. Minimum Support count is 2%. Minimum confidence is 60%.

T_{id}	Items
10	1, 3, 4
20	2, 3, 5
30	1, 2, 3, 5
40	2, 5

Text Book

	T1: Tan P. N., Steinbach M & Kumar V. “Introduction to Data Mining”, Pearson Education, 2016.
	References **R1**: Han J & Kamber M, “Data Mining: Concepts and Techniques”, Elsevier, Second Edition, 2006. **R2**: G K Gupta, “Introduction to Data Mining with Case Studies”, PHI, Third Edition, 2014. **R3**: Alex Berson and Stephen J. Smith, “Data Warehousing, Data Mining and OLAP”, Tata McGraw – Hill. **Additional Web-Based Resources** • https://onlinecourses.swayam2.ac.in/cec20_cs12/preview (Related to "Data Mining: Concepts and Techniques, Jiawei Han, Micheline Kamber and Jian Pei, Morgan Kaufmann Publishers, 2012.") • https://puniversity.informaticsglobal.com:2284/ehost/detail/detail?vid=7&sid=e2d7362a-fd30-49a9-8f03-93e963521dbd%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=377411&db=nlebk • https://nptel.ac.in/courses/105105157
	Topics relevant to “EMPLOYABILITY SKILLS”: Data Mining Techniques and FP-Growth are directly relevant for developing Employability Skills through Participative Learning techniques. This hands-on problem-solving approach is reinforced through the assessment components mentioned in the course handout.

Course Code: CSE3413	**Course Title: No SQL Data Management** **Type of Course: Program Core**	**L-T-P-C**	2	0	2	3
Version No.		1.0				
Course Pre-requisites		CSE3156 – Database Management System				
Anti-requisites		NIL				
Course Description		This course provides a comprehensive understanding of non-relational database systems, emphasizing their emergence as scalable and flexible alternatives to traditional relational databases. It covers various NoSQL data models, including key-value, document, column-family, and graph databases, exploring their structure, use cases, and design principles. Students will learn about the challenges of data persistence, concurrency, and integration, and how aggregate-oriented models address these issues. The course delves into distribution models such as sharding and replication, the implications of the CAP theorem, and consistency trade-offs. Additionally, it introduces Map-Reduce for large-scale data processing and guides students in modeling data for access efficiency. Practical applications and limitations of each NoSQL type are discussed, preparing students to design robust, scalable data solutions for real-world applications.				
Course Objectives		The objective of the course is to introduce students to the principles, models, and practical applications of NoSQL databases. It aims to equip students with the knowledge and skills necessary to understand the advantages of NoSQL over traditional relational databases, comprehend various data models (key-value, document, column-family, and graph), explore distribution models for scalability and availability, and apply Map-Reduce and other techniques for data processing.				
Course Out Comes		On successful completion of the course, the students shall be able to: • **CO1**: Understand the limitations of relational databases and explain the motivation behind the emergence of NoSQL databases.[Understand] • **CO2**: Differentiate between various NoSQL data models such as key-value, document, column-family, and graph databases based on structure, use cases, and performance.[Analyze] • **CO3**: Analyze and apply appropriate distribution strategies including sharding, replication, and consistency models in distributed NoSQL systems.[Apply] • **CO4**: Design and implement data access patterns using aggregate-oriented modeling and schema-less approaches for scalable NoSQL applications.[Creation] • **CO5**: Utilize Map-Reduce and other data processing techniques to handle large-scale data operations efficiently in NoSQL				

		environments.[Apply]			
Course Content:					
Module 1	**Introduction to NoSQL and Aggregate-Oriented Data Models**	**Quiz**		**Knowledge based quiz**	**No. of sessions:8**
	Topics: Why NoSQL? The Value of Relational Databases, Getting at Persistent Data, Concurrency, Integration, A (Mostly) Standard Model, Impedance Mismatch, Application and Integration Databases, Attack of the Clusters, The Emergence of NoSQL, Aggregate Data Models; Aggregates, Example of Relations and Aggregates, Consequences of Aggregate Orientation, Key-Value and Document Data Models, Column-Family Stores, Summarizing Aggregate-Oriented Databases. More Details on Data Models; Relationships, Graph Databases, Schema less Databases, Materialized Views, Modelling for Data Access. **Assessment:** Knowledge-based quiz				
Module 2	**Distributed Data Systems and Consistency Models**	**Assignment**		**Data Visualization**	**No. of sessions:10**
	Topics: Distribution Models; Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication Consistency, Update Consistency, Read Consistency, Relaxing Consistency, The CAP Theorem, Relaxing Durability, Quorums. Version Stamps, Business and System Transactions, Version Stamps on Multiple Nodes **Assessment**: Data Visualization Assignment				
Module 3	**Key-Value Stores and Map-Reduce Framework**	**Design an algorithm using Example**		**Random Forest**	**No. of sessions:10**
	Topics: Map-Reduce, Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations, A Two Stage Map-Reduce Example, Incremental Map-Reduce Key-Value Databases, What Is a Key-Value Store, Key-Value Store Features, Consistency, Transactions, Query Features, Structure of Data, Scaling, Suitable Use Cases, Storing Session Information, User Profiles, Preference, Shopping Cart Data, When Not to Use, Relationships among Data, Multi operation Transactions, Query by Data, Operations by Sets. **Assessment**: Design an algorithm using Example, Random Forest				
Module 4	**Document-Oriented Databases and Use Cases**	**Case Study**		**Conduct a case study on how data sets can be gathered and implemented in real time application.**	**No. of sessions:10**
	Topics: Document Databases, What Is a Document Database?, Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Event Logging,				

	Content Management Systems, Blogging Platforms, Web Analytics or Real-Time Analytics, E- Commerce Applications, When Not to Use, Complex Transactions Spanning Different Operations, Queries against Varying Aggregate Structure **Assessment**: Case Study - Conduct a case study on how data sets can be gathered and implemented in real time application.				
Module 5	**Graph Databases and Connected Data Solutions**	**Case Study**			**No. of sessions:10**
	Topics: Graph Databases, What Is a Graph Database?, Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Connected Data, Routing, Dispatch, and Location-Based Services, Recommendation Engines, When Not to Use. **Assessment**: Case Study				

List of Laboratory Tasks:

Module 1: Introduction to NoSQL and Aggregate-Oriented Data Models

Exploring NoSQL Concepts

1. **Level 1:** Compare and contrast Relational Databases (SQL) and NoSQL databases. Discuss the limitations of SQL databases that led to the emergence of NoSQL.
2. **Level 2:** Analyze a real-world scenario (e.g., social media feed, e-commerce product catalog) and design an aggregate-oriented data model for it, highlighting the benefits over a relational model.

Key-Value Data Model

1. **Level 1:** Implement basic CRUD (Create, Read, Update, Delete) operations using a Key-Value store like Redis.
2. **Level 2:** Explore advanced features of Redis like data expiration, publish/subscribe, and transactions. Implement a simple caching mechanism using Redis.

Document Data Model

1. **Level 1:** Install and set up a Document database like MongoDB. Create a sample collection and insert documents with nested structures.
2. **Level 2:** Perform complex queries on MongoDB using operators like $match, $group, $aggregate. Design a schema for a sample application (e.g., a blog) in MongoDB.

Module 2: Distributed Data Systems and Consistency Models

Sharding and Replication

1. **Level 1:** Understand the concepts of sharding and replication. Discuss their advantages and disadvantages in the context of scalability and availability.

2. **Level 2:** Design a sharding strategy for a large dataset. Explore different sharding techniques (e.g., range-based, hash-based). Simulate data distribution across shards.

CAP Theorem

1. **Level 1:** Explain the CAP theorem and its implications for distributed systems. Discuss the trade-offs between Consistency, Availability, and Partition Tolerance.
2. **Level 2:** Analyze different NoSQL databases (e.g., Cassandra, MongoDB, Redis) in terms of their CAP theorem choices. Discuss use cases where different choices are appropriate.

Consistency Models

1. **Level 1:** Implement different consistency levels (e.g., eventual consistency, strong consistency) in a distributed Key-Value store. Observe the behavior of read and write operations under different consistency levels.
2. **Level 2:** Explore the concept of Quorums. Implement a read/write quorum system. Analyze the impact of quorum size on consistency and availability.

Module 3: Key-Value Stores and Map-Reduce Framework

Map-Reduce Basics

1. **Level 1:** Understand the basic principles of the Map-Reduce programming model.
2. **Level 2:** Implement a simple Map-Reduce job (e.g., word count) using a framework like Hadoop or Spark.

Advanced Map-Reduce

1. **Level 1:** Explore partitioning and combining techniques in Map-Reduce.
2. **Level 2:** Implement a multi-stage Map-Reduce job to process a large dataset (e.g., log file analysis).

Key-Value Store Use Cases

1. **Level 1:** Design a Key-Value store schema for storing session information in a web application.
2. **Level 2:** Implement a simple shopping cart functionality using a Key-Value store.

Module 4: Document-Oriented Databases and Use Cases

Document Database Queries

1. **Level 1:** Perform basic CRUD operations and simple queries on a Document database (e.g., MongoDB).
2. **Level 2:** Implement complex queries using aggregation pipelines in

MongoDB to analyze data.

Document Database Schema Design

1. **Level 1:** Design a schema for a content management system (CMS) using a Document database.
2. **Level 2:** Implement a real-time analytics dashboard using a Document database to store and query event logs.

E-commerce Application with Document Database

1. **Level 1:** Model product catalog data using a Document database.
2. **Level 2:** Implement features like product search and filtering using the Document database's query capabilities.

Module 5: Graph Databases and Connected Data Solutions

Graph Database Basics

1. **Level 1:** Understand the concepts of nodes, relationships, and properties in a graph database.
2. **Level 2:** Install and set up a graph database like Neo4j. Create a sample graph with nodes and relationships.

Graph Database Queries

1. **Level 1:** Learn Cypher query language for Neo4j. Perform basic graph traversals.
2. **Level 2:** Implement complex graph queries to find shortest paths, recommendations, and connected components.

Graph Database Use Cases

1. **Level 1:** Design a graph database schema for a social network.
2. **Level 2:** Implement a recommendation engine using a graph database to find similar users or products.

Textbook

T1 Sadalage, P. & Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Pearson Addision Wesley, 2012

This is a good resource for understanding the different types of NoSQL databases and their characteristics.

Reference book

R1 Dan Sullivan, "NoSQL For Mere Mortals", 1st Edition, Pearson Education India, 2015. (ISBN- 13: 978-9332557338)
R2 Dan McCreary and Ann Kelly, "Making Sense of NoSQL: A guide for Managers and the Rest of us", 1st Edition, Manning Publication/Dreamtech Press, 2013. (ISBN-13: 978-

	9351192022) **R3** Kristina Chodorow, "Mongodb: The Definitive Guide- Powerful and Scalable Data Storage", 2nd Edition, O'Reilly Publications, 2013. (ISBN-13: 978-9351102694) These books provide more in-depth coverage of specific NoSQL databases and their use cases. VIDEO LINKS: 1. https://www.geeksforgeeks.org/introduction-to-nosql/ (and related links in the page) 2. https://www.youtube.com/watch?v=0buKQHokLK8 (How do NoSQL databases work? Simply explained)

Course Code: CSE3414	Course Title: Applied Data Intelligence Type of Course: Program Core	L-T-P-C	2	0	2	3
Version No.		1.0				
Course Pre-requisites		Nil				
Anti-requisites		NIL				
Course Description		The aim of this course is to provide a complete overview of **Python's data analytics tools and techniques**. Recognizing Python as a crucial skill for many data science roles, this course helps students understand and develop **feature engineering** capabilities. Through a blended learning approach, students will acquire practical skills in Python for data science, alongside core concepts like **data wrangling**, **mathematical computing** with NumPy, and more.				
Course Objectives		The objective of the course is to familiarize the learners with the concepts of **Applied Data Science** and attain **Employability** through **Experiential Learning** techniques.				
Course Out Comes		On successful completion of this course, the students shall be able to: • **CO1**: Understand NumPy and perform essential matrix operations. [Knowledge] • **CO2**: Analyze the need for data preprocessing and apply various data visualization techniques. [Comprehensive] • **CO3**: Demonstrate the performance of different supervised learning algorithms like Decision Tree, Random Forest, Linear Regression, Logistic Regression, etc. [Application] • **CO4**: Apply unsupervised learning algorithms like K-Means, K-Medoids, etc., for grouping given data. [Application]				
Course Content:						
Module 1	**Introduction to Data Science, Python Data Structures, Python Numpy Package**	**Quiz**		**Knowledge based quiz**	**No. of sessions:8**	
	Topics: Data Science - Need, Applications, Difference between data analysis and data analytics. Python - Variables, data types, control structures, Operators, Simple operations, Array and its operations, NumPy operations, Matrix and its operations. Learning Activities: Interactive Python coding exercises, discussions on data science use cases. Assessment: Quiz: Knowledge-based quiz.					

Module 2	**Data preparation and preprocessing using Pandas dataframe, Exploratory Data Analysis, Data Visualization**	**Assignment**		**Data Visualization**	**No. of sessions:10**
	Topics: Dealing with missing values, Normalization, statistical description about the data, Accessing the data, Summary of the data, Relationship between the data, Data Visualization using Matplotlib and Seaborn. Learning Activities: Hands-on data wrangling, EDA, and visualization exercises using Pandas and Matplotlib. Assessment: Assignment: Data Visualization.				
Module 3	**Supervised Learning Algorithms**	**Design an algorithm using Example**		**Random Forest**	**No. of sessions:10**
	Topics: Decision Tree Algorithm (ID3 Classifier), Random Forest, Classifier Accuracy metrics (Confusion Matrix, Precision, Recall, F1-Score), Linear Prediction (Simple and Multiple Linear Regression), Logistic Regression – Case study. Learning Activities: Algorithm design discussions, implementation of supervised models, case study analysis. Assessment: Design an algorithm using Example (e.g., Random Forest).				
Module 4	**Unsupervised Learning Algorithms**	**Case Study**		**Conduct a case study on how data sets can be gathered and implemented in real time application.**	**No. of sessions:10**
	Topics: Various distance Functions (Euclidean, Manhattan), Dissimilarity between mixed types of data, K-Means Algorithm, K-Medoids Algorithm - Case Study. Learning Activities: Implementation of clustering algorithms, case study analysis on real-time applications. Assessment: Case Study: Conduct a case study on how datasets can be gathered and implemented in real-time application.				

List of Laboratory Tasks:

Module 1: Introduction to Data Science, Python Data Structures, Python NumPy Package

1. **Python Fundamentals & NumPy Array Operations**
    1. **Level 1:** Write Python programs to demonstrate basic data types, control flow structures (if/else, loops), and essential operations on Python lists and dictionaries. Create 1D and 2D NumPy arrays and perform basic array arithmetic (addition, subtraction, multiplication).
    2. **Level 2:** Implement advanced NumPy operations such as matrix multiplication, transpose, and slicing. Explain the concept of broadcasting in NumPy with an example.

Module 2: Data Preparation and Preprocessing using Pandas DataFrame, Exploratory

Data Analysis, Data Visualization

Pandas Data Handling & Initial Exploration

1. **Level 1:** Load a dataset (e.g., CSV file) into a Pandas DataFrame. Display basic information about the DataFrame (e.g., head(), info(), describe()) and select specific columns or rows.
2. **Level 2:** Identify and handle missing values using various strategies (e.g., imputation with mean/median/mode, dropping rows/columns). Perform data type conversions where necessary and explain the impact.

Data Preprocessing: Normalization & Standardization

1. **Level 1:** Apply Min-Max scaling (Normalization) to a numerical feature in your dataset. Visualize the distribution before and after scaling.
2. **Level 2:** Apply Z-score standardization to the same or different numerical features. Compare the effects of normalization and standardization on the data's distribution and range.

Exploratory Data Analysis (EDA) & Basic Visualization

1. **Level 1:** Calculate summary statistics for numerical features and frequency distributions for categorical features. Create basic bar plots and histograms using Matplotlib or Seaborn to visualize these distributions.
2. **Level 2:** Generate scatter plots to investigate relationships between two numerical variables. Create box plots to compare distributions across different categories, identifying potential outliers.

Advanced Data Visualization

1. **Level 1:** Use Seaborn to create a heatmap of the correlation matrix for numerical features in your dataset. Interpret the strength and direction of relationships.
2. **Level 2:** Generate a pair plot (Seaborn's pairplot) to visualize pairwise relationships and distributions for multiple variables in a single view. Discuss insights derived from these advanced visualizations.

Module 3: Supervised Learning Algorithms

Linear Regression

1. **Level 1:** Implement a Simple Linear Regression model using scikit-learn to predict a continuous target variable based on a single independent feature. Visualize the regression line.
2. **Level 2:** Extend to Multiple Linear Regression with several features. Evaluate the model's performance using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

Logistic Regression

1. **Level 1:** Implement a Logistic Regression model using scikit-learn for a binary classification problem. Make predictions and obtain class probabilities.
2. **Level 2:** Evaluate the Logistic Regression model using a confusion matrix, classification report (precision, recall, F1-score), and plot the Receiver Operating Characteristic (ROC) curve with AUC score.

Decision Trees

1. **Level 1:** Build a Decision Tree Classifier using scikit-learn on a classification dataset. Train the model and make predictions.
2. **Level 2:** Evaluate the Decision Tree's performance and visualize the tree structure. Experiment with pruning parameters (e.g., max_depth) to avoid overfitting.

Random Forest Classifier

1. **Level 1:** Implement a Random Forest Classifier using scikit-learn for a classification task. Compare its basic accuracy with the Decision Tree.
2. **Level 2:** Analyze the feature importances derived from the Random Forest model and discuss how ensemble methods can improve predictive performance.

Naïve Bayesian Classifier

1. **Level 1:** Implement a Naïve Bayes Classifier (e.g., Gaussian Naive Bayes or Multinomial Naive Bayes, depending on data type) using scikit-learn for a classification problem.
2. **Level 2:** Evaluate the Naïve Bayes model's performance and discuss its assumptions and suitability for different types of data.

Module 4: Unsupervised Learning Algorithms

K-Means Clustering

1. **Level 1:** Apply the K-Means clustering algorithm using scikit-learn to an unlabeled dataset. Visualize the clustered data points.
2. **Level 2:** Use the Elbow Method or Silhouette Score to determine the optimal number of clusters (K) for your dataset and justify your choice.

Dimensionality Reduction: Principal Component Analysis (PCA)

1. **Level 1:** Implement Principal Component Analysis (PCA) using scikit-learn to reduce the dimensionality of a dataset. Transform the data to the first two principal components.
2. **Level 2:** Visualize the data in the reduced PCA space and explain how PCA captures the most variance. Discuss the cumulative explained variance ratio.

	Dimensionality Reduction: Singular Value Decomposition (SVD) 1. **Level 1:** Apply Singular Value Decomposition (SVD) to a given matrix (e.g., from a dataset) using NumPy's linalg.svd function. 2. **Level 2:** Explain the components of SVD (U, S, V) and discuss its applications, particularly in dimensionality reduction and recommender systems. **Association Rule Mining (Conceptual & Basic Implementation)** 1. **Level 1:** Understand the concepts of Support, Confidence, and Lift in association rule mining. Discuss a real-world scenario (e.g., market basket analysis). 2. **Level 2:** Use a library like mlxtend to find frequent itemsets using the Apriori algorithm and generate association rules from a small transactional dataset. **Applying Unsupervised Learning for Insights** 1. **Level 1:** Select a real-world dataset (e.g., customer transaction data, image dataset) and apply a clustering algorithm (e.g., K-Means). 2. **Level 2:** Interpret the characteristics of the identified clusters or reduced dimensions, drawing meaningful insights or potential applications for the business/domain.
	Targeted Application & Tools that can be used: • IBM SPSS • Julia and Jupyter Notebook • Matplotlib
	Project work/Assignment:
	• Design Forest Fire and Wildfire Prediction System: Implement a system to predict forest fires or wildfires using historical weather data, terrain information, and fire incidents. Utilize appropriate supervised learning models learned in the course. • Driver Drowsiness Detection System with OpenCV & Keras: Develop a system that can detect driver drowsiness in real-time using computer vision techniques (e.g., eye aspect ratio, yawn detection) and a machine learning model. • Credit Card Fraud Detection using Python: Build a machine learning model to detect fraudulent credit card transactions from a given transactional dataset. Focus on handling imbalanced datasets and evaluating the model's performance using appropriate metrics for fraud detection.
	Textbook(s): **T1**: Alex Galea, Applied Data Science with Python and Jupyter, Packt Publishing, October 2018. **T2**: David Landup, Data Visualization in Python with Pandas and Matplotlib, Paperback, June 16, 2021.

	References: **R1**: Jesse Daniel, Data Science with Python and Dask, 1st Edition, July 30, 2019. **Weblinks:** • Udemy: https://www.udemy.com/course/applied-data-science-with-python-specialization-mhm/ • NPTEL online course: https://nptel.ac.in/courses/106106179 • https://presiuniv.knimbus.com/user#/home
	Topics relevant to "EMPLOYABILITY SKILLS": Data Science, Decision Tree Algorithm, and other machine learning algorithms learned throughout the course are central to developing Employability Skills through Experiential Learning techniques. The practical application of these concepts in assignments and projects directly contributes to this goal, as demonstrated by the assessment components mentioned in the course handout.

Course Code: CSE3515	**Course Title: Could Data Engineering** **Type of Course: Theory**	**L-T-P- C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	CSE2506 – Cloud Computing					
Anti-requisites	nil					
Course Description	This course is designed to introduce the concepts of Cloud Computing as a new computing paradigm. Cloud Computing has emerged in recent years as a new paradigm for hosting and delivering services over the Internet. Students will explore various Cloud Computing terminology, principles, and applications, gaining an understanding of the theoretical, technical, and commercial aspects of Cloud Computing. Topics include: Evolution of cloud computing and its services available today, Introduction, Architecture of cloud computing, Infrastructure, platform, software, Types of cloud, Business models, cloud services, Collaborating using cloud services, Virtualization for cloud, Security, Standards and Applications.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Cloud Computing and Virtualization and to enhance Employability through Participative Learning techniques.					
Course Out Comes	On successful completion of the course, students shall be able to: • **CO1:** Describe the fundamentals of cloud computing, virtualization, and various cloud computing services. (Familiarization, Understanding) • **CO2:** Discuss principles and technologies related to high-throughput and data-intensive computing in cloud environments. (Discussion, Analysis) • **CO3:** Explain the security challenges, standards, and best practices in cloud computing. (Explanation, Application) • **CO4:** Demonstrate the practical installation, configuration, and management of virtual machines and basic cloud services. (Application, Demonstration)					
Course Content:						

Module 1 : **Introduction to Cloud and Virtualization**		**10 Sessions**

- Cloud **Computing at a Glance:** Definition, characteristics, and historical developments.
- Building **Cloud Computing Environments:** Overview of essential components and infrastructure.
- Computing **Platforms and Technologies:** Evolution from traditional computing to cloud.
- Virtualization**:** Concepts, characteristics of virtualized environments, Taxonomy of Virtualization Techniques.
- Virtualization **and Cloud Computing:** Interplay and importance, Technology Examples.
- Cloud **Computing Architecture:** Introduction to IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) models.
- Types **of Clouds:** Public, Private, Hybrid, and Community clouds.
- Economics **of Cloud:** Cost benefits, pay-as-you-go model.

Module 2 : **High Throughput and Data Intensive Computing**:		**10 Sessions**

- Task **Computing:** Concepts of distributed task execution.
- MPI **Applications:** Introduction to Message Passing Interface for parallel computing.
- Task**-Based Programming:** Paradigms for distributed task execution.
- **Introduction to Data Intensive Computing (DIC):** Definition, challenges, and importance.
- **Technologies for DIC:** Overview of frameworks for large-scale data processing.
- **Aneka MapReduce Programming:** Introduction to MapReduce paradigm, its implementation details, and basic programming concepts in the context of Aneka or similar frameworks.

Module 3 : Cloud Security and Standards		**09 Sessions**

- **Cloud Security Challenges:** Data breaches, identity theft, compliance issues, shared responsibility model.
- **Software-as-a-Service Security:** Security considerations specific to SaaS applications.
- **Application Standards:** Open APIs, interoperability.
- **Client Standards:** Authentication, authorization mechanisms.
- **Infrastructure and Service Standards:** Compliance, governance, regulatory aspects in cloud.

Module 4 : Cloud Platforms, Advances in Cloud		**09 Sessions**

- **Introduction to Major Cloud Platforms:**

- Amazon Web Services (AWS): Overview of core services (EC2, S3, IAM).
- Google App Engine (GAE): Introduction to platform-as-a-service concepts.
- Microsoft Azure: Overview of key services and offerings.

- **Emerging Cloud Paradigms:**

- Media Clouds: Streaming, content delivery.
- Security Clouds: Cloud-based security services.
- Computing Clouds: High-performance computing in the cloud.
- Mobile Clouds: Cloud services for mobile applications.

- Federated Clouds: Interoperation between different cloud providers.
- Hybrid Cloud: Integrating on-premise infrastructure with public clouds.

Text Book

- John Rittinghouse and James Ransome, "Cloud Computing, Implementation, Management and Security", CRC Press.
- Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "Mastering Cloud

	Computing", McGraw Hill Education.
	References • David E.Y. Sarna, "Implementing and Developing Cloud Applications", CRC Press. • Anthony T Velte, Toby J Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach", Tata McGraw-Hill. **Web resources :** • https://presiuniv.knimbus.com/user#/home • Official documentation for AWS, Google Cloud, and Microsoft Azure.
	Skill Development This course aims to develop the following employability skills through participative learning techniques and problem-solving methodologies, which will be assessed through various components mentioned in the course handout: • **AWS & Azure:** Practical understanding and basic operations on leading cloud platforms. • **APIs:** Familiarity with cloud service APIs for programmatic interaction. • **Aneka Cloud Platform:** Conceptual understanding of task-based and MapReduce programming models. • **EC2:** Hands-on experience with Elastic Compute Cloud instances. • **Installation of VM Workstation:** Practical skills in setting up and managing virtual machines. • **Infrastructure Security Challenges:** Understanding and mitigating security risks in cloud infrastructure. • **Convex Optimization Models and Methods:** For problem-solving methodologies in resource allocation and efficiency (conceptual introduction for skill development).

<table>
<tr><td>Course Code:
CSE3516</td><td colspan="3">Course Title: Federated Learning
Type of Course: Theory Only Course</td><td>L- T-P-C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">CSE2506 – Cloud Computing</td></tr>
<tr><td>Anti-requisites</td><td colspan="8"></td></tr>
<tr><td>Course Description</td><td colspan="8">This course introduces Federated Learning, a new paradigm that improves model performance by leveraging data diversity across distributed devices while preserving privacy. Students will learn the foundational concepts of Federated Learning, its architecture, and various algorithms. The course emphasizes the practical application of real-time model updates in diverse scenarios and covers critical aspects of privacy, security, and scalability in machine learning. Given the students' prior exposure to AI and ML, this elective delves into the distributed nature of model training.</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of this course is to expose learners to the necessity of distributed model updates, understand the importance of privacy and security in machine learning techniques, and equip them with the skills to apply and evaluate federated learning systems.</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of the course the students shall be able to:

On successful completion of the course, students shall be able to:
CO1: Describe the key concepts, architecture, and terminology of Federated Learning. (Knowledge)
CO2: Apply different methods and algorithms to develop horizontal and vertical federated learning systems. (Comprehension, Application)
CO3: Apply optimization techniques specific to Federated Learning, especially for handling non-IID data. (Application)
CO4: Construct and scale a simple federated learning system using relevant tools and platforms. (Application)
CO5: Evaluate privacy and security concerns in Federated Learning and implement privacy-preserving techniques. (Application, Evaluation)</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction to Federated Learning:</td><td>Assignment</td><td></td><td colspan="3">Data Collection/Interpretation</td><td colspan="2">10 Sessions</td></tr>
<tr><td></td><td colspan="8">Overview of Federated Learning: Definition, historical context, and diverse applications (e.g., mobile devices, IoT, healthcare).
Concepts and Terminology: Central server, clients (edge devices), local models, global model, rounds of communication.
Federated Learning Architecture: Client-server model, peer-to-peer federated learning (brief overview).
Machine Learning Perspective: Recap of distributed machine learning, challenges of centralized data collection (privacy, bandwidth, legal).</td></tr>
</table>

- **Security & Privacy in Federated Learning:** Introduction to data leakage, inference attacks, and the need for privacy-preserving mechanisms.
- **Federated Learning vs. Centralized Learning:** Comparison and contrast of methodologies, advantages, and disadvantages.

Module 2	**Horizontal and Vertical Federated Learning**	Case studies / Case let		Case studies / Case let	**13 Sessions**

- **Horizontal Federated Learning (HFL):**
  - Definition and Architecture of Horizontal Federated Learning (feature space alignment, different samples).
  - **Federated Averaging (FedAvg) Algorithm:** Detailed explanation of the algorithm, aggregation process, and convergence.
  - **Improvements on the FedAvg Algorithm:** FedProx, SCAFFOLD, adaptive optimization (conceptual overview).
- **Vertical Federated Learning (VFL):**
  - Definition and Architecture of Vertical Federated Learning (sample space alignment, different features).
  - **VFL Algorithms:** Secure Federated Linear Regression, Secure Federated Tree Boosting (conceptual understanding and practical challenges).
- **Federated Learning with Non-IID Data:**
  - **Heterogeneity in Federated Learning:** Data distribution skew, statistical heterogeneity, system heterogeneity.
  - **Stratification and Local Update Rules:** Strategies to mitigate performance degradation with non-IID data.
  - **Advanced Optimization Techniques in Federated Learning:** Adaptive Learning Rate, Momentum, and Weight Decay in a federated context.

Module 3	**Federated Transfer Learning and Security**	Case studies / Case let		Case studies / Case let	**14 Sessions**

- **Federated Transfer Learning (FTL):**
  - Framework of Federated Transfer Learning: Leveraging pre-trained models in a federated setting.
  - Homomorphic Encryption in FTL: Introduction to its role in privacy-preserving computations.
  - FTL Training Process and Prediction Process: Workflow and benefits.
  - Security Analysis of FTL: Understanding potential vulnerabilities.
  - Secret Sharing based FTL: An alternative privacy-preserving technique.

	• **Security in Federated Learning:** • **Protecting Against Data Leakage in FL:** Overview of common attacks (model inversion, membership inference). • **Private Parameter Aggregation for FL:** Secure multi-party computation (SMC) concepts. • **Data Leakage in FL Advanced Security Issues:** Sybil attacks, poisoning attacks. • **Dealing with Byzantine Threats to Neural Networks in FL:** Robust aggregation methods to handle malicious clients.
	Targeted Application & Tools that can be used: Building and simulating federated learning systems using tensorFlowFederated (TFF), PySyft, Google Colab / Jupyter Notebook. Simulating and deploying FL across nodes using Cloud Platforms (AWS, GCP, Azure) It can help small, medium and large businesses in any sector keep information assets secure.
	Project work/Assignment:
	Assignment: Practical Applications and Case Studies -Real-world Applications of Federated Learning
	Text Book • Heiko Ludwig (Editor), Nathalie Baracaldo, "Federated learning comprehensive overview of methods and applications", Springer Nature Switzerland AG; 1st ed. 2022 edition. • Ronald J. Brachman, Francessa Rossi, and Peter Stone (Series Editors), "Federated Learning (Synthesis Lectures on Artificial Intelligence and Machine Learning)", Morgan & Claypool Publishers, 2019.
	References • Kiyoshi Nakayama PhD, George Jeno, "Federated Learning with Python", O'Reilly Media, Inc. • Emily Glanz, Nova Fallen, "What-is-federated learning?", O'Reilly Media, Inc. • **E-book Link R1:** http://www.iso.org/iso/home/standards/management-standards/iso27001.html (Relevant for security standards) • **E-book Link R2:** https://www.oreilly.com/library/view/what-is-federated/9781098107253/ch03.html • Official documentation for TensorFlow Federated (TFF) • Official documentation for PySyft

	• Official documentation for AWS, Google Cloud Platform (GCP), and Microsoft Azure related to ML services.
	This course aims to develop the following employability skills through participative learning techniques and problem-solving methodologies, which will be assessed through various components mentioned in the course handout: • **Building and Simulating Federated Learning Systems:** Practical experience with TensorFlow Federated (TFF) and PySyft. • **Cloud Platform Integration:** Simulating and potentially deploying FL across nodes using Cloud Platforms (AWS, GCP, Azure). • **Privacy-Preserving Techniques:** Understanding and conceptualizing the implementation of Homomorphic Encryption and Differential Privacy. • **Security Policy Implementation & Roles:** Understanding and analyzing security policies relevant to distributed ML and data governance. • **Problem Solving:** Applying FL concepts to real-world scenarios and addressing challenges like non-IID data and adversarial attacks.

Course Code: CSE3517	**Course Title: Edge Computing** **Type of Course: Theory Only Course Discipline Elective**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2506 – Cloud Computing					
Anti-requisites	**Nil**					
Course Description	This course provides a comprehensive study of Edge Computing, building upon the foundational concepts of distributed systems and cloud computing. It will cover significant tools, applications, and platforms that comprise today's edge computing landscape, with a special focus on its intersection with IoT and big data. Students will explore the evolution of the computing industry leading to edge, understand various types of edge deployments (e.g., CDN Edge, IoT Edge, Multi-access Edge (MEC)), and delve into vendor platforms, software services, standard bodies, and open-source communities relevant to edge computing. A research project component will allow students to apply their knowledge to real-world scenarios.					
Course Objective	The objective of this course is to familiarize the learners with the concepts of Edge Computing and to enhance Employability through Problem Solving Methodologies, enabling them to design, implement, and manage edge-based solutions.					
Course Out Comes	On successful completion of the course, students shall be able to: • **CO1:** Understand the fundamental principles, architectures, and use cases of edge computing. (Knowledge) • **CO2:** Describe IoT Architectures, core IoT modules, and their relationship with edge computing. (Comprehension) • **CO3:** Summarize and apply various edge-to-cloud communication protocols for data exchange. (Comprehension, Application) • **CO4:** Demonstrate practical skills in implementing edge computing solutions using single-board computers like Raspberry Pi. (Comprehension, Application)					
Course Content:						

Module 1	IoT and Edge Computing Definition and Use Cases	Term paper/Assignment/Case Study	Programming/Simulation/Data Collection/any other such associated activity		**9 Sessions**

- **Introduction to Edge Computing:** Scenarios and Use cases (e.g., smart cities, industrial automation, healthcare).
- **Edge Computing Purpose and Definition:** Why edge computing, its benefits (latency, bandwidth, privacy), and core characteristics.
- **Edge Computing Hardware Architectures:** Types of edge devices, gateways, and compute nodes.
- **Edge Platforms:** Overview of common edge computing platforms.
- **Edge vs. Fog Computing:** Comparison and contrast of concepts, roles, and applications.
- **Communication Models:** Edge, Fog, and Machine-to-Machine (M2M) communication.

Module 2	IoT Architecture and Core IoT Modules	Term paper/Assignment/**Case Study**	Programming/Simulation/Data Collection/any other such associated activity		**9 Sessions**

- **A Connected Ecosystem:** Understanding the broader IoT landscape.
- **IoT vs. Machine-to-Machine (M2M) vs. SCADA:** Differentiating concepts and their evolution.
- **The Value of a Network:** Metcalfe's and Beckstrom's laws applied to IoT.
- **IoT and Edge Architecture:** Layered architectures (perception, network, service, application), role of an architect.
- **Understanding Implementations with Examples:**
  - Example use case and deployment.
  - Case study – Telemedicine palliative care: Requirements, Implementation, Use case retrospective.

.

Module 3	RaspberryPi	Term paper/Assignment/Case Study	**Programming/Simulation**/Data Collection/any other such associated activity		**10 Sessions**

- **Introduction to Raspberry Pi:** History, models, and capabilities for edge computing.
- **About the Raspberry Pi Board:** Hardware Layout, Pinouts (GPIO).
- **Operating Systems on Raspberry Pi:** Raspberry Pi OS (formerly Raspbian), other options.
- **Configuring Raspberry Pi:** Initial setup, network configuration, headless setup.
- **Programming Raspberry Pi:** Introduction to Python for GPIO control.
- **Connecting Raspberry Pi via SSH:** Remote access methods and tools.
- **Interfacing DHT Sensor with Pi:** Practical example of sensor integration.
- **Pi as Webserver:** Setting up a simple web server (e.g., Flask or Apache).
- **Pi Camera:** Introduction to capturing images and video.
- **Image & Video Processing using Pi:** Basic concepts and libraries (e.g., OpenCV for simple tasks).

Module 4	Edge to Cloud Protocols	Term paper/Assignment/Case Study	Programming/Simulation/Data Collection/any other such associated activity		**7 Sessions**

- **Edge to Cloud Protocols:** Overview of common protocols for IoT and edge communication.

- **MQTT (Message Queuing Telemetry Transport):**
  - Introduction to MQTT: Lightweight publish-subscribe protocol.
  - MQTT Publish-Subscribe Model: Brokers, topics, publishers, subscribers.
  - MQTT Architecture Details: Components and their interactions.
  - MQTT State Transitions, Packet Structure, Data Types.
  - MQTT Communication Formats: QoS levels.
  - MQTT 3.1.1 Working Example.

Module 5	Edge computing with RaspberryPi	Term paper/Assignment/Case Study	**Programming/Simulation**/Data Collection/any other such associated activity	**7 Sessions**

- **Edge Computing with Raspberry Pi:** Advanced examples and project ideas.
- **Industrial and Commercial IoT and Edge:** Use cases and solutions in various sectors.
- **Edge Computing Solutions:** Overview of enterprise-grade edge solutions.
- **Targeted Application:** Smart Surveillance Video Stream Processing at the Edge for Real-Time Human Objects Tracking (Conceptual discussion).
- **Tools:** Eclipse ioFog (conceptual understanding): An integrated development environment for open-source edge computing platforms.

Targeted Application & Tools that can be used:

Application : Smart Surveillance Video Stream Processing at the Edge for Real-Time Human Objects Tracking.

Tools :Eclipse ioFog : An integrated development environment built by the Eclipse Foundation, backed by IBM. Eclipse ioFog is the organization's open-source edge computing platform.

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

Exploring topics such as developing scalable architectures, moving from closed systems to open systems, and ethical issues rising from data sensing, addresses both the challenges and opportunities of Edge computing presents. Students can harness federating Edge resources, middleware design issues, data management and predictive analysis, smart transportation and surveillance applications, and more. A coordinated and integrated solutions can be provided by thorough knowledge of the foundations, applications, and issues that are central to Edge computing.

ext Books:

1. Perry Lea, "IoT and Edge Computing for Architects - Second Edition", Packt Publishing, 2020. ISBN: 9781839214806.
2. Simon Monk, "Raspberry Pi Cookbook, 3rd Edition", O'Reilly Media, Inc., 2019. ISBN: 978149204322.

References:

	• Official documentation for Raspberry Pi. • Official documentation for MQTT. • Documentation for Eclipse ioFog and other open-source edge platforms.
	Topics relevant to "EMPLOYABILITY SKILLS": This course aims to develop the following employability skills through participative learning techniques and problem-solving methodologies, which will be assessed through various components mentioned in the course handout: • **Implementation of Microcomputer Raspberry Pi:** Practical skills in setting up, configuring, and programming Raspberry Pi for edge tasks. • **Device Interfacing:** Hands-on experience with connecting and reading data from various sensors and peripherals. • **Network Protocol Understanding:** In-depth knowledge and application of IoT communication protocols like MQTT. • **Edge Application Development:** Designing and implementing simple edge computing applications. • **Problem Solving:** Applying knowledge to develop scalable architectures, move from closed to open systems, and address ethical issues related to data sensing in edge environments. • **System Integration:** Harnessing federating edge resources, understanding middleware design issues, data management, and predictive analysis at the edge. • **Application-Specific Design:** Developing solutions for smart transportation, surveillance, and other real-world edge computing scenarios.

Course Code: CSE3518	**Course Title: Network Security and Firewall Management** **Type of Course: Integrated**		**L-T- P- C**	2	0	2	3
Version No.	1						
Course Pre-requisites	CSE2503 – Cryptography and Network Security						
Anti-requisites							
Course Description	This course provides an in-depth study of various network attack techniques and methods to defend against them. A number of threats and vulnerabilities of the Internet will be covered, including various vulnerabilities of TCP/IP protocols, denial of service (DoS), attacks on routing, attacks on DNS servers, TCP session hijacking, and so on. This course will also cover defending mechanisms, including intrusion detection, firewalls, tracing the source of attacks, anonymous communication, IPsec, virtual private networks (VPN), and Public Key Infrastructure (PKI). To make it easy for students to understand these attacks, basics of the TCP/IP protocols will also be covered in the course.						
Course Objective	The objective of the course is to familiarize the learners with the concepts of Firewall and Internet security and to enhance Skill Development through Problem Solving Methodologies.						
Course Out Comes	On successful completion of the course, students shall be able to: • **CO1:** Identify elements of firewall design, categorize types of security threats, and formulate responses to security attacks. (Knowledge, Analysis) • **CO2:** Examine security incident post-mortem reporting and ongoing network security activities. (Analysis, Evaluation) • **CO3:** Construct code for fundamental authentication algorithms. (Application, Construction) • **CO4:** Develop a signature scheme using the Digital Signature Standard (DSS). (Application, Development) • **CO5:** Demonstrate the implementation of network security systems using open-source tools. (Application, Demonstration)						
Course Content:							

Module 1	Introduction to Firewall	Assignment		Data Collection/Interpretation	**12 Sessions**

- **Introduction of Firewall in Computer Networks:** Definition, purpose, and importance.
- **Categories of Firewall:** Packet-filtering, stateful inspection, application-level gateways (proxies).
- **How Firewall Works:** Rule sets, packet inspection.
- **Types of Firewall:** Hardware vs. Software, Personal vs. Network.
- **Firewall Location and Configuration:** DMZ, screening router, dual-homed host.
- **Firewall Policies:** Best practices, common policy types.
- **Firewall Biasing:** Understanding different security postures.
- **Network Architecture:** Segmentation, zones.

- **Net masks:** Role in network security.
- **Packet Filters:** Basic functionality and limitations.
- **Stateful Firewalls:** Advantages over stateless.
- **Resources:** Overview of firewall resources.

Module 2	Computer security	Case studies / Case let		Case studies / Case let	12 Sessions

- **Attacks on Computers and Computer Security:**
  - **Need for Security:** Confidentiality, Integrity, Availability (CIA Triad).
  - **Security Approaches:** Layered security, defense in depth.
  - **Principles of Security:** Least privilege, separation of duties.
  - **Types of Attacks:** Passive vs. Active, Insider vs. Outsider, Malware (viruses, worms, trojans, ransomware).
- **Transport Level Security:**
  - **Web Security Considerations:** Common vulnerabilities.
  - **Secure Sockets Layer (SSL) / Transport Layer Security (TLS):** Handshake protocol, record protocol, cryptographic parameters.
  - **HTTPS:** How it uses SSL/TLS.
  - **Secure Shell (SSH):** Remote secure access.

Module 3	Network Security	Quiz		Case studies / Case let	10 Sessions

- **Overview of Network Security:**
  - **Elements of Network Security:** Authentication, Authorization, Accounting (AAA).
  - **Classification of Network Attacks:** DoS, DDoS, IP Spoofing, ARP Spoofing, DNS poisoning, Session Hijacking.
  - **Security Methods:** Cryptography, Firewalls, IDS/IPS.
- **Symmetric-Key Cryptography:**
  - **Data Encryption Standard (DES):** Principles, strengths, weaknesses.
  - **Advanced Encryption Standard (AES):** Principles, modes of operation.
- **Public-Key Cryptography:**
  - **RSA Algorithm:** Key generation, encryption, decryption.
  - **Diffie-Hellman Key-Exchange Protocol:** Secure key establishment.
- **Authentication:**
  - **Hash Function:** Properties, Message Authentication Code (MAC).
  - **Secure Hash Algorithm (SHA):** SHA-1, SHA-256.

- **Digital Signatures:** Creation, verification, non-repudiation.

Module 4	Cyber laws and Compliance Standards	Quiz		Case studies / Case let	**11 Sessions**

- **Kerberos:** Working mechanism, Authentication Server (AS), Ticket Granting Server (TGS), Service Server (SS).

- **Internet Security Protocols:**

    - **IPsec:** Architecture, Authentication Header (AH), Encapsulating Security Payload (ESP).
    - **IPsec Modes:** Transport and Tunnel modes.

- **Email Security:** PGP (Pretty Good Privacy), S/MIME.

- **Public Key Infrastructure (PKI):**

    - **Certificates:** X.509 format.
    - **Certificate Authorities (CA):** Role in trust.

- **Cyber Crime:**

    - **Introduction:** Definition, types.
    - **Common Cybercrimes:** Hacking, Digital Forgery, Cyber Stalking, Identity Theft and Fraud, Cyber Terrorism, Cyber Defamation.
    - **Crimes against:** Individual, Government, Property.

List of Laboratory Tasks:

1. **Experiment 1: Basic Cryptographic Substitution Techniques**
2. **Low-Level:** Implement Caesar Cipher and Playfair Cipher for encryption and decryption using a programming language (e.g., Python).
3. **High-Level:** Implement Hill Cipher and Vigenere Cipher. Test them with various keys and analyze their strengths and weaknesses.
4. **Experiment 2: Transposition Techniques**
5. **Low-Level:** Implement Rail Fence cipher for encryption and decryption.
6. **High-Level:** Implement Row & Column Transformation cipher for encryption and decryption. Compare the security of transposition techniques with substitution techniques.
7. **Experiment 3: Data Encryption Standard (DES) Implementation**
8. **Low-Level:** Understand the core concepts of DES. Implement a simplified version of DES (e.g., Feistel structure with fewer rounds or smaller block size) to illustrate the process.
9. **High-Level:** Apply a DES algorithm library (e.g., PyCryptodome in Python) to encrypt and decrypt a practical file (e.g., a text document). Discuss the concept of modes of operation (e.g., ECB, CBC).
10. **Experiment 4: Advanced Encryption Standard (AES) Implementation**
11. **Low-Level:** Understand the core concepts of AES. Use an AES algorithm library (e.g., PyCryptodome) to encrypt and decrypt a small text string using different key sizes (e.g., 128-bit).

12. **High-Level:** Apply AES algorithm with a specific mode of operation (e.g., CBC or GCM) for practical applications such as encrypting and decrypting a larger file securely. Discuss initialization vectors (IVs).
13. **Experiment 5: RSA Algorithm Implementation**
14. **Low-Level:** Implement the core steps of RSA algorithm (key generation, encryption, decryption) using a programming language (e.g., Python), focusing on small prime numbers for demonstration.
15. **High-Level:** Implement RSA Algorithm using HTML and JavaScript for a basic client-side encryption/decryption demonstration, showcasing public/private key pairs. *Note: For security, full-scale crypto should be server-side.*
16. **Experiment 6: Diffie-Hellman Key Exchange**
17. **Low-Level:** Implement the Diffie-Hellman Key Exchange algorithm to establish a shared secret key between two parties using Python.
18. **High-Level:** Demonstrate how this shared secret can then be used to encrypt subsequent communication using a symmetric cipher. Discuss man-in-the-middle attack concepts.
19. **Experiment 7: Secure Hash Algorithm (SHA-1/SHA-256)**
20. **Low-Level:** Calculate the message digest (hash) of a given text using the SHA-1 algorithm in Python.
21. **High-Level:** Compare SHA-1 with SHA-256. Calculate the message digest of a file (not just text) using SHA-256 and demonstrate the impact of even a single-bit change on the hash value.
22. **Experiment 8: Digital Signature Standard (DSS)**
23. **Low-Level:** Understand the concept of Digital Signatures and their importance (authentication, integrity, non-repudiation).
24. **High-Level:** Implement a simplified SIGNATURE SCHEME using the principles of Digital Signature Standard (or a similar ECDSA/RSA-based digital signature process) for a given message. Demonstrate signature creation and verification.
25. **Experiment 9: Intrusion Detection System (IDS) Demonstration**
26. **Low-Level:** Install and configure Snort (or a similar open-source IDS) on a Linux VM. Configure a simple rule to detect basic network activities (e.g., ICMP packets).
27. **High-Level:** Generate various types of network traffic (e.g., port scans using Nmap, simple DoS attempts using Hping3) in the lab environment and observe how Snort detects and logs these events. Analyze Snort alerts.
28. **Experiment 10: Vulnerability Assessment using Tools**
29. **Low-Level:** Use a simple network scanner like Nmap to scan a target VM in the lab environment and identify open ports and running services.
30. **High-Level:** Explore a vulnerability assessment tool (e.g., OpenVAS/Greenbone Security Assistant, or understand the concepts of N-Stalker if a free version is unavailable) to scan a target system for known vulnerabilities and generate a basic report.
31. **Experiment 11: Understanding Malware: Building Trojans (Conceptual/Safe Simulation)**
32. **Low-Level:** Understand the *concept* of a Trojan Horse and its delivery mechanisms. (NO ACTUAL MALWARE CREATION). Discuss safe ways to analyze malware binaries (e.g., in a sandboxed environment).
33. **High-Level:** (Safe Simulation): Simulate a "Trojan-like" behavior using a legitimate script (e.g., a Python script that appears to do one thing but also performs a hidden, harmless action like creating a file in a different directory). Discuss prevention methods.
34. **Experiment 12: Rootkit Hunter**
35. **Low-Level:** Install and run Rootkit Hunter (rkhunter) on a Linux VM. Understand its basic functionality and what it checks for.
36. **High-Level:** Interpret the scan results from rkhunter. Discuss how rootkits operate and why tools like rkhunter are essential for detecting them. (No actual rootkit installation).
37. **Experiment 13: Firewall Configuration and Policy Enforcement**

38. **Low-Level:** Configure a host-based firewall (e.g., ufw on Ubuntu or Windows Firewall) to allow specific inbound/outbound traffic (e.g., SSH, HTTP) and block others.
39. **High-Level:** Simulate a network with two VMs and a "firewall" VM between them. Configure packet filtering rules on the firewall to control traffic flow based on IP addresses, ports, and protocols. Test the rules.
40. **Experiment 14: Network Traffic Analysis with Wireshark**
41. **Low-Level:** Capture network traffic using Wireshark on your host machine or a VM. Filter packets based on protocol (e.g., TCP, UDP, ICMP) and source/destination IP addresses.
42. **High-Level:** Analyze captured HTTP, FTP, or Telnet traffic to identify unencrypted credentials or sensitive information. Discuss the importance of encryption for network security.
43. **Experiment 15: Implementing a Simple Virtual Private Network (VPN) Concept**
44. **Low-Level:** Understand the basic concept of a VPN and its purpose. Configure a simple SSH tunnel between two Linux VMs to securely forward traffic.
45. **High-Level:** (Conceptual) Explore the configuration steps for a simple OpenVPN server and client setup in a lab environment. Discuss the role of public key infrastructure in VPNs for authentication.

Targeted Application & Tools that can be used :

- **Application:** Designing and implementing secure network architectures, protecting against various cyber threats, and managing network security policies.
- **Tools:**
  - **Snort:** For intrusion detection and packet analysis.
  - **N-Stalker:** A vulnerability assessment tool (for exploring concepts).
  - **Wireshark:** For network protocol analysis.
  - **OpenSSL:** For cryptographic operations and certificate management.
  - **Netcat/Nmap:** For basic network reconnaissance and port scanning.
  - **Virtualization Software:** (e.g., VirtualBox, VMware) for creating isolated lab environments.

Project Work/Assignment :

Students will undertake a project focusing on a specific aspect of network security or firewall management. This could involve:

- **Design and Implementation of a Secure Network Segment:** Proposing and configuring a secure network architecture for a small organization, including firewall rules, network segmentation, and basic intrusion detection.
- **Vulnerability Assessment and Penetration Testing (Ethical Hacking):** Using open-source tools to identify vulnerabilities in a controlled lab environment and propose mitigation strategies.
- **Development of a Cryptographic Application:** Building a small application that uses an encryption algorithm (e.g., AES, RSA) or a digital signature scheme to secure data or communication.
- **Incident Response Simulation:** Analyzing a simulated network attack scenario,

conducting a post-mortem analysis, and recommending improvements to security posture.
- **Firewall Rule Optimization:** Analyzing existing firewall rule sets and proposing optimizations for performance and security.

Text Book

T1 : Behrouz A Forouzan, Data and Communications and Networking, Fifth Edition, McGraw Hill, Indian Edition

T2: James F Kurose and Keith W Ross, Computer Networking, A Top-Down Approach, Sixth edition, Pearson,2017

References

R1: Andrew S Tanenbaum, Computer Networks, fifth edition, Pearson Edition

R2: Nader F Mir, Computer and Communication Networks, 2nd Edition, Pearson, 2014.

Web resources:
1. https://networklessons.com/cisco/asa-firewall
2. https://www.udemy.com/course/cisco-asa-firewall-lab-guide
3. https://geekflare.com/learn-network-security

Skill Development

This course aims to develop the following employability skills through problem-solving methodologies and practical lab exercises, which will be assessed through various components mentioned in the course handout:

- **Network Security Implementation:** Practical skills in configuring and managing network security devices and software (e.g., firewalls, IDS/IPS).
- **Cryptographic Algorithm Implementation:** Ability to implement fundamental encryption, decryption, hashing, and digital signature algorithms.
- **Threat Identification and Analysis:** Skills in recognizing various network attack techniques and analyzing their impact.
- **Vulnerability Assessment:** Practical experience with tools and methodologies for identifying security weaknesses.
- **Incident Response:** Understanding the process of responding to and reporting on security incidents.
- **Problem Solving:** Applying knowledge to complex network security challenges, including design, defense, and analysis.
- **Cyber Law Awareness:** Understanding legal and ethical implications of cybersecurity activities.

Course Code: CSE3519	Course Title: Information Security and Management **Type of Course: Theory Only Course**	L- T-P-C	3	0	0	3
Course Pre-requisites	CSE2503 – Cryptography and Network Security					
Anti-requisites						
Course Description	This course explores **information security** through introductory material, helping students gain an appreciation of the scope and context of information security. It includes a brief introduction to **cryptography**, **security management**, **network,** and **computer security**. This course allows students to begin a fascinating journey into the study of information security and develop an appreciation of some key security concepts. The course concludes with a discussion of a simple model of information security in industry and explores the skills, knowledge, and roles required for **employability**. Students will be able to determine and analyze potential career opportunities in this profession.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Information Security and Management** and attain Employability through Participative Learning techniques.					
Course Out Comes	On successful completion of the course, students shall be able to: • **CO1:** Describe the basic concept of **information security**, including its foundational principles and common threats. (Knowledge) • **CO2:** Explain the core concepts and methods of **cryptography** and its role in securing information. (Comprehension) • **CO3:** Demonstrate understanding and application of **risk management** aspects in information security. (Application) • **CO4:** Analyze **data leakage threats** and propose strategies for reducing the risk of data loss. (Analysis) • **CO5:** Understand and apply principles of **information security policies** and **management frameworks** (e.g., ISO 27001). (Application)					
Course Content:						
Module 1	**Information Security Management:**	Assignment		Data Collection/Interpretation	**10 Sessions**	
	• **Information Security Overview:** Definition, importance, and objectives (Confidentiality, Integrity, Availability - CIA Triad). • **Threat and Attack Vectors:** Understanding sources and methods of attacks (e.g., human error, malware, social engineering, physical threats). • **Types of Attacks:** Passive vs. Active attacks, Insider vs. Outsider threats. • **Common Vulnerabilities and Exposure (CVE):** Introduction to CVE database and its role. • **Security Attacks:** Overview of common attack types (e.g., DoS, phishing, viruses, worms, ransomware).					

<table>
<tr><td></td><td colspan="6">

- **Fundamentals of Information Security:** Key principles and best practices.
- **Computer Security Concerns:** Operating system security, application security.
- **Information Security Measures:** Technical, administrative, and physical controls.

</td></tr>
<tr><td>Module 2</td><td>Fundamentals of Information Security and Data Leakage</td><td>Case studies / Case let</td><td></td><td>Case studies / Case let</td><td>13 Sessions</td></tr>
<tr><td></td><td colspan="6">

- **Key Elements of Networks:** Network components and their security implications.
- **Logical Elements of Networks:** Network segmentation, zones, protocols from a security perspective.
- **Critical Information Characteristics:** Identifying sensitive data, data classification.
- **Information States:** Data at Rest, Data in Transit, Data in Use, and their respective security considerations.
- **What is Data Leakage and Statistics:** Definition, prevalence, and impact of data breaches.
- **Data Leakage Threats:** Internal and external sources of data leakage.
- **Reducing the Risk of Data Loss:** Data Loss Prevention (DLP) strategies, encryption, access controls.
- **Key Performance Indicators (KPI) in Security:** Metrics for measuring security effectiveness.
- **Database Security:** Common vulnerabilities, access controls, encryption for databases.

</td></tr>
<tr><td>Module 3</td><td>Information Security Policies and Management</td><td>Case studies / Case let</td><td></td><td>Case studies / Case let</td><td>14 Sessions</td></tr>
<tr><td></td><td colspan="6">

- **Information Security Policies:** Necessity, purpose, and types (e.g., acceptable use, password policy).

- **Key Elements and Characteristics of Security Policies:** Clarity, enforceability, comprehensiveness.
- **Security Policy Implementation:** Development, communication, and enforcement.
- **Configuration Management:** Secure configurations, baseline security.
- **Security Standards, Guidelines, and Frameworks:** Introduction to ISO 27000 series, NIST, COBIT.
- **Security Roles and Responsibilities:** Defining roles (e.g., CISO, Security Analyst, Data Owner).
- **Accountability:** Establishing clear lines of responsibility.
- **Roles and Responsibilities of Information Security Management Team:** Incident response team, security operations.
- **Responding to Emergency Situations:** Incident response lifecycle.
- **Risk Analysis Process:** Identification, analysis, evaluation, and treatment of risks.

.

</td></tr>
<tr><td></td><td colspan="6">Targeted Application & Tools that can be used:</td></tr>
</table>

- **Application:** Implementation and management of an **Information Security Management System (ISMS)**, which is a systematic approach to managing sensitive company information to ensure its security. This includes integrating people, processes, and IT systems by applying a **risk management process**. An ISMS helps organizations of all sizes and sectors keep information assets secure, including financial information, intellectual property, employee details, or information entrusted by third parties.

- **Tools:**

  - **ISO 27000 family of standards:** Conceptual framework for ISMS.
  - **ISO/IEC 27001:** The best-known standard in the family, providing requirements for an ISMS.
  - **Security Information and Event Management (SIEM) tools (conceptual):** For centralized logging and security event correlation.
  - **Vulnerability Scanners (conceptual):** For identifying system weaknesses.
  - **Data Loss Prevention (DLP) solutions (conceptual):** For preventing sensitive data from leaving the organization.

Project work/Assignment: Students will undertake a project or assignment that allows them to apply the concepts learned in the course to a practical scenario. This can involve:

- **Developing a Mini Information Security Policy Document:** Drafting essential security policies (e.g., Password Policy, Acceptable Use Policy) for a hypothetical organization, adhering to standard policy characteristics.
- **Conducting a Basic Risk Assessment:** Identifying assets, threats, vulnerabilities, and calculating risks for a small IT system or business process, and proposing risk treatment options.
- **Designing a Data Classification Scheme:** Proposing a data classification scheme for an organization and outlining handling requirements for different data categories.
- **Simulating an Incident Response Plan:** Creating a simple incident response plan for a specific security incident type (e.g., a phishing attack or data breach) and outlining the steps to be taken.
- **Analyzing a Real-World Security Incident:** Researching a recent major information security breach, identifying its causes, impact, and the lessons learned for security management.

This project work aims to explore topics such as developing scalable security architectures, transitioning from closed to open security systems, and addressing ethical issues arising from data sensing and management in a practical context. Through a coordinated and integrated approach, students will gain a thorough understanding of the foundations, applications, and issues central to **Information Security and Management**.

Text Books:

1. Michael E.Whilman and Herbert J.Mattord, "Management of Information Security".
2. Mark Rhodes-Ousley, "Information Security: The Complete Reference, Second Edition", McGraw-Hill, 2013.

References:

1. Forouzan, "Cryptography & Network Security (Sie) 2E", McGraw-Hill Education (India) Pvt Limited.
2. Nina Godbole, "Information Systems Security, 2ed: Security Management, Metrics, Frameworks and Best Practices".

Web Resources:

1. http://www.iso.org/iso/home/standards/management-standards/iso27001.html (ISO 27001 standard information)
2. http://csrc.nist.gov/publications/nistpubs/800-55-Rev1/SP800-55-rev1.pdf (NIST SP 800-55 Rev. 1, Performance Measurement Guide for Information Security)
3. pu.informatics.global

Topics relevant to development of "SKILL DEVELOPMENT":

This course aims to develop the following **employability skills** through problem-solving methodologies and practical lab exercises, which will be assessed through various components mentioned in the course handout:

- **Information Security Fundamentals:** Ability to describe and apply core information security concepts (CIA triad, threat vectors).
- **Security Policy Implementation:** Practical skills in developing, implementing, and enforcing information security policies and procedures.
- **Security Roles Understanding:** Knowledge of various security roles and responsibilities within an organization, fostering awareness of career paths.
- **Risk Management:** Ability to identify, analyze, assess, and mitigate information security risks.
- **Data Protection Strategies:** Skills in recognizing data leakage threats and applying methods to protect sensitive information (e.g., data classification, basic DLP concepts).
- **Incident Response Planning:** Conceptual understanding and basic planning for responding to security incidents.
- **Compliance and Standards:** Familiarity with information security standards and frameworks (e.g., ISO 27001), enabling adherence to regulatory requirements.
- **Critical Thinking & Analysis:** Ability to analyze security scenarios, identify vulnerabilities, and propose appropriate security measures.

Course Code: CSE3520	**Course Title: Network Intrusion Detection and Prevention** **Type of Course:1] Program Core** **2] Theory Only**	**L- T-P-C**	3	0	3

Version No.		1.0
Course Pre-requisites		CSE2503 Cryptography and Network Security
Anti-requisites		**NIL**
Course Description		This course provides an in-depth study of **Intrusion Detection Systems (IDS)** and **Intrusion Prevention Systems (IPS)**. The objective is to understand when, where, how, and why to apply these tools and techniques to improve an enterprise's security posture. Students will gain knowledge of the fundamentals and history of intrusion detection to avoid common pitfalls in the creation and evaluation of new IDS. The course emphasizes practical skills in analyzing intrusion detection alerts and logs to distinguish attack types from false alarms.
Course Objectives		The objective of the course is to familiarize the learners with the concepts of **Intrusion Detection and Prevention Systems** and attain Skill Development through Participative Learning techniques.
Course Out Comes		On successful completion of the course, students shall be able to: • **CO1:** Understand various types of **intruders** and their motivations. (Knowledge) • **CO2:** Define and formulate **intrusion detection and prevention policies** for organizational security. (Comprehension) • **CO3:** Explain the fundamental concepts of **Network Protocol Analysis** and demonstrate the skill to capture and analyze network packets. (Comprehension, Application) • **CO4:** Use various **protocol analyzers** and **Network Intrusion Detection Systems** as security tools to detect network attacks and troubleshoot network problems. (Application)
Course Content:		

Module 1	**Introduction to Intrusion Detection and Prevention System**	Assignment	Programming Task	**10 Sessions**

- **Understanding Intrusion Detection:** Definition, goals, and history.
- **Intrusion Detection and Prevention Basics:** Differences between IDS and IPS.

- **IDS and IPS Analysis Schemes:**

- **Misuse Detection:** Signature-based detection, pattern matching.
- **Anomaly Detection:** Statistical anomaly, protocol anomaly, traffic anomaly.
- **Specification-based Detection:** Rule-based systems, stateful protocol analysis.
- **Hybrid Detection:** Combining multiple approaches.
- **Internal and External Threats to Data:** Understanding threat landscapes.

- **Need and Types of IDS:** Network-based IDS (NIDS), Host-based IDS (HIDS).
- **Information Sources for IDS:**
- **Host-based Information Sources:** System logs, audit trails, file integrity monitoring.
- **Network-based Information Sources:** Network traffic, flow data.

- **Intrusion Prevention Systems (IPS):** Inline deployment, blocking capabilities.
- **Network IDS (NIDS) vs. Protocol-based IDS:** Deep dive into specific types.
- **Hybrid IDS:** Integrating NIDS and HIDS.
- **Analysis Schemes and Intrusion Analysis Model:** Steps for analyzing intrusions.
- **Techniques for Intrusion Analysis:** Data reduction, correlation.
- **Responses to Intrusions:**
    - **Requirement of Responses:** Why timely responses are crucial.
    - **Types of Responses:** Active vs. Passive, automated vs. manual.
    - **Mapping Responses to Policy:** Aligning response actions with security policies.
- **Vulnerability Analysis:**
    - **Credential Analysis:** Assessing weaknesses in authentication mechanisms.
    - **Non-credential Analysis:** Identifying other system and network vulnerabilities.
- **Architecture Models of IDS and IPS:** Deployment strategies (e.g., inline, promiscuous mode).

Module 3	**Applications and tools**	Assignment	Programming/Data analysis task	**12 Sessions**

- **Tool Selection and Acquisition Process:** Criteria for choosing IDS/IPS solutions.
- **Overview of Major IDS Tools:**
    - **Bro Intrusion Detection (Zeek):** Introduction to its capabilities and event-driven scripting.
    - **Prelude Intrusion Detection:** Overview of its architecture and aggregation.
    - **Cisco Security IDS:** Commercial IDS solutions.
    - **Snort Intrusion Detection:**
        - Introduction to Snort: History, features, and modes of operation.
        - Snort Installation Scenarios: Source compilation vs. package installation.

- Installing Snort: Step-by-step procedure.
- Running Snort on Multiple Network Interfaces.
- Snort Command Line Options.
- Location of Snort Files (rules, logs, configurations).
- Snort Alert Modes (console, syslog, database).

- **Working with Snort Rules:**
  - **Rule Headers:** Action, protocol, source/destination IP/port.
  - **Rule Options:** Message, content, sid, rev.
  - **The Snort Configuration File:** Modifying `snort.conf`.

Module 4	Legal issues and organizations standards	Assignment	Programming/Data analysis task	9 Sessions

- **Law Enforcement / Criminal Prosecutions:** Role of IDS/IPS in digital forensics.
- **Standard of Due Care:** Legal obligations in cybersecurity.
- **Evidentiary Issues:** Admissibility of IDS logs in legal proceedings.
- **Organizations and Standardizations:**
  - Relevant industry standards (e.g., ISO 27000 series, NIST cybersecurity framework).
  - Legal frameworks (e.g., GDPR, HIPAA, relevant national laws).
  - Addressing common legal concerns and myths about Intrusion Detection Systems.

Textbooks:

1. Carl Endorf, Eugene Schultz, and Jim Mellander, "Intrusion Detection & Prevention", 1st Edition, Tata McGraw-Hill, 2004.
2. Earl Carter, Jonathan Hogue, "Intrusion Prevention Fundamentals", Pearson Education, 2006.

References:

1. Rafeeq Rehman, "Intrusion Detection with SNORT, Apache, MySQL, PHP and ACID", 1st Edition, Prentice Hall, 2003.
2. Christopher Kruegel, Fredrik Valeur, Giovanni Vigna, "Intrusion Detection and Correlation Challenges and Solutions", 1st Edition, Springer, 2005.
3. Paul E. Proctor, "The Practical Intrusion Detection Handbook", Prentice Hall, 2001.

Web Resources:

1. https://www.youtube.com/watch?v=RYB4cG8G2xo

2. https://www.coursera.org/lecture/detecting-cyber-attacks/intrusion-detection-systems-UeDqJ

Project Work/Assignment

Students will undertake a project or assignment focused on practical aspects of Network Intrusion Detection and Prevention. This can involve:

- **Network Packet Analysis and Report Generation:** Capturing and analyzing network packets from a given scenario (e.g., web browsing, file transfer, suspected attack) and generating a detailed report on observed protocols, anomalies, and potential security issues.
- **Custom Snort Rule Development and Testing:** Developing a set of custom Snort rules to detect specific, novel attack patterns or suspicious network behavior, and then rigorously testing these rules against simulated traffic.
- **IDS/IPS Deployment and Configuration in a Lab Environment:** Setting up a small network in a virtualized environment, deploying Snort (or a similar tool) as an NIDS/NIPS, and configuring its rules to protect specific services.
- **Incident Scenario Analysis and Response Plan:** Given a detailed network intrusion scenario (e.g., a malware outbreak, a data exfiltration attempt), analyze the provided logs and IDS alerts to identify the attack steps, propose containment and eradication strategies, and outline a post-incident recovery plan.
- **Research on Advanced IDS/IPS Techniques:** Investigating and presenting on advanced topics like AI/ML-driven anomaly detection, deception technologies (honeypots), or the use of threat intelligence feeds in IDS.

The assignments mentioned in the modules (e.g., "Demonstrating the skills to capture and analyze network packets using network packet analyzer," "Applying Intrusion detection in security applications," "Demonstrate the working with Snort Rules," and "Addressing common legal concerns and myths about Intrusion Detection system") will serve as foundational elements or sub-components of these larger projects.

Skill Development

This course aims to develop the following **employability skills** through problem-solving methodologies and practical lab exercises, which will be assessed through various components mentioned in the course handout:

- **Network Protocol Analysis:** Ability to capture, filter, and deeply analyze network packets to understand communication flows and identify anomalies.
- **Intrusion Detection System (IDS) Operation:** Practical skills in installing, configuring, and managing open-source IDS tools like Snort, including rule writing and alert interpretation.
- **Intrusion Prevention System (IPS) Concepts:** Understanding of IPS functionalities and deployment, with hands-on exposure to simulating prevention capabilities.
- **Security Policy Definition:** Ability to define and implement security policies that govern IDS/IPS behavior and incident response.
- **Threat Identification:** Skills in identifying various types of network attacks and attack signatures.
- **Security Log Analysis:** Competence in analyzing IDS alerts and system logs to identify malicious activity and distinguish it from false positives.

	• **Incident Response Fundamentals:** Conceptual understanding of the incident response lifecycle, with a focus on initial detection and analysis. • **Security Tool Proficiency:** Hands-on experience with industry-relevant network security tools. • **Legal & Ethical Awareness:** Understanding the legal implications of network security monitoring and data handling. • **Agent Development for Intrusion Detection (Conceptual/Scripting):** The ability to conceptualize or script small components that could contribute to intrusion detection logic, enhancing participative learning.

Course Code: CSE3521	**Course Title: Principles and Practices of Web Security** **Type of Course: Integrated**	**L- T-P-C**	2	0	2	3
Version No.	1					
Course Pre-requisites	CSE2503 – Cryptography and network security					
Anti-requisites	Nil					
Course Description	The purpose of this course is to introduce students to the field of **web security** by understanding web functionality and various security validations. The web is our gateway to many critical services and is quickly evolving as a platform to connect all our devices. **Web vulnerabilities** are growing on a year-to-year basis, and designing secure web applications is challenging. The course covers fundamental concepts of web security principles, web vulnerability and exploitation, various attacks on web applications, and a few basic topics on **web encryption**.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Web Security** and attain Skill Development through Experiential Learning techniques.					
Course Out Comes	On successful completion of the course, students shall be able to: • **CO1:** Define the fundamentals of **web applications** and **validation mechanisms**. (Knowledge) • **CO2:** Recognize the significance of **password** and **authentication** in web applications. (Comprehension) • **CO3:** Explain the importance of **session management** in web applications. (Comprehension) • **CO4:** Apply **web attack techniques** to find vulnerabilities in web applications. (Application) • **CO5:** Propose **defensive strategies** and secure coding practices against common web vulnerabilities. (Application, Synthesis) •					
Course Content:						
Module 1	**Introduction to Web Security Fundamentals**	Quiz		Comprehension based Quiz on web fundamentals	**10 Sessions**	
	• **Web Functionality:** HTTP/HTTPS, client-server model, web architecture overview. • **Encoding Schemes:** URL encoding, HTML encoding, JavaScript encoding – their purpose and misuse. • **Mapping the Application:** Enumerating content and functionality, understanding application flow. • **Analyzing the Application:** Using browser developer tools, proxy tools. • **Bypassing Client-Side Controls:** Transmitting data via the client, capturing user data,					

manipulating client-side logic.

- **Handling Client-Side Data Securely:** Input Validation (Client-side vs. Server-side).
- **Validation Techniques:** Blacklist Validation vs. Whitelist Validation.
- **Defense-in-Depth Approach:** Principles and application.
- **Attack Surface Reduction:** Minimizing exposure.
- **Rules of Thumb:** General security guidelines.
- **Classifying and Prioritizing Threats:** Risk assessment basics for web applications.

Module 2	**Web Application Authentication**	Assignment		Comprehensive based assignment on Web authentication	**11 Sessions**

- **Authentication Fundamentals:** Principles of authentication, common authentication types.
- **Two-Factor and Three-Factor Authentication:** Concepts, benefits, and implementation considerations.
- **Web Application Authentication Mechanisms:** Password-Based, Built-in (e.g., Basic, Digest HTTP), Single Sign-On (SSO), Custom Authentication.
- **Validating Credentials:** Secure password storage (hashing, salting), common pitfalls.
- **Secured Password-Based Authentication:**
  - Attacks against Password: Brute-force, dictionary attacks, credential stuffing, rainbow tables.
  - Importance of Password Complexity.
- **Design Flaws in Authentication Mechanisms:** Logic errors, insecure password recovery.
- **Implementation Flaws in Authentication Mechanisms:** Hardcoded credentials, improper error handling.
- **Securing Authentication:** Best practices, multi-factor authentication integration.

Module 3	**Session Management &Web Security Principles**	Quiz		Comprehension based Quiz on web security techniques.	**11 Sessions**

- **Need for Session Management:** Maintaining state in stateless HTTP.

- **Session Token Generation:** Weaknesses (predictable tokens, insufficient entropy).

- **Session Token Handling:** Weaknesses (session fixation, session hijacking, insecure cookie attributes).

- **Securing Session Management:** Best practices (randomness, expiration, HttpOnly, Secure flags, token rotation).

- **Access Control:**
  - **Access Control Overview:** Authentication vs. Authorization.
  - **Common Vulnerabilities:** Insecure Direct Object References (IDOR), Broken Function Level Authorization (BFLA), Missing Function Level Access Control.
  - **Attacking Access Controls:** Exploitation techniques.
  - **Securing Access Control:** Principle of Least Privilege, proper authorization checks.

- **Origin Policy:** Same-Origin Policy (SOP) and its importance.

- **Browser Security Principles:** Introduction to browser security mechanisms.

- **Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF):** High-level overview, types, and impact.

- **File Security Principles:**
  - **Source Code Security:** Protecting sensitive code.
  - **Forceful Browsing:** Direct access to protected resources.
  - **Directory Traversal (Path Traversal):** Exploiting file system access.

Module 4	**Web Application Vulnerability**	Assignment		Comprehension based assignment on web vulnerabilities	**10 Sessions**

- **Attacking Data-stores and Backend Components:**
  - **Injecting into Interpreted Contexts:** General injection concept.
  - **Injecting into SQL (SQL Injection):** Types (Union-based, Error-based, Blind), prevention.
  - **NoSQL Injection:** Attacks on NoSQL databases.
  - **XPath Injection, LDAP Injection:** Attacks on directory services.
  - **Injecting OS Commands (Command Injection):** Executing arbitrary commands on the server.
  - **Manipulating File Paths:** Exploiting file read/write vulnerabilities.
  - **Injecting into XML Interpreters (XXE):** XML External Entity attacks.
  - **Injecting into Back-end HTTP Requests (SSRF):** Server-Side Request Forgery.

- **Injecting into Mail Services:** Email header injection.

- **Attacking Application Logic:**
  - **Real-World Logic Flaws:** Examples of business logic vulnerabilities.

- **Attacking Users (Client-Side Attacks):**
  - **Cross-Site Scripting (XSS):** Varieties of XSS (Reflected, Stored, DOM-based), XSS attacks in action, finding and exploiting XSS vulnerabilities, preventing XSS attacks.

- **Other Techniques:**
  - **Cookie-based Attacks:** Session cookie manipulation, insecure cookie handling.
  - **HTTP Header Injection:** Manipulating HTTP response headers.

List of Laboratory Tasks:

Each experiment includes a low-level (foundational) and a high-level (advanced/complex) component to ensure comprehensive skill development. Students should use a controlled, vulnerable web application environment (e.g., DVWA, WebGoat, OWASP Juice Shop, or a custom-built vulnerable application) for these experiments.

Experiment 1: Web Application Basics & Encoding Schemes

- **Low-Level:** Using a web browser's developer tools, inspect HTTP requests and responses for a simple web page. Identify common HTTP headers and URL-encoded parameters.
- **High-Level:** Create a simple HTML form with special characters. Submit it and observe the URL encoding. Use a proxy tool (e.g., Burp Suite) to intercept the request and manually decode/re-encode the parameters, observing the effect on the application.

Experiment 2: Bypassing Client-Side Controls

- **Low-Level:** Interact with a web form that has client-side input validation (e.g., JavaScript to check email format). Bypass this client-side validation by disabling JavaScript or intercepting the request with a proxy and modifying the data before sending it to the server.
- **High-Level:** Identify an application that uses client-side data storage (e.g., localStorage, sessionStorage, cookies) for non-sensitive information. Manipulate this client-side data using browser developer tools and observe how it affects the application's behavior.

Experiment 3: Web Application Authentication: Password Attacks

- **Low-Level:** On a vulnerable web application, perform a simple dictionary attack

(manual or using a basic script) against a login form to guess weak passwords.
- **High-Level:** Using Burp Suite's Intruder tool, configure and execute a brute-force attack against a login mechanism. Analyze the responses to identify valid credentials based on different response lengths or error messages.

Experiment 4: Authentication Bypass Techniques

- **Low-Level:** Explore common authentication bypass techniques such as default credentials or exploiting weak password reset functionalities on a vulnerable web application.
- **High-Level:** Identify and exploit a logic flaw in an authentication mechanism (e.g., manipulating HTTP headers, changing POST parameters) to gain unauthorized access to an account without knowing the password.

Experiment 5: Session Management Vulnerabilities

- **Low-Level:** Log in to a web application and identify the session token (e.g., in cookies). Attempt to use the same session token after logging out to see if the session remains active.
- **High-Level:** Demonstrate a **Session Fixation** attack: capture a session ID from the application before a user logs in, then trick the user into logging in with that pre-determined session ID. Show how you can then hijack their authenticated session.

Experiment 6: Access Control Vulnerabilities

- **Low-Level:** Identify an Insecure Direct Object Reference (IDOR) vulnerability: access a resource (e.g., a user profile, a document) by directly manipulating an ID in the URL or request parameters to access data belonging to another user.
- **High-Level:** Demonstrate **Broken Function Level Authorization (BFLA)**: identify an administrative function or a function meant for a different user role, and attempt to access it as a lower-privileged user by directly navigating to the URL or sending a crafted request.

Experiment 7: Cross-Site Scripting (XSS) Attacks

- **Low-Level:** Perform a reflected XSS attack on a vulnerable web application by injecting a simple script (`<script>alert('XSS')</script>`) into a reflected parameter (e.g., search query, username field).
- **High-Level:** Demonstrate a stored XSS attack by injecting a persistent script into a comment section or profile field that executes for other users viewing the page. Discuss its potential impact (e.g., session hijacking, defacement).

Experiment 8: SQL Injection Attacks and Prevention

- **Low-Level:** Perform a basic SQL Injection attack on a login form using common payloads like `' OR '1'='1' --` to bypass authentication.
- **High-Level:** Perform a blind SQL Injection attack (e.g., using boolean-based or time-based techniques) to extract information from the database without direct error

messages. Discuss how parameterized queries prevent these attacks.

Experiment 9: Command Injection / OS Command Injection

- **Low-Level:** Identify a vulnerable input field where a web application executes system commands (e.g., a "ping" utility). Inject a simple command like `& whoami` or `; ls -la`.
- **High-Level:** Demonstrate command injection to achieve remote code execution (e.g., create a new file on the server, execute a reverse shell if permissible in the lab environment). Discuss input sanitization and least privilege as prevention.

Experiment 10: File Path Traversal (Directory Traversal)

- **Low-Level:** Identify a vulnerable file inclusion or image loading function. Attempt to perform a directory traversal attack using `../../` payloads to access a sensitive file outside the intended directory (e.g., `/etc/passwd` or `boot.ini`).
- **High-Level:** Demonstrate the impact of successful path traversal (e.g., reading sensitive configuration files, potentially writing files to arbitrary locations if combined with other vulnerabilities).

Experiment 11: Cross-Site Request Forgery (CSRF) Attacks

- **Low-Level:** Create a simple HTML page with a hidden form that, when clicked by a victim, performs an unwanted action on a vulnerable web application (e.g., changes their email, transfers funds).
- **High-Level:** Demonstrate a CSRF attack on a vulnerable web application that changes a user's password. Discuss the role of anti-CSRF tokens and SameSite cookies in prevention.

Experiment 12: HTTP Header Injection & Cookie-Based Attacks

- **Low-Level:** Using a proxy tool, intercept a request and manually modify HTTP headers (e.g., `User-Agent`, `Referer`) to observe if the application behaves differently or if it reveals information.
- **High-Level:** Explore **Cookie Tampering**: Modify a non-HttpOnly, non-Secure cookie value (e.g., user role, item quantity) and demonstrate how it can lead to privilege escalation or other unauthorized actions. Discuss secure cookie attributes.

Experiment 13: Web Tracking Mechanisms and Privacy

- **Low-Level:** Using browser developer tools, identify first-party and third-party cookies on popular websites. Analyze their purpose (e.g., session, analytics, advertising).
- **High-Level:** Discuss the ethical implications of excessive web tracking. Research and demonstrate how browser privacy extensions (e.g., uBlock Origin, Privacy Badger) can block tracking mechanisms.

Experiment 14: Using Web Application Security Scanners

- **Low-Level:** Use OWASP ZAP (or Nessus/Nmap conceptually) to perform a basic automated scan on a vulnerable web application. Review the scan report to identify the types of vulnerabilities detected.
- **High-Level:** Manually verify a few of the low-severity findings from the automated scan. Discuss the limitations of automated scanners and the importance of manual penetration testing.

Experiment 15: Secure Coding Practices: Input Validation and Output Encoding

- **Low-Level:** For a simple web form (e.g., username input), implement server-side **whitelist input validation** to only accept alphanumeric characters. Demonstrate how this prevents basic injection attacks.
- **High-Level:** For displaying user-generated content (e.g., a comment section), implement **output encoding** (e.g., HTML entity encoding) to prevent XSS. Demonstrate how injecting HTML/JavaScript payloads no longer executes.

Targeted Application & Tools that can be used

- **Application:** Building, testing, and securing modern web applications by identifying, exploiting (ethically), and mitigating various vulnerabilities. This includes securing the entire web application stack from front-end to back-end, including authentication, session management, and data handling.

- **Tools:**

  - **Vulnerable Web Applications:** OWASP Juice Shop, Damn Vulnerable Web Application (DVWA), WebGoat – for hands-on vulnerability practice.
  - **Proxy Tools:** Burp Suite (Community Edition), OWASP ZAP – for intercepting, analyzing, and modifying HTTP traffic.
  - **Network Scanners:** Nmap – for port scanning and service enumeration.
  - **Vulnerability Scanners:** Nessus (conceptual), OWASP ZAP (active scan) – for automated vulnerability discovery.
  - **Web Authoring Tools:** Standard web development environments for creating and testing secure web applications.
  - **Programming Languages:** Python, JavaScript, PHP – for developing exploits and defensive measures.
  - **Databases:** MySQL/PostgreSQL (for SQL Injection practice).

Project work/Assignment:

Assignment: Group assignment to identify and write different web exploits to demonstrate vulnerabilities in web applications. This project will involve a structured approach:

1. **Vulnerability Research:** Students will research a specific set of web vulnerabilities (e.g., from OWASP Top 10).
2. **Lab Setup:** Set up a vulnerable web application environment (e.g., DVWA, WebGoat, or a custom-built vulnerable application).
3. **Exploitation:** Develop and demonstrate (ethically) proof-of-concept exploits for the chosen vulnerabilities. This will involve using tools like Burp Suite, Nmap, and custom scripts.
4. **Mitigation Strategies:** Propose and implement code-level or configuration-level fixes to mitigate the exploited vulnerabilities.
5. **Reporting:** Document the vulnerabilities, exploitation steps, and mitigation strategies in a comprehensive report.
6. **Presentation/Demonstration:** Present their findings and demonstrations to the class.

This project work encourages exploring topics such as developing scalable secure architectures, moving from closed systems to open systems (secure coding practices for open-source components), and addressing ethical issues arising from data sensing and user tracking in web applications. Through this coordinated and integrated approach, students will gain a thorough knowledge of the foundations, applications, and issues that are central to Web Security.

Text Books & References

Text Books:

1. Dafydd Stuttard, Marcus Pinto, "The Web Application Hacker's Handbook", Wiley Publishing Inc.
2. Andrew Hoffman, "Web Application Security: Exploitation and Countermeasure for Modern Web Applications".

References:

1. B. Sullivan, V. Liu, and M. Howard, "Web Application Security", A B Guide. New York: McGraw-Hill Education, 2011.

Web Resources:

1. E-book Link R1: https://presiuniv.knimbus.com/user#/home
2. E-book Link R2: https://presiuniv.knimbus.com/user#/home
3. NPTEL / Swayam Link: Introduction to Information Security I, IIT Madras: https://nptel.ac.in/courses/106106129
4. PU Library Link: https://puniversity.informaticsglobal.com/login
5. OWASP Top 10: https://owasp.org/www-project-top-10/
6. PortSwigger Web Security Academy: https://portswigger.net/web-security

Topics relevant to "Skill Development":

This course aims to develop the following **employability skills** through experiential learning techniques and problem-solving methodologies, which will be attained through the assessment components mentioned in the course handout:

- **Web Application Fundamentals & Security:** Ability to define and understand the core components and security challenges of web applications.
- **Session Management & Web Security Principles:** Practical understanding and implementation of secure session management and other fundamental web security principles.
- **Web Application Vulnerability Identification:** Skills in identifying, analyzing, and classifying common web application vulnerabilities (e.g., SQL Injection, XSS, CSRF, Access Control flaws).
- **Web Exploitation Techniques:** Hands-on experience in ethically exploiting web vulnerabilities to understand their impact.
- **Secure Coding Practices:** Ability to apply defensive coding techniques to prevent common web application attacks, especially focusing on **Input Validation** and **Output Encoding**.
- **Security Tool Proficiency:** Competence in using industry-standard tools like web proxies

	(Burp Suite, OWASP ZAP), vulnerability scanners, and network reconnaissance tools. • **Problem Solving & Analytical Thinking:** Ability to diagnose security flaws, devise exploitation strategies, and formulate effective countermeasures. • **Ethical Hacking Mindset:** Developing a responsible and ethical approach to discovering and reporting vulnerabilities. • **Web Tracking and Privacy Awareness:** Understanding the mechanisms of web tracking and associated privacy implications

Course Code: CSE3522	**Course Title: Penetration Testing and Risk Assessment** **Type of Course: Theory Only Course**		**L-T- P-C**	3	0	0	3
Version No.		1.0					
Course Pre-requisites		CSE2503 – Cryptography and Network Security					
Anti-requisites	**NIL**						
Course Description	This course provides a comprehensive exploration of **Penetration Testing** and **Risk Assessment** methodologies. It covers essential tools and techniques used for **information gathering**, **vulnerability analysis** (both automated and manual), and the **exploitation of common security weaknesses** across various domains including data systems, web applications, mobile applications, and wireless networks. Students will also gain an understanding of advanced exploitation frameworks like Metasploit and Meterpreter to automate attacks and penetration testing techniques.						
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Vulnerability Assessment and Penetration Testing** and attain Employability through Problem Solving Methodologies.						
Course Out Comes	On successful completion of the course, students shall be able to: • **CO1:** Understand the basic principles and techniques for **information gathering** and **detecting vulnerabilities** in target systems. (Knowledge, Comprehension) • **CO2:** Determine and analyze **security threats and vulnerabilities** specifically in SDN networks and web applications. (Analysis, Application) • **CO3:** Apply and utilize **exploitation techniques** relevant to mobile applications and wireless networks. (Application) • **CO4:** Understand and leverage **Metasploit** and **Meterpreter** for automating attacks and streamlining penetration testing. (Comprehension, Application) • **CO5:** Conduct basic **risk assessment** based on identified vulnerabilities and potential impacts. (Application)						
Course Content:							

Module 1	**Information Gathering, Host Discovery and Evading Techniques**	Quiz		Theory	**10 Sessions**
	• **Introduction to Penetration Testing:** Terminologies, Categories of Penetration Testing (Black Box, White Box, Grey Box), Phases of Penetration Test (Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks). • **Penetration Testing Reports:** Structure and importance.				

- **Information Gathering Techniques:**
  - **Passive Reconnaissance:** OSINT (Open-Source Intelligence), domain registration records, public archives, social media.
  - **Active Reconnaissance:** DNS queries, banner grabbing.
  - **Sources of Information Gathering:** Search engines, public databases.
- **Host Discovery:** Techniques for identifying active hosts on a network.
- **Scanning for Open Ports and Services:** Types of Port scans (TCP SYN, UDP, ACK, FIN, NULL, Xmas), Port states.
- **Vulnerability Scanners:** Functionality, pros, and cons.
- **Vulnerability Assessment with NMAP:** Using Nmap Scripting Engine (NSE) for vulnerability detection.
- **Testing SCADA Environment with NMAP (Conceptual):** Specific considerations for industrial control systems.

Module 2	**Vulnerability Scanner in SDN Networks and Web application**	Quiz		Theory	**10 Sessions**

- **Nessus Vulnerability Scanner:** Introduction to features, capabilities, and report analysis.
- **Vulnerability Data Resources:** CVE (Common Vulnerabilities and Exposures), NVD (National Vulnerability Database).
- **Software-Defined Networking (SDN) Security:**
  - **SDN Architecture:** Data Plane, Control Plane, Application Plane.
  - **SDN Security Attack Vectors:** Attacks on controller, data plane, communication.
  - **SDN Hardening:** Security measures for SDN.
- **Web Application Vulnerabilities (Advanced):**
  - **Authentication Bypass with Insecure Cookie Handling:** Exploiting weak session management.
  - **XSS Vulnerability (Advanced):** Persistent XSS, DOM XSS, XSS exploitation scenarios.
  - **File Inclusion Vulnerability:** Local File Inclusion (LFI) and Remote File Inclusion (RFI).
  - **Patching File Inclusions:** Secure coding practices.
  - **Testing a Website for SSI (Server-Side Include) Injection:** Exploitation and prevention.

Module 3	Mobile Application Security and wireless network Vulnerability analysis	Quiz		Theory	11 Sessions

- **Mobile Application Security:**
  - **Types of Mobile Applications:** Native, Hybrid, Web.
  - **Key Challenges in Mobile Application Security:** Data storage, insecure communication, lack of binary protections.
  - **Mobile Application Penetration Testing Methodology:** Stages of mobile PT.
  - **Android and iOS Vulnerabilities:** Common weaknesses (e.g., insecure data storage, weak cryptography, insecure IPC).
  - **OWASP Mobile Security Risk:** Top 10 mobile risks.
- **Exploiting Mobile Platforms (Conceptual):**
  - Exploiting Windows Mobile (WM), BlackBerry Vulnerabilities (Historical Context).
  - Vulnerability Landscape for Symbian (Historical Context).
  - Exploit Prevention in Mobile Apps.
  - Handheld Exploitation.
- **Wireless Network Vulnerability Analysis:**
  - **WLAN and its Inherent Insecurities:** Open networks, WEP, WPA/WPA2 weaknesses.
  - **Bypassing WLAN Authentication:** Fake authentication, deauthentication attacks.
  - **Uncovering Hidden SSIDs, MAC Filters Bypassing:** Techniques and tools.
  - **Advanced WLAN Attacks:** Wireless eavesdropping using MITM (Man-in-the-Middle) attacks, session hijacking over wireless.
  - **WLAN Penetration Test Methodology:** Steps for wireless security assessment.

Module 4	Exploits	Quiz		Theory	8 Sessions

- **Introduction to Exploits:** Definition, types (remote, local).
- **Metasploit Framework:**
  - **Architecture and Environment:** Modules (exploits, payloads, auxiliary, post), databases, user interfaces (msfconsole).
  - **Leveraging Metasploit on Penetration Tests:** Workflow and phases.
  - **Understanding Metasploit Channels, Metasploit Framework, and Advanced Environment Configurations.**
  - **Understanding the Soft Architecture, Configuration, and Locking.**
- **Advanced Payloads and Add-on Modules:** Bind vs. Reverse shells, stageless vs. staged payloads.

	• **Global Datastore, Module Datastore, Saved Environment.** • **Meterpreter:** • **Introduction to Meterpreter:** Advanced payload, in-memory execution. • **Meterpreter Commands:** File system interaction, process migration, screenshot, webcam, keylogging. • **Post-Exploitation with Meterpreter:** Privilege escalation, persistence mechanisms.
	Targeted Application & Tools that can be used: **Application:** Performing comprehensive **penetration tests** on various IT infrastructures, including corporate networks, web applications, mobile applications, and wireless networks. This course equips students to understand and mitigate threats and vulnerabilities in real-world scenarios. **Tools:** • **Nmap:** Network scanner for information gathering and vulnerability assessment. • **Nessus (Conceptual/Trial):** Vulnerability scanner for identifying security weaknesses. • **Burp Suite (Community Edition):** Web proxy for intercepting and manipulating web traffic. • **OWASP ZAP:** Open-source web application security scanner. • **Aircrack-ng Suite (Conceptual/Lab Environment):** Wireless network auditing tools. • **Metasploit Framework:** Penetration testing framework for exploitation and post-exploitation. • **Kali Linux / Parrot OS:** Penetration testing distributions with pre-installed tools. • **Virtualization Software:** (e.g., VirtualBox, VMware Workstation Player) for creating isolated lab environments. • **ADB (Android Debug Bridge):** For mobile application analysis.
	Project work/Assignment: Students will work on a comprehensive **Penetration Testing Project** that simulates a real-world scenario. This project will involve: 1. **Scope Definition & Reconnaissance:** Defining the scope of a simulated target environment (e.g., a small company network, a web application with a mobile component). Performing extensive information gathering (passive and active) to map the target. 2. **Vulnerability Scanning & Analysis:** Using automated tools (Nmap, Nessus/OWASP ZAP) and manual techniques to identify vulnerabilities across network services, web applications, or mobile apps within the scope. 3. **Exploitation:** Ethically attempting to exploit identified vulnerabilities using techniques learned (e.g., SQL Injection, XSS, file inclusion, Metasploit exploits) to gain access or achieve a specific objective. 4. **Post-Exploitation & Lateral Movement (Conceptual):** If initial exploitation is successful, explore basic post-exploitation techniques (e.g., information gathering on the compromised host, simple lateral movement). 5. **Risk Assessment:** Based on the discovered vulnerabilities and successful exploitations, perform a basic risk assessment, prioritizing findings based on likelihood and impact. 6. **Reporting & Recommendations:** Preparing a professional penetration test report detailing the findings, exploitation steps, impact, and actionable remediation recommendations. 7. **Ethical Considerations:** Throughout the project, strictly adhere to ethical hacking principles and operate within the defined scope, documenting all actions. This project work encourages exploring topics such as developing scalable secure architectures, moving from closed systems to open systems (e.g., assessing open-source components for vulnerabilities), and addressing ethical issues rising from data sensing, smart transportation, and

surveillance applications in a penetration testing context. Through this coordinated and integrated approach, students will gain a thorough knowledge of the foundations, applications, and issues that are central to **Penetration Testing and Risk Assessment**.

Text Books:

1. Rafay Baloch, "Ethical Hacking and Penetration Testing Guide", CRC Press, 2015. ISBN: 978-1-4822-3161-8.
2. Dr. Patrick Engebretson, "The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy", Syngress Publications, Elsevier, 2013. ISBN: 978-0-12-411644-3.
3. Mayor, K.K. Mookey, Jacopo Cervini, Fairuzan Roslan, Kevin Beaver, "Metasploit Toolkit for Penetration Testing, Exploit Development and Vulnerability Research", Syngress Publications, Elsevier, 2007. ISBN: 978-1-59749-074-0.

References:

1. Prakhar Prasad, "Mastering Modern Web Penetration Testing", Packt Publishing, October 2016.
2. Justin Clarke-Salt, "SQL Injection Attacks and Defense", 1st Edition, Syngress Publication.

Web Resources:

1. NPTEL / Swayam Link: IIT Kharagpur, Prof. Indranil Sen Gupta: https://onlinecourses.nptel.ac.in/noc19_cs68/preview
2. OWASP Top 10: https://owasp.org/www-project-top-10/
3. OWASP Mobile Security Testing Guide (MSTG): https://owasp.org/www-project-mobile-security-testing-guide/
4. Metasploit Documentation: https://www.metasploit.com/modules/

3.

Topics relevant "SKILL DEVELOPMENT":

This course aims to develop the following **employability skills** through experiential learning techniques and problem-solving methodologies, which will be attained through the assessment components mentioned in the course handout:

- **Information Gathering & Reconnaissance:** Ability to effectively gather information about target systems and networks.
- **Vulnerability Assessment:** Skills in identifying, analyzing, and classifying security weaknesses using various tools and techniques.
- **Exploitation Techniques:** Hands-on experience in ethically exploiting vulnerabilities across different platforms (web, mobile, network, wireless).
- **Penetration Testing Methodologies:** Understanding and applying structured approaches to penetration testing.
- **Security Tool Proficiency:** Competence in using industry-standard penetration testing tools (e.g., Nmap, Burp Suite, Metasploit, Aircrack-ng).
- **Risk Assessment:** Ability to evaluate the likelihood and impact of vulnerabilities and prioritize remediation efforts.
- **Reporting & Communication:** Skills in documenting findings and communicating security risks effectively to technical and non-technical audiences.

	• **Ethical Hacking Mindset:** Developing a responsible, legal, and ethical approach to security testing. • **Problem Solving & Analytical Thinking:** Ability to diagnose complex security flaws, devise exploitation paths, and formulate effective countermeasures. • **Advanced Exploitation:** Practical knowledge of using frameworks like Metasploit for automating attacks and post-exploitation.

<table>
<tr><td>Course Code:
CSE3523</td><td colspan="3">Course Title: Introduction to Fintech
Type of Course: Lab Integrated</td><td>L-T-P-C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td></td><td colspan="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td>•</td><td colspan="7">FIN1001</td></tr>
<tr><td>Anti-requisites</td><td></td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td></td><td colspan="7">This course aims to familiarize students with the FinTech ecosystem and the disruptive and innovative forces of emerging technology within the finance sector. A core component of a specialized business administration program, it delivers essential, leading-edge knowledge in financial technology, crucial for professionals entering the banking and financial services industry.</td></tr>
<tr><td>Course Object</td><td></td><td colspan="7">The objective of the course is to familiarize the learners with the concepts of Introduction to Fintech attain Skill Development through Experiential Learning techniques.</td></tr>
<tr><td>Course Out Comes</td><td></td><td colspan="7">On successful completion of the course the students shall be able to:
CO1: Describe the historical development of financial technology. (Understand)
CO2: Analyze the impact of financial technology on the financial services landscape. (Apply)
CO3: Explain the fundamental technical aspects of financial technology. (Apply)
CO4: Identify and interpret key technological trends within the financial services sector. (Apply)</td></tr>
<tr><td>Course Content:</td><td></td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td>Introduction to FinTech</td><td>Assignment</td><td></td><td colspan="4"></td><td>14 Sessions</td></tr>
<tr><td colspan="9"><u>Topics:</u>

What is FinTech Industry? Evolution of FinTech, FinTech Evolution 1.0: Infrastructure, FinTech Evolution 2.0: Banking industry, FinTech Evolution 3.0 & 3.5: Startups and Emerging Markets, Importance of FinTech, Global FinTech Investment, Main FinTech Hubs</td></tr>
<tr><td>Module 2</td><td>FinTech Reshaping Financial Services Industry-I</td><td>Assignment</td><td></td><td colspan="4"></td><td>15 Sessions</td></tr>
<tr><td colspan="9"><u>Topics:</u>

FinTech in Payment Industry-Multichannel digital wallets, applications supporting wallets, onboarding and KYC application, FinTech in Lending Industry- Formal lending, Informal lending, P2P lending, POS lending, Online lending, Payday lending, Microfinance, Crowdfunding</td></tr>
<tr><td>Module 3</td><td>FinTech as disruptor empowering</td><td>Assignment</td><td></td><td colspan="4"></td><td>16 Sessions</td></tr>
</table>

	Financial Services Industry-II				
	Topics: FinTech in Wealth Management Industry-Financial Advice, Automated investing, Socially responsible investing, Fractional Investing, Social Investing. FinTech in Insurance Industry- P2P insurance, On-Demand Insurance, On-Demand Consultation, Customer engagement through Quote to sell, policy servicing, Claims Management, Investment linked health insurance.				
Module 4	Technology Disruptions enabling FinTech Innovations	Assignment			**15 Sessions**
	Topics: 4G and 5G networks fuelling FinTech Opportunities, transforming customer experience using Mobile Applications and smart phones, embedded sensors and social media, Cloud computing, Web 2.0, Rapid Web Design, JavaScript Technologies, IoT, Big Data, analytics and AI and Blockchain Case Studies: PayTm, Aadhar				
	Project work/Assignment:				
1.	2. **Assignment 1 on (Module 1 and Module 2)** 3. **Assignment 2 on (Module 3 and Module 4)**				
	Text Book Parag Y Arjunwadkar (2018), FinTech: The Technology Driving Disruption in the financial service industry CRC Press. 1) Sanjay Phadke (2020), Fintech Future : The Digital DNA of Finance Paperback .Sage Publications 2) Pranay Gupta, T. Mandy Tham (2018). Fintech: The New DNA of Financial Services Paperback 3) RBI(2017). Report of working group on FinTech and Digital Banking				
	References **1. Bitcoin for Non-Mathematicians: Exploring the foundations of Crypto, SlavaGomzin/ Universal Publishers, USA, Latest 1 ST Edition 2020** **2. The Robotics Process Automation, Handbook: A Guide to Implementing, Tom Taulli/ Apress, Latest 1 ST Edition 2020** **Web Resources** W1. https://www.ibm.com/industries/banking-financial-markets/resources/omnichannelbanking-paper/ W2. https://thefinancialbrand.com/111080/evolution-future-digital-banking-baastransformation/				
	Topics relevant to development of "Employability": Real time Analysis of FinTech applications and opportunities. **Topics relevant to "PROFESSIONAL ETHICS":** Case studies on Paytm, AAdhar				

<table>
<tr><td>Course Code:
CSE3524</td><td colspan="2">Course Title: Banking Technology

Type of Course: Program Core
Theory & Lab Integrated</td><td>L-T-P-C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td></td><td colspan="6">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td>•</td><td colspan="6">FIN1001</td></tr>
<tr><td>Anti-requisites</td><td></td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td></td><td colspan="6">This course explores the evolution and application of technology in banking, starting from branch computerization to centralized banking. It covers delivery channels such as ATMs, internet/mobile banking, and UPI/BHIM. Topics include risk management, treasury, forex operations, and CRM. Students will understand INFINET, SFMS, RTGS, NEFT, and e-payment systems. Emphasis is placed on practical implementation and emerging digital banking trends.</td></tr>
<tr><td>Course Object</td><td></td><td colspan="6">The objective of the course is to familiarize the learners with the concepts of Banking Technology attain Skill Development through Experiential Learning techniques.</td></tr>
<tr><td>Course Out Comes</td><td></td><td colspan="6">On successful completion of the course the students shall be able to:
CO1: Understand the evolution of banking technology and its impact on modern banking operations.(Understand)
CO2: Explain the role of delivery channels like ATMs, mobile banking, and UPI in digital banking systems.(Apply)
CO3: Evaluate the effectiveness of risk, treasury, and data center management in banking operations.(Apply)
CO4: Analyze the functioning of centralized banking and payment systems like RTGS, NEFT, and SFMS (Apply)
CO5: Understand blockchain, cryptocurrency concepts, and analyze recent core banking software. (Understand)</td></tr>
<tr><td>Course Content:</td><td></td><td colspan="6"></td></tr>
<tr><td>Module 1</td><td>Branch Operation and Core Banking</td><td>Assignment</td><td></td><td colspan="3"></td><td>14 Sessions</td></tr>
<tr><td></td><td colspan="7"><u>Topics:</u>

Introduction and Evolution of Bank Management Analysis of Rangarajan - Committee Reports - Technological Impact in Banking Operations– Total Branch Computerization - Concept of Opportunities– Centralized Banking – Concept, Opportunities, Challenges & Implementation.</td></tr>
<tr><td>Module 2</td><td>Delivery Channels</td><td>Assignment</td><td></td><td colspan="3"></td><td>15 Sessions</td></tr>
</table>

	Topics: Overview of delivery channels – Automated Teller Machine (ATM) – Phone Banking –Call centers – Internet Banking – Mobile Banking- USSD, UPI, BHIM – Payment Gateways – Card technologies – MICR electronic clearing.					
Module 3	Back office Operations	Assignment				**16 Sessions**
	Topics: Bank back office management –Inter branch reconciliation – Treasury Management– Forex Operations – Risk Management – Data center Management – Network Management – Knowledge Management (MIS/DSS/EIS) – Customer Relationships Management (CRM).					
Module 4	Interbank Payment System	Assignment				**15 Sessions**
	Topics: INFINET Interface with Payment system Network– Structured Financial Messaging system – Electronic Fund transfer – RTGSS – Negotiated Dealing Systems & Securities Settlement Systems – Electronic Money – E Cheques.					
Module 5	Contemporary Issues in Banking					
	Techniques Block Chain and Bit-coin – Crypto currency Analysis of Recent Core Banking Software-Case study.					
	Project work/Assignment:					
5.	6. **Assignment 1 on (Module 1 and Module 2)** 7. **Assignment 2 on (Module 3,4 and Module 5)**					
	Text Book 1) Financial Services Information Systems-Jessica Keyes Auerbach publication, October 2019. 2) **Rajesh, R.** (2020). *Banking Technology*. New Delhi: McGraw Hill Education. 3) IIBF X Taxmann's International Trade Finance – Complete | Expert-vetted Guide on—Trade Theories | ICC Rules | Regulatory Frameworks | Risk Management | Digitisation | LIBOR-ARR Transitions, February 2025 4) **Kalakota, R., & Robinson, M.** (2017). *E-Banking Management: Issues, Solutions, and Strategies.* New Delhi: Pearson Education.					
	References **1. Vasudeva,E–Banking, Common Wealth Publishers, New Delhi, 2010** **2. Turban Rainer Potter, Information Technology, John Wiely& Sons Inc,2012.** **3. Banking Technology – Indian Institute of Bankers Publication,2010.** **Web Resources** W2. https://www.ibm.com/industries/banking-financial-markets/resources/omnichannelbanking-paper/ W3. https://thefinancialbrand.com/111080/evolution-future-digital-banking-baastransformation/					
	Design and Develop the following Banking Software using the appropriate technologies:					

	■ Mobile Banking **▪ Balance Enquiry ▪ Cheque book Request ▪ Stop Cheque ▪ Credit/Debit Notification ▪ Bill Payment** **■ Internet Banking** **▪ Electronic Funds Transfer ▪ Account Management ▪ Loan Application ▪ Registering of new bank services ▪ Customer Information Management** **■ATM system** **▪ Balance Enquiry ▪ Withdrawal ▪ Deposit ▪ Pin change ▪ Mini statement**
	Topics relevant to development of "Employability": Real time Data Analysis for Banking Technology. **Topics relevant to "PROFESSIONAL ETHICS":** Mobile, Internet Banking for Project Development.

Course Code: **CSE3525**	**Course Title: Blockchain Technology** **Type of Course: Theory**	**L-T-P-C**	2	0	2	3
Version No.		1.0				
Course Pre-requisites	•	CBC2000				
Anti-requisites		NIL				
Course Description		This course provides a comprehensive introduction to the fundamental concepts and applications of blockchain technology. Students will explore the principles of decentralization, understand the mechanics of Bitcoin, delve into the world of smart contracts and alternative cryptocurrencies, and gain practical experience in developing and deploying smart contracts using industry-standard tools. The course emphasizes both the theoretical underpinnings and the practical implementation of blockchain solutions.				
Course Object		The objective of the course is to familiarize the learners with the concepts of Blockchain Technology attain **Skill Development** through **Experiential Learning** techniques.				
Course Out Comes		On successful completion of the course the students shall be able to: **CO1: Understand** blockchain and decentralization principles. (Understand) **CO2: Analyze** Bitcoin's operation and transactions. (Analyze) **CO3: Apply** smart contract concepts and explore altcoins. (Apply) **CO4: Utilize** Truffle for smart contract development and deployment. (Apply)				
Course Content:						

Module 1	**Fundamentals of Blockchain and Decentralization**	Assignment			**14 Sessions**
	Topics: Introduction to Blockchain: the growth of blockchain technology, the history of blockchain and Bitcoin, distributed systems, blockchain, consensus. Decentralization: decentralization using blockchain, methods of decentralization, routes to decentralization, blockchain and full ecosystem decentralization, pertinent terminology, platforms for decentralization, innovative trends				
Module 2	**Bitcoin: The First Cryptocurrency**	Assignment			**15 Sessions**
	Topics: Introduction to Bitcoin: bitcoin — an overview, cryptographic keys, transactions, blockchain, mining. The Bitcoin Network and Payments: the bitcoin network, wallets,				

<table>
<tr><td></td><td colspan="5">bitcoin payments, innovation in bitcoin, advanced protocols, bitcoin investment and buying and selling Bitcoin. Bitcoin Clients and APIs: bitcoin client installation, experimenting further with bitcoin-cli, bitcoin programming</td></tr>
<tr><td>Module 3</td><td>Beyond Bitcoin: Altcoins and Smart Contracts</td><td>Assignment</td><td></td><td></td><td>16 Sessions</td></tr>
<tr><td></td><td colspan="5"><u>Topics:</u>

Consensus Algorithms: introducing the consensus problem, analysis and design, classification, algorithms, choosing an algorithm. Alternative Coins: introducing altcoins, theoretical foundations, difficulty adjustment and retargeting algorithms, bitcoin limitations, extended protocols on top of bitcoin, development of altcoins, Initial Coin Offerings (ICOs). Smart Contracts: history, definition, ricardian contracts, smart contract templates, oracles, deploying smart contracts, the DAO</td></tr>
<tr><td>Module 4</td><td>Ethereum and the Decntralized Web (Web3)</td><td>Assignment</td><td></td><td></td><td>15 Sessions</td></tr>
<tr><td></td><td colspan="5"><u>Topics:</u>
Ethereum 101: ethereum – an overview, the ethereum network, components of the ethereum ecosystem, Ethereum Virtual Machine (EVM), smart contracts, ethereum development environment . Further Ethereum: blocks and blockchain, wallets and client software, nodes and miners, APIs, tools, and DApps, supporting protocols, programming languages. Introducing Web3: contract deployment, exploring Web3 with Geth</td></tr>
<tr><td></td><td colspan="5">Project work/Assignment:</td></tr>
<tr><td>1.</td><td colspan="5">2. Assignment 1 on (Module 1 and Module 2)
3. Assignment 2 on (Module 3 and Module 4)</td></tr>
<tr><td></td><td colspan="5"></td></tr>
<tr><td></td><td colspan="5">Text Book

1. Banafa, A. (2024). Blockchain technology and applications. River Publishers.
2. Ramachandran, M. (2025). Blockchain engineering: Secure, sustainable frameworks for healthcare applications. Springer.
3. Tanwar, S. (2022). Blockchain technology: From theory to practice. Springer.
4. Vyas, S., Shukla, V. K., Gupta, S., & Prasad, A. (Eds.). (2022). Blockchain technology: Exploring opportunities, challenges, and applications. CRC Press.</td></tr>
<tr><td></td><td colspan="5">References
1. Chuen, D. L. K. (Ed.). (2024). Handbook of digital currency: Bitcoin, innovation, financial instruments and big data (2nd ed.). Academic Press.
2. Idrees, S. M. & Nowostawski, M. (Eds.). (2023). Transformations through blockchain technology: The new digital revolution. Springer.
3. Jena, A. K., Panda, S. K., & Swain, S. K. (Eds.). (2022). Blockchain technology: Applications and challenges (Vol. 203). Springer.</td></tr>
</table>

	4. Maleh, Y., Zhang, J., & Hansali, A. (2024). *Advances in emerging financial technology and digital money*. Routledge. 5. Rahman, H. (Ed.). (2025). *Blockchain technology applications in knowledge management*. IGI Global. **Web Resources** 1. Blockgeeks. Retrieved from https://www.google.com/search?q=blockgeeks.com 2. Bitcoin.org. Retrieved from https://bitcoin.org/ 3. CoinDesk. Retrieved from https://www.coindesk.com/ 4. Ethereum.org. Retrieved from https://ethereum.org/ 5. Investopedia. Retrieved from https://www.investopedia.com/ 6. Medium. Retrieved from https://medium.com/ 7. Solidity Documentation. Retrieved from https://docs.soliditylang.org/ 8. Truffle Suite Documentation. Retrieved from https://trufflesuite.com/docs 9. Web3.js Documentation. Retrieved from https://web3js.readthedocs.io/ 10. GitHub. Retrieved from https://github.com/ **YouTube Channels:** 1. Andreas Antonopoulos. Retrieved from https://www.youtube.com/@aantonop 2. Chainlink. Retrieved from https://www.youtube.com/@chainlink 3. Coin Bureau. Retrieved from https://www.youtube.com/@CoinBureau 4. Eat The Blocks. Retrieved from https://www.youtube.com/@EatTheBlocks 5. freeCodeCamp.org. Retrieved from https://www.youtube.com/@freecodecamp 6. MetaMask. Retrieved from https://www.youtube.com/@MetaMask 7. Patrick Collins. Retrieved from https://www.youtube.com/@patrickdcollins 8. Simply Explained. Retrieved from https://www.youtube.com/@SimplyExplained 9. The Defiant. Retrieved from https://www.youtube.com/@TheDefiant 10. Whiteboard Crypto. Retrieved from https://www.youtube.com/@WhiteboardCrypto
	Experiment 1: Integrated Development Environments (IDEs) for Smart Contracts • **Level 1:** Explore the features and interface of Remix IDE. Deploy a simple "Hello World" smart contract on the in-browser JavaScript VM. Observe the transaction details and contract interaction options. • **Level 2:** Install and configure MetaMask browser extension. Connect MetaMask to the Remix IDE. Deploy the same "Hello World" contract to the Ganache private network via MetaMask. Examine the transaction process in both Remix and MetaMask. **Experiment 2: MetaMask in a Private Network** • **Level 1:** Set up a local Ganache private network. Add a custom network in MetaMask, configuring the RPC URL and Chain ID to connect to your Ganache instance. Create a new account in MetaMask and observe its balance. • **Level 2:** Deploy a simple token contract (e.g., ERC-20 minimal) using Remix IDE and MetaMask on your private Ganache network. Transfer some tokens between the accounts you created in MetaMask and observe the balance changes. **Experiment 3: Smart Contract with Solidity - Basic Data Types and Structures** • **Level 1:** Write a Solidity smart contract that declares and initializes variables of different basic data types (uint, string, bool, address). Implement functions to read and

	modify these variables. Deploy and interact with the contract in Remix IDE. • **Level 2:** Create a Solidity smart contract that utilizes structs and arrays. Implement functions to add, retrieve, and update elements within these data structures. Deploy and test the contract with various inputs in Remix IDE. **Experiment 4: Smart Contract with Solidity - Control Flow and Functions** • **Level 1:** Write a Solidity smart contract that uses if-else statements and for loops within its functions. Implement a function that performs a simple calculation based on input parameters. Deploy and test the different control flow paths in Remix. • **Level 2:** Design and implement a Solidity smart contract with multiple functions, including internal and private functions. Demonstrate how these functions can be called and how visibility modifiers affect their accessibility. **Experiment 5: Contract Deployment** • **Level 1:** Deploy a pre-written simple smart contract (provided by the instructor) using MetaMask connected to the Ganache network. Observe the deployment transaction details (gas used, transaction hash, contract address). • **Level 2:** Explore different deployment parameters in Remix IDE (e.g., setting gas limit and gas price). Deploy the same contract multiple times with varying gas settings and analyze the impact on deployment cost and confirmation time in Ganache. **Experiment 6: MetaMask and Remix IDE Interaction** • **Level 1:** Deploy a simple counter smart contract using Remix IDE on the Ganache network via MetaMask. Use the Remix interface to call the contract's functions (e.g., increment, decrement, get count) and observe the state changes reflected in both Remix and MetaMask (balance changes for transactions). • **Level 2:** Deploy a more complex smart contract (e.g., a simple voting contract) using Remix and MetaMask. Interact with the contract through MetaMask's custom interaction interface (sending transactions to specific functions with appropriate arguments). **Experiment 7: Use of Geth - Installation and Account Management** • **Level 1:** Install the Geth Ethereum client on your local machine. Use Geth commands to create new Ethereum accounts and list the available accounts. Observe the keystore directory where private keys are stored. • **Level 2:** Use Geth commands to export and import Ethereum account private keys. Understand the security implications of managing private keys. Connect the Geth console to a running private network (e.g., Ganache or a custom Geth network). **Experiment 8: Genesis Block Creation in Geth** • **Level 1:** Understand the structure of a Genesis Block JSON file. Modify a sample Genesis Block configuration (e.g., changing the initial coin distribution). Initialize a new Geth data directory using this modified Genesis Block. • **Level 2:** Create a custom Genesis Block for a private Ethereum network with specific pre-allocated accounts, custom gas limit, and difficulty. Start a Geth node using this custom Genesis Block and connect to it using the Geth console. **Experiment 9: Interacting with a Private Geth Network** • **Level 1:** Start a Geth node using a previously initialized data directory. Use the Geth console to check the node's peer count and block number. Create a transaction to send Ether between two accounts within your private network using Geth commands. • **Level 2:** Deploy a simple smart contract to your private Geth network using the Geth console and web3.js (or similar library). Interact with the deployed contract's functions using the Geth console.

Experiment 10: Exploring Ethereum Transaction Structure

- **Level 1:** Send a transaction (Ether transfer) using MetaMask on the Ropsten test network. Examine the transaction details on a block explorer (e.g., Etherscan for Ropsten) and identify key fields like to, from, value, gas limit, gas price, and nonce.
- **Level 2:** Construct and sign a raw Ethereum transaction using web3.js (or similar library) without relying on MetaMask. Broadcast this signed transaction to a test network (e.g., Ropsten) and analyze its details on a block explorer.

Experiment 11: Working with Smart Contract Events

- **Level 1:** Write a Solidity smart contract that emits events when certain actions occur (e.g., a value is updated). Deploy the contract in Remix and trigger the actions. Observe the emitted events in the Remix console.
- **Level 2:** Modify the previous contract to include indexed event parameters. Write a simple web3.js script (or use the Remix event listener) to filter and listen for specific events based on the indexed parameters.

Experiment 12: Understanding Smart Contract Security - Common Vulnerabilities (Part 1)

- **Level 1:** Study a simple smart contract with a known vulnerability (e.g., integer overflow/underflow - using an older Solidity version). Deploy the contract in Remix and attempt to exploit the vulnerability through function calls.
- **Level 2:** Research and demonstrate another common smart contract vulnerability (e.g., reentrancy - using a simplified example). Write a vulnerable contract and a separate "attacker" contract to exploit it on a local test network.

Experiment 13: Understanding Smart Contract Security - Common Vulnerabilities (Part 2)

- **Level 1:** Analyze a smart contract with access control implemented using onlyOwner modifier. Deploy the contract and attempt to call restricted functions from a non-owner account.
- **Level 2:** Explore the concept of gas limits and denial-of-service (DoS) attacks in smart contracts. Write a contract that could be susceptible to a simple gas-based DoS attack and demonstrate how it can be exploited.

Experiment 14: Interacting with Standard ERC-20 Tokens

- **Level 1:** Deploy a standard ERC-20 token contract (using OpenZeppelin library in Remix or a pre-written contract) on a local test network. Interact with the token contract's functions (e.g., totalSupply, balanceOf, transfer) using Remix.
- **Level 2:** Write a simple Solidity smart contract that interacts with the deployed ERC-20 token contract. Implement a function in your contract that allows users to spend a certain amount of the deployed tokens (requiring approval).

Experiment 15: Introduction to Truffle Framework

- **Level 1:** Install Truffle and Node.js. Create a new Truffle project. Understand the basic directory structure of a Truffle project (contracts, migrations, test). Compile a simple Solidity contract using Truffle commands.
- **Level 2:** Write a simple test case for your smart contract using Truffle's testing framework (Chai and Mocha). Run the tests to ensure the contract functions as expected. Deploy your compiled contract to a local Ganache network using Truffle migrations.

Course Code: **CSE3526**	**Course Title: Embedded & Decentralized Finance** **Type of Course:** **Theory & Lab Integrated**		**L-T-P-C**	2	0	2	3
Version No.		1.0				
Course Pre-requisites	•	FIN1001				
Anti-requisites		NIL				
Course Description		This course explores the integration of financial services into digital platforms through Embedded Finance and DeFi. It covers APIs, open banking, blockchain fundamentals, smart contracts, and cryptocurrencies. Students learn about decentralized platforms like DEXs, DAOs, and lending protocols. It highlights legal, security, and regulatory aspects, along with real-world case studies. The course also examines the convergence of DeFi with traditional finance and Web3 applications.				
Course Object		The objective of the course is to familiarize the learners with the concepts of Embedded and Decentralized Finance and attain **Skill Development** through **Experiential Learning** techniques.				
Course Out Comes		On successful completion of the course the students shall be able to: **CO1:** Understand the principles and applications of Embedded Finance and Decentralized Finance. (Understand) **CO2:** Explore blockchain technologies and smart contracts enabling DeFi. (Apply) **CO3**: Analyze the architecture and key components of Decentralized Finance (Apply) **CO4:** Evaluate security risks, vulnerabilities, and legal considerations in DeFi system. (Apply) **CO5:** Explain APIs, embedded wallets, and DeFi tools to develop basic solutions (Understand)				
Course Content:						
Module 1	Introduction to Embedded Finance	Assignment				**5L+6P Sessions**
	Topics: Concept, Scope, and Evolution, APIs, Open Banking, and BaaS, Embedded Lending, Payments, Insurance, Case Studies: BNPL, Embedded Investment Platforms					
Module 2	Blockchain & Cryptocurrency Foundations	Assignment				**7L+6P Sessions**
	Topics: Blockchain Basics: Hashing, Consensus Mechanisms, Smart Contracts (Ethereum focus), Cryptocurrencies, Wallets, and Gas Fees, Public vs Private Blockchains					

Module 3		Decentralized Finance	Assignment			**6L+6PSessions**
	Topics: Concept, Benefits and Risks Associated with DeFi, Centralized vs Decentralized finance, DeFi Projects, DeFi future trends.					
Module 4		Risk Management and Regulation	Assignment			**6L+6P Sessions**
	Topics: Security Threats in DeFi: Smart Contract Bugs, Flash Loans, Legal and Compliance Issues: AML, KYC, Regulatory Perspectives (India, US, EU), Notable DeFi Hacks & Case Studies .					
Module 5		Future of Embedded and Decentralized Finance				**6L+6P**
	DeFi 2.0, Interoperability, Layer-2 Scaling, Real-World Asset Tokenization, Central Bank Digital Currencies (CBDCs), Web3 Integration and Embedded Wallets					
	Project work/Assignment:					
1.	2. **Assignment 1 on (Module 1 and Module 2)** 3. **Assignment 2 on (Module 3,4 and Module 5)**					
	Text Book 1) Scarlett Sieber and Sophie Guibaud, "The Embedded Finance Handbook" , Wiley, 2023. 2) Campbell R. Harvey, "DeFi and the Future of Finance", Wiley, 2021. 3) Imran Bashir , "Mastering Blockchain", Packt, 2022.					
	References **1. Antony Lewis, "The Basics of Bitcoins and Blockchains", Mango Publishing, 2021** **2. Turban Rainer Potter, Information Technology, John Wiely& Sons Inc,2012.** **3..** **Web Resources** W1. https://medium.com/search?q=decentralized+exchange W2. https://thefinancialbrand.com/111080/evolution-future-digital-banking-baastransformation/					
	Lab Experiments **Experiment 1: Exploring Blockchain Basics** **Objective: Simulate a blockchain transaction and analyze block structure.** **Experiment 2: Create a Smart Contract for Token Transfer (ERC-20)** **Objective: Deploy a basic ERC-20 token on a local blockchain using Remix.**					

	Experiment 3: Build an Embedded Payment Flow using Stripe API **Objective: Integrate an embedded payment gateway using Stripe's API.** **Experiment 4: Create and Test a Decentralized Lending Contract** **Objective: Build a simple DeFi lending smart contract with collateral logic.** **Experiment 5: Use a Decentralized Exchange (DEX)** **Objective: Swap tokens using Uniswap on testnet or via a demo platform.** **Experiment 6: Wallet Integration and Transaction Monitoring** **Objective: Build a Web3 app that connects MetaMask and shows wallet balance.** **Experiment 7: Implement a DAO Voting Mechanism** **Objective: Build and deploy a basic voting contract simulating DAO governance.** **Experiment 8: Explore Decentralized Insurance Use Case** **Objective: Create a smart contract for crop/weather-based insurance.** **Experiment 10: Analyze a Real DeFi Protocol** **Objective: Study the architecture and working of Aave/Compound/Sushiswap.**
	Topics relevant to development of "Employability": Real-world usage of APIs in FinTech, Building financial products with embedded APIs **Topics relevant to "PROFESSIONAL ETHICS":** Ethical handling of user data in embedded finance platform

<table>
<tr><td>Course Code:
CS3527</td><td colspan="2">Course Title: Financial and Capital Markets

Type of Course:
Theory & Lab Integrated</td><td>L-T-P-C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td></td><td colspan="6">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td>•</td><td colspan="6">FIN1001</td></tr>
<tr><td>Anti-requisites</td><td></td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td></td><td colspan="6">This course provides a comprehensive overview of global financial markets, asset classes, and investment instruments. It covers the structure and functioning of various markets including money, equity, debt, derivatives, forex, and commodities. Students will learn about the roles of different participants, trading mechanisms, corporate actions, and market instruments such as bonds, mutual funds, and structured products. The course also delves into equity capital raising, trade life cycles, and global fund structures including open-ended and closed-ended investment vehicles. Emphasis is placed on both theoretical concepts and practical knowledge of global financial systems.</td></tr>
<tr><td>Course Object</td><td></td><td colspan="6">The objective of the course is to familiarize the learners with the concepts of Embedded and Decentralized Finance and attain Skill Development through Experiential Learning techniques.</td></tr>
<tr><td>Course Out Comes</td><td></td><td colspan="6">On successful completion of the course the students shall be able to:
CO1: Understand the evolution of banking technology and its impact on modern banking operations.(Understand)
CO2: Explain the role of delivery channels like ATMs, mobile banking, and UPI in digital banking systems.(Apply)
CO3: Evaluate the effectiveness of risk, treasury, and data center management in banking operations.(Apply)
CO4: Analyze the functioning of centralized banking and payment systems like RTGS, NEFT, and SFMS (Apply)
CO5: Understand blockchain, cryptocurrency concepts, and analyze recent core banking software. (Understand)</td></tr>
<tr><td>Course Content:</td><td></td><td colspan="6"></td></tr>
<tr><td>Module 1</td><td colspan="2">Overview of Financial Markets And Assets Classes</td><td>Assignment</td><td></td><td colspan="2"></td><td>10 Sessions</td></tr>
<tr><td></td><td colspan="7"><u>Topics:</u>

Cash and Money Markets, Bond markets, Foreign Exchange Markets, Equities Markets, Indices and Stocks, Derivatives Markets, Products and Settlement, Commodities Markets and</td></tr>
</table>

	Products, Saving and Investment Products, Mutual Fund and other Investment Products .					
Module 2	Global Equities Markets and Instruments	Assignment				**19 Sessions**
	Topics: Introduction to Equity Market-Introduction to Capital Markets, Equity Capital Markets, Raising Equity Through IPO, Raising Equity Through Private Sources, Equity buybacks, de-listing and reversion to a 'private' company. Equity Instruments & their characteristics-Stock Prices and Corporate Actions, Preference Shares, Depository Receipts, Rights Issues & Warrants, Convertibles, Equity Structured Products. Participants in the Equity Markets-Introduction and Role of the Buy Side, Buy Side Participants, Introduction and Role of Sell Side. Services and Participants in the Sell Side, Market Makers. Types of Equity Markets-Exchanges and Indices in the Equity Markets, Indices and their roles, Understand the difference between exchange and OTC markets, Types of weighted index, other indices and global indices, Electronic and Hybrid Markets and Order and Quote Driven Markets, Global Equity Markets. Trading of Equity Instruments-Equity Investments and its benefits and risks, Stock Quotations, Delivery or cash trading, Long and short positions, Leverage and Margin, Investing, trading and hedging, Placing Orders-limit orders, stop loss orders and GTD/GTC orders, Online and Offline Trading , Introduction to Trade Life Cycle, Clearing and Settlement					
Module 3	Global Foreign Exchange Markets and Instruments	Assignment				**16 Sessions**
	Topics: Introduction to Forex Market-What is foreign exchange market, Functions and purposes of the FX market, Introduction to types of Foreign Exchange Market. Participants in the foreign exchange market-Consumers & Travelers, Businesses, Investors & speculators, Commercial & Investment Banks, Government & Central Banks. Theories governing foreign exchange-Interest rate parity, Purchasing power parity, Nominal v/s real exchange rates, etc. Spot Market-Market organization, Quotation conventions, Direct and indirect prices, Cross rates, Value of a pip, Interpreting news and economic statistics, Delivery and operations. Forward Forex Market-Outright forward and swap deals, Relation between spot & forward markets, Quoting forward rates, Quoting swap points, Forward discounts and premiums, Forward forward transactions					
Module 4	Global Fixed Income (Bond) Markets and Instruments	Assignment				**8 Sessions**
	Topics: Overview of Debt Capital Markets-Characteristics of Debt Capital Markets, The differences between equity and debt products, The differences between loans and bonds, Hybrid securities, Securitization. Bond-An Introduction- Bond definition, Bond Issuer & Bond Investor, Types of bond, Bond characteristics, Zero Coupon Bond, Price/yield relationship, Government bond markets, The Eurobond market .					
Module 5	Global Funds					**7 Sessions**
	ntroduction -Potential advantages and disadvantages of collective investment, Difference between active and passive management .Open-Ended/Mutual Funds-Characteristics and different types of open-ended fund / mutual fund: • US • Europe , Purpose and principal features of the Undertakings for Collective Investment in Transferable Securities (UCITS) directive in European markets .Closed Ended Investment Companies-Characteristics of closed-ended investment companies, share classes, Meaning					

	of the discounts and premiums in relation to the pricing of closed-ended investment companies , How closed-ended investment companies' shares are traded. Off shore and On-shore Global financial centers
	Project work/Assignment:
8.	**9. Assignment 1 on (Module 1 and Module 2)** 10. **Assignment 2 on (Module 3,4 and Module 5)**
	Text Book 5) Financial Markets and Institutions 7th Edition By Anthony Saunders and Marcia Cornett, Ninth Edition, McGraw Hill Education, 2024. 6) Mishkin, F. S., & Eakins, S. G. (2018) "*Financial Markets and Institutions"* (9th Edition). Pearson Education.
	References **1. Gordon, E. & Natarajan, K. (2022) *Financial Markets and Services* (Latest Edition). Himalaya Publishing House.** **2.**Bhole,L.M.&Mahakud,J.(2017) Financial Institutions and Markets: Structure, Growth, and Innovations (5th Edition). McGraw Hill Education**.** **Web Resources** W4. https://www.ibm.com/industries/banking-financial-markets/resources/omnichannelbanking-paper/ W5. https://thefinancialbrand.com/111080/evolution-future-digital-banking-baastransformation/
	Design and Develop the following Banking Software using the appropriate technologies: **■ Mobile Banking** **▪ Balance Enquiry ▪ Cheque book Request ▪ Stop Cheque ▪ Credit/Debit Notification ▪ Bill Payment** **■ Internet Banking** **▪ Electronic Funds Transfer ▪ Account Management ▪ Loan Application ▪ Registering of new bank services ▪ Customer Information Management** **■ATM system** **▪ Balance Enquiry ▪ Withdrawal ▪ Deposit ▪ Pin change ▪ Mini statement**
	Topics relevant to development of "Employability": Real time Data Analysis for Banking Technology. **Topics relevant to "PROFESSIONAL ETHICS":** Mobile, Internet Banking for Project Development.

Course Code: **CSE3528**	**Course Title: Blockchain Development and Programming** **Type of Course:** **Theory & Lab Integrated**	**L-T-P-C**	2	0	2	3
Version No.		1.0				
Course Pre-requisites	•	CBC2000				
Anti-requisites		NIL				
Course Description		This course provides a comprehensive introduction to the fundamental concepts and applications of blockchain technology. Students will explore the principles of decentralization, understand the mechanics of Bitcoin, delve into the world of smart contracts and alternative cryptocurrencies, and gain practical experience in developing and deploying smart contracts using industry-standard tools. The course emphasizes both the theoretical underpinnings and the practical implementation of blockchain solutions.				
Course Object		The objective of the course is to familiarize the learners with the concepts of Blockchain Development and Programming and attain **Skill Development** through **Experiential Learning** techniques.				
Course Out Comes		On successful completion of the course the students shall be able to: **CO1:** Understand blockchain and decentralization principles. (Understand) **CO2:** Analyze Bitcoin's operation and transactions. (Apply) **CO3**: Apply smart contract concepts and explore altcoins. (Apply) **CO4:** Utilize Truffle for smart contract development and deployment. (Apply)				
Course Content:						
Module 1	**Fundamentals of Blockchain and Decentralization**	Assignment				**6L+6P Sessions**
	Introduction to Blockchain: the growth of blockchain technology, the history of blockchain and Bitcoin, distributed systems, blockchain, consensus. Decentralization: decentralization using blockchain, methods of decentralization, routes to decentralization, blockchain and full ecosystem decentralization, pertinent terminology, platforms for decentralization, innovative trends.					
Module 2	**Bitcoin: The First Cryptocurrency**	Assignment				**7L+8P Sessions**
	Topics:					

	Introduction to Bitcoin: bitcoin — an overview, cryptographic keys, transactions, blockchain, mining. The Bitcoin Network and Payments: the bitcoin network, wallets, bitcoin payments, innovation in bitcoin, advanced protocols, bitcoin investment and buying and selling Bitcoin. Bitcoin Clients and APIs: bitcoin client installation, experimenting further with bitcoin-cli, bitcoin programming					
Module 3		**Beyond Bitcoin: Altcoins and Smart Contracts**	Assignment			**10L+8PSessions**
	Topics: Consensus Algorithms: introducing the consensus problem, analysis and design, classification, algorithms, choosing an algorithm. Alternative Coins: introducing altcoins, theoretical foundations, difficulty adjustment and retargeting algorithms, bitcoin limitations, extended protocols on top of bitcoin, development of altcoins, Initial Coin Offerings (ICOs). Smart Contracts: history, definition, ricardian contracts, smart contract templates, oracles, deploying smart contracts, the DAO. .					
Module 4		**Ethereum and the Decentralized Web (Web3)**	Assignment			**7L+8P Sessions**
	Topics: Ethereum 101: ethereum – an overview, the ethereum network, components of the ethereum ecosystem, Ethereum Virtual Machine (EVM), smart contracts, ethereum development environment. Further Ethereum: blocks and blockchain, wallets and client software, nodes and miners, APIs, tools, and DApps, supporting protocols, programming languages. Introducing Web3: contract deployment, exploring Web3 with Geth					
	Project work/Assignment:					
11.	**12.** **Assignment 1 on (Module 1 and Module 2)** 13. **Assignment 2 on (Module 3 and Module 4)**					
	Text Books 1. Banafa, A. (2024). *Blockchain technology and applications*. River Publishers. 2. Ramachandran, M. (2025). *Blockchain engineering: Secure, sustainable frameworks for healthcare applications*. Springer. 3. Tanwar, S. (2022). *Blockchain technology: From theory to practice*. Springer. 4. Vyas, S., Shukla, V. K., Gupta, S., & Prasad, A. (Eds.). (2022). *Blockchain technology: Exploring opportunities, challenges, and applications*. CRC Press.					

	References 1. Chuen, D. L. K. (Ed.). (2024). *Handbook of digital currency: Bitcoin, innovation, financial instruments and big data* (2nd ed.). Academic Press. 2. Idrees, S. M. & Nowostawski, M. (Eds.). (2023). *Transformations through blockchain technology: The new digital revolution*. Springer. 3. Jena, A. K., Panda, S. K., & Swain, S. K. (Eds.). (2022). *Blockchain technology: Applications and challenges* (Vol. 203). Springer. 4. Maleh, Y., Zhang, J., & Hansali, A. (2024). *Advances in emerging financial technology and digital money*. Routledge. 5. Rahman, H. (Ed.). (2025). *Blockchain technology applications in knowledge management*. IGI Global **Web Resources** W1. Blockgeeks. Retrieved from https://www.google.com/search?q=blockgeeks.com W2. Bitcoin.org. Retrieved from https://bitcoin.org/ W3. CoinDesk. Retrieved from https://www.coindesk.com/ W4. Ethereum.org. Retrieved from https://ethereum.org/ W5. Investopedia. Retrieved from https://www.investopedia.com/ W6. Medium. Retrieved from https://medium.com/ W7. Solidity Documentation. Retrieved from https://docs.soliditylang.org/ W8. Truffle Suite Documentation. Retrieved from https://trufflesuite.com/docs W9. Web3.js Documentation. Retrieved from https://web3js.readthedocs.io/ W10. GitHub. Retrieved from https://github.com/

	Experiment 1: Integrated Development Environments (IDEs) for Smart Contracts • **Level 1:** Explore the features and interface of Remix IDE. Deploy a simple "Hello World" smart contract on the in-browser JavaScript VM. Observe the transaction details and contract interaction options. • **Level 2:** Install and configure MetaMask browser extension. Connect MetaMask to the Remix IDE. Deploy the same "Hello World" contract to the Ganache private network via MetaMask. Examine the transaction process in both Remix and MetaMask. **Experiment 2: MetaMask in a Private Network** • **Level 1:** Set up a local Ganache private network. Add a custom network in MetaMask, configuring the RPC URL and Chain ID to connect to your Ganache instance. Create a new account in MetaMask and observe its balance. • **Level 2:** Deploy a simple token contract (e.g., ERC-20 minimal) using Remix IDE and MetaMask on your private Ganache network. Transfer some tokens between the accounts you created in MetaMask and observe the balance changes. **Experiment 3: Smart Contract with Solidity - Basic Data Types and Structures** • **Level 1:** Write a Solidity smart contract that declares and initializes variables of different basic data types (uint, string, bool, address). Implement functions to read and modify these variables. Deploy and interact with the contract in Remix IDE. • **Level 2:** Create a Solidity smart contract that utilizes structs and arrays. Implement functions to add, retrieve, and update elements within these data structures. Deploy and test the contract with various inputs in Remix IDE. **Experiment 4: Smart Contract with Solidity - Control Flow and Functions** • **Level 1:** Write a Solidity smart contract that uses if-else statements and for loops within its functions. Implement a function that performs a simple calculation based on input parameters. Deploy and test the different control flow paths in Remix. • **Level 2:** Design and implement a Solidity smart contract with multiple functions, including internal and private functions. Demonstrate how these functions can be called and how visibility modifiers affect their accessibility. **Experiment 5: Contract Deployment** • **Level 1:** Deploy a pre-written simple smart contract (provided by the instructor) using MetaMask connected to the Ganache network. Observe the deployment transaction details (gas used, transaction hash, contract address). • **Level 2:** Explore different deployment parameters in Remix IDE (e.g., setting gas limit and gas price). Deploy the same contract multiple times with varying gas settings and analyze the impact on deployment cost and confirmation time in Ganache. **Experiment 6: MetaMask and Remix IDE Interaction** • **Level 1:** Deploy a simple counter smart contract using Remix IDE on the Ganache network via MetaMask. Use the Remix interface to call the contract's functions

	(e.g., increment, decrement, get count) and observe the state changes reflected in both Remix and MetaMask (balance changes for transactions). • **Level 2:** Deploy a more complex smart contract (e.g., a simple voting contract) using Remix and MetaMask. Interact with the contract through MetaMask's custom interaction interface (sending transactions to specific functions with appropriate arguments). **Experiment 7: Use of Geth - Installation and Account Management** • **Level 1:** Install the Geth Ethereum client on your local machine. Use Geth commands to create new Ethereum accounts and list the available accounts. Observe the keystore directory where private keys are stored. • **Level 2:** Use Geth commands to export and import Ethereum account private keys. Understand the security implications of managing private keys. Connect the Geth console to a running private network (e.g., Ganache or a custom Geth network). **Experiment 8: Genesis Block Creation in Geth** • **Level 1:** Understand the structure of a Genesis Block JSON file. Modify a sample Genesis Block configuration (e.g., changing the initial coin distribution). Initialize a new Geth data directory using this modified Genesis Block. • **Level 2:** Create a custom Genesis Block for a private Ethereum network with specific pre-allocated accounts, custom gas limit, and difficulty. Start a Geth node using this custom Genesis Block and connect to it using the Geth console. **Experiment 9: Interacting with a Private Geth Network** • **Level 1:** Start a Geth node using a previously initialized data directory. Use the Geth console to check the node's peer count and block number. Create a transaction to send Ether between two accounts within your private network using Geth commands. • **Level 2:** Deploy a simple smart contract to your private Geth network using the Geth console and web3.js (or similar library). Interact with the deployed contract's functions using the Geth console. **Experiment 10: Exploring Ethereum Transaction Structure** • **Level 1:** Send a transaction (Ether transfer) using MetaMask on the Ropsten test network. Examine the transaction details on a block explorer (e.g., Etherscan for Ropsten) and identify key fields like to, from, value, gas limit, gas price, and nonce. • **Level 2:** Construct and sign a raw Ethereum transaction using web3.js (or similar library) without relying on MetaMask. Broadcast this signed transaction to a test network (e.g., Ropsten) and analyze its details on a block explorer. **Experiment 11: Working with Smart Contract Events** • **Level 1:** Write a Solidity smart contract that emits events when certain actions occur (e.g., a value is updated). Deploy the contract in Remix and trigger the actions. Observe the emitted events in the Remix console.

- **Level 2:** Modify the previous contract to include indexed event parameters. Write a simple web3.js script (or use the Remix event listener) to filter and listen for specific

  events based on the indexed parameters.

Experiment 12: Understanding Smart Contract Security - Common Vulnerabilities (Part 1)

- **Level 1:** Study a simple smart contract with a known vulnerability (e.g., integer overflow/underflow - using an older Solidity version). Deploy the contract in Remix and attempt to exploit the vulnerability through function calls.
- **Level 2:** Research and demonstrate another common smart contract vulnerability (e.g., reentrancy - using a simplified example). Write a vulnerable contract and a separate "attacker" contract to exploit it on a local test network.

Experiment 13: Understanding Smart Contract Security - Common Vulnerabilities (Part 2)

- **Level 1:** Analyze a smart contract with access control implemented using onlyOwner modifier. Deploy the contract and attempt to call restricted functions from a non-owner account.
- **Level 2:** Explore the concept of gas limits and denial-of-service (DoS) attacks in smart contracts. Write a contract that could be susceptible to a simple gas-based DoS attack and demonstrate how it can be exploited.

Experiment 14: Interacting with Standard ERC-20 Tokens

- **Level 1:** Deploy a standard ERC-20 token contract (using OpenZeppelin library in Remix or a pre-written contract) on a local test network. Interact with the token contract's functions (e.g., totalSupply, balanceOf, transfer) using Remix.
- **Level 2:** Write a simple Solidity smart contract that interacts with the deployed ERC-20 token contract. Implement a function in your contract that allows users to spend a certain amount of the deployed tokens (requiring approval).

Experiment 15: Introduction to Truffle Framework

- **Level 1:** Install Truffle and Node.js. Create a new Truffle project. Understand the basic directory structure of a Truffle project (contracts, migrations, test). Compile a simple Solidity contract using Truffle commands.
- **Level 2:** Write a simple test case for your smart contract using Truffle's testing framework (Chai and Mocha). Run the tests to ensure the contract functions as expected. Deploy your compiled contract to a local Ganache network using Truffle migrations.

	Topics relevant to development of "Employability": Hands-on experience with blockchain platforms, smart contract development, and decentralized applications **Topics relevant to "PROFESSIONAL ETHICS":** Understanding ethical considerations in blockchain use, such as data privacy, transparency, and responsible innovation, promotes integrity in decentralized systems

Course Code: **CSE3529**	**Course Title: Statistics and Data Analysis for Finance** **Type of Course: Theory & Lab Integrated**	**L-T-P-C**	2	0	2	3
Version No.		1.0				
Course Pre-requisites	•	FIN1001				
Anti-requisites		NIL				
Course Description		This course provides a comprehensive understanding of the theory and practice of data analysis in the Accounting and wider business domains. Students who complete this course should develop the skills to apply and interpret data-based initiatives that address real-world problems across many financial activities such as financial accounting, management accounting, taxation, auditing and corporate finance.				
Course Object		The objective of the course is to familiarize the learners with the concepts of Statistics and Data Analysis for Finance and attain **Skill Development** through **Experiential Learning** techniques.				
Course Out Comes		On successful completion of the course the students shall be able to: **CO1:** Understand the strategic processes, benefits and challenges. (Understand) **CO2:** Apply preprocessing techniques to business datasets (Apply) **CO3**: Apply statistical techniques to the datasets . (Apply) **CO4:** Analyse data protection, data privacy and other ethical issues. (Apply)				
Course Content:						

Module 1	**Strategic Data Management in in Finance**	Assignment			**6L+6P Sessions**
Topics: Challenges in Human Decision Making, Data Analytics Processes - Introduction to Process Models, Financial Big Data for Competitive Advantage, Strategic Data Management in Finance, Management Challenges in Data Driven Environments, Developing a Data Driven Culture					
Module 2	**Exploratory Data Analysis**	Assignment			**7L+8P Sessions**
Topics: Exploratory Data Analysis, Data Preparation – Normalization, Binning, Sampling Descriptive Statistics, Data Quality Issues - Missing Values, Outliers, Visualizing Relationships Between Features, Measuring Covariance and Correlation, Data Distributions and Confidence Intervals, • Simple Linear Regression, Correlation Coefficient, Calculation of Regression parameters					
Module 3	**Statistics for Big data**	Assignment			**10L+8PSessions**

	Topics: Effect size, Statistical power and sample size, Effect of Variation, Hypothesis testing Interpret outputs from statistical software to analyse patterns in accounting data for signalling unexpected fluctuations e.g. Fraud Analysis, detecting anomaly transactions etc.. , Time Series basics, Decomposition of Time Series, Seasonality, Linear Trend models, Smoothing models, Interpret outputs from statistical software to support prediction of accounting data.					
Module 4	**Dimension Reduction & Data Ethics**	Assignment			**7L+8P Sessions**	
	Topics: Factor Analysis, Principal Component Analysis (PCA), Interpret outputs from statistical software to analyse reasons behind fluctuations in accounting data e.g. defining cost drivers in Activity Based Costing etc, Data Ethics in Finance, Data Legislation GDPR, Data and Statistical reporting					
	Project work/Assignment:					
14.	**15. Assignment 1 on (Module 1 and Module 2)** 16. **Assignment 2 on (Module 3 and Module 4)**					
	Text Books 5. Foster Provost, Tom Fawcett, Data Science for Business, O'Reilly Media, 2023. 6. Wes McKinney, Python for Data Analysis, O'Reilly Media (2nd Ed.), 2022. 7. Peter J. Brockwell, Richard A., Introduction to Time series and Forecasting, Springer, 2023. 8. W. Gregory Voss, Hélène J. Lefebvre, Data Ethics in the Digital Age, Springer, 2023.					
	References 6. Mark J. Bennett, Dirk L. Hugen, Financial Analytics with R: Building a Laptop Laboratory for Data Science, Cambridge University Press, 2021. 7. Thomas H. Davenport, Analytics at Work: Smarter Decisions, Better Results, Harvard Business Press, 2022. 8. Peter Bruce, Andrew Bruce, Peter Gedeck, Practical Statistics for Data Scientists, O'Reilly Media (2nd Ed., 2023. 9. Satish Kumar, Principles and Practice of Multi-dimensional Data Analysis, Wiley, 2022. **Web Resources** W1. https://link.springer.com/book/10.1007%2F978-3-030-01279-3 W2. https://link.springer.com/book/10.1007%2F978-1-4939-2122-5 W3. https://link.springer.com/book/10.1007%2F978-3-319-55444-0					
	Strategic Data Management in Finance • Identify and discuss the challenges in Human Decision Making, particularly relating to large datasets. • Discuss the benefits and challenges of utilising Process Models to manage Finance Data Analysis projects • Identify and discuss the strategic benefits to be derived from Financial Big Data					

	. Identify and discuss the management challenges in leveraging the benefits of Big Data for strategic competitive advantage Exploratory Data Analysis • Discuss the process of data cleaning and preparation – e.g. Normalization, Binning, Sampling • Apply and Evaluate key descriptive statistics, including Covariance and Correlation, in a data set for large business datasets • Discuss solutions to overcome data quality issues in Data Analysis projects - missing values, outliers etc. Apply and Evaluate methods for visualizing relationships between features Statistics for Big data • Explain relationships between sample size, effect size, statistical power • Describe and Evaluate measures of variation for large datasets • Describe hypothesis testing and evaluate outputs from hypothesis tests performed using software such as Excel, R and Python etc. Interpret outputs from statistical software to analyse patterns in accounting data for signaling unexpected fluctuations - fraud analysis, detection of anomaly transactions etc. Dimension reduction • Explain PCA and factor analysis and discuss its uses in the analysis of large financial datasets Interpret outputs from statistical software to analyse reasons behind fluctuations in accounting data e.g. defining cost drivers in Activity Based Costing etc. Data Ethics & Legal Considerations • Identify and discuss the ethical issues surrounding the use of data analytics in finance • Demonstrate an understanding of Data Legislation GDPR and its impact on data analytics Discuss the societal impacts of the increasing use of Data Analysis techniques in Finance and Business
	Topics relevant to development of "Employability": Hands-on experience with PCA, Hypothesis test. **Topics relevant to "PROFESSIONAL ETHICS":** Understanding ethical considerations and legal considerations

<table>
<tr><td>ourse Code:
CSE3530</td><td>Course Title:
Financial Regulations and Compliances
Type of Course: Theory</td><td>L- T- P- C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="6">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="6">FIN1001</td></tr>
<tr><td>Anti-requisites</td><td colspan="6">NIL</td></tr>
<tr><td>Course Description</td><td colspan="6">This course provides a comprehensive understanding of the evolving landscape of financial technology (FinTech) regulations and the growing importance of Regulatory Technology (RegTech). It examines the historical development of FinTech regulation, analyzes the specific regulatory framework governing FinTech innovations in India, and explores the application of regulatory sandboxes. Furthermore, the course evaluates the challenges associated with the adoption and implementation of RegTech solutions in the financial services industry. By the end of this course, students will gain a strong foundation in the key regulatory considerations shaping the FinTech ecosystem and the role of technology in navigating this complex environment.</td></tr>
<tr><td>Course Objective</td><td colspan="6">The objective of this course is to equip learners with a practical understanding of FinTech regulations and RegTech, and to develop their analytical and problem-solving skills through active participation in learning activities, thereby enhancing their employability in the evolving financial technology sector.</td></tr>
<tr><td>Course Outcomes</td><td colspan="6">On successful completion of the course the students shall be able to:
CO1: Understand the evolution of FinTech regulation and the emergence of RegTech.
CO2: Explain the regulations governing FinTech within the Indian context.
CO3: Describe the purpose and function of regulatory sandboxes.
CO4: Analyze the challenges in adopting RegTech solutions.</td></tr>
<tr><td>Course Content:</td><td colspan="6"></td></tr>
<tr><td>Module 1</td><td>Introduction to FinTech regulation and RegTech</td><td>Assignment</td><td colspan="2">RegTech Ecosystem</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="7">FinTech Regulation, Evolution of RegTech- RegTech 1.0, RegTech 2.0, RegTech 3.0, RegTech ecosystem-Financial institutions, start-ups, and regulators. The future of Regtech and the technologies impacting it</td></tr>
<tr><td>Module 2</td><td>Regulations governing FinTech in India</td><td>Assignment</td><td colspan="2">Regulations</td><td colspan="2">10 Sessions</td></tr>
<tr><td colspan="7">Regulation of mobile money, Regulation of smart contracts, Regulation of Robo-Advisory services, legal and regulatory implications of cryptocurrencies, Payment and Settlements System Act 2007,</td></tr>
</table>

Master direction on Issuance and operation of prepaid payments instruments. NPCI guidelines governing UPI payments, Master direction-NBFC, Guidelines regulating P2P lending platforms, payment aggregators/intermediaries, payment banks, Anti money-laundering regulations, Data privacy and protection.				
Module 3	Regulatory Sandboxes	Assignment	Regulatory Sandboxes	**9 Sessions**
Introduction, what is regulatory sandbox-Covered FinTech products and eligible participants, parameters, regulatory safe harbour, Post sandbox engagement. Benefits-Participant-regulator dialogue, reduced time and cost of market penetration, stronger appeal to stakeholders, market signalling. Shortcoming-Multi-tiered regimes, Pre-judging innovative value, scalability, race to the bottom. Regulatory sandbox in India, China, USA, Europe and other countries.				
Module 4	Challenges, Future and Use Case	Assignment	Complaince	**9 Sessions**
Risks and challenges of RegTech adoption-Procurement and approval process, Preference for large and established players, Fragmented markets, Regulatory uncertainty, Concentration risk, Data protection security and cyber threats. Future Trends in RegTech-Quantitative Regulation, Machine readable regulation, Agile Regulation, Regulatory Sandboxes, International regulation. Compliance, Identity management and control, risk management, Regulatory reporting, Transaction monitoring, Trading in markets				
Targeted Application & Tools that can be used:				

Text Book(s):

1. Madir, J. (Ed.). (2024). *FinTech: Law and Regulation* (3rd ed.). Edward Elgar Publishing.
2. Securities and Exchange Board of India. (2014). *Consultation paper on crowdfunding in India*. https://www.sebi.gov.in/sebi_data/attachdocs/1403005615257.pdf
3. Reserve Bank of India. (2025, April 9). *Enabling Framework for Regulatory Sandbox* https://www.rbi.org.in/Scripts/PublicationReportDetails.aspx?UrlPage=&ID=938

Reference(s):

1. Ren, D. (2018). Tightening regulations make FinTechs easy takeover targets for banks stepping up digitalisation drive. *SCMP*. Retrieved from https://www.scmp.com/business/companies/article/2159718/tightening-regulations-make-fintechs-easy-takeover-targets-banks
2. Zetzsche, D. A., Buckley, R. P., Arner, D. W., & Barberis, J. N. (2017). *From FinTech to TechFin: The regulatory challenges of data-driven finance* (University of Hong Kong Faculty of Law Research Paper No. 2017/007). http://dx.doi.org/10.2139/ssrn.2959925
3. Magnuson, W. J. (2017). *Regulating Fintech* (Texas A&M University School of Law Legal Studies Research Paper No. 17-55). Retrieved from https://ssrn.com/abstract=3027525
4. Sethi, V. (n.d.). *Fintech & Regtech - your definitive guide on the convergence of finance, technology and regulation* (p. 3) [Kindle Edition]. Max Krish Publishers.
5. Lui, A., & Ryder, N. (Eds.). (2023). *FinTech, Artificial Intelligence and the Law: Regulation and Crime Prevention*. Routledge.
6. Shrier, D. L. (Ed.). (2022). Global Fintech: Financial Innovation in the Connected World. MIT Press.
7. McGurk, B. KC., & Reichenbach, S. (2024). Financial Services Law and Distributed Ledger

Technology: Regulating Cryptoassets and Decentralised Finance. Edward Elgar Publishing.
8. Buckley, R. P., Arner, D. W., & Zetzsche, D. A. (2023). Fintech finance technology and regulation. Cambridge University Press.
9. Justin, M. S. M., et al. (Eds.). (2024). Examining Global Regulations During the Rise of Fintech. IGI Global.
10. El Dimachki, M. (2024). Fintech Regulation In Practice. Kogan Page.
11. Madir, J. (Ed.). (2024). FinTech: Law and Regulation (3rd ed.). Edward Elgar Publishing.
12. McGurk, B. KC., & Reichenbach, S. (2024). Financial Services Law and Distributed Ledger Technology: Regulating Cryptoassets and Decentralised Finance. Edward Elgar Publishing.
13. Risk Books. (2025). Regtech, Suptech and Beyond: Innovation in Financial Services.
14. Gupta, S., et al. (Eds.). (2024). Integrating RegTech Solutions for Industry 4.0. IGI Global.

Websites:
1. **Financial Conduct Authority.** (n.d.). *FinTech*. Retrieved from https://www.fca.org.uk/firms/innovation/fintech
2. **Securities and Exchange Board of India.** (n.d.). *Homepage*. Retrieved from https://www.sebi.gov.in/
3. **Reserve Bank of India.** (n.d.). *Homepage*. Retrieved from https://www.rbi.org.in/
4. **Financial Industry Regulatory Authority.** (n.d.). *FinTech*. Retrieved from https://www.finra.org/rules-guidance/key-topics/fintech
5. **International Financial Services Centres Authority.** (n.d.). *FinTech Hub*. Retrieved from https://ifsca.gov.in/FinTechHub2023/ifsca.gov.in/Pages/Contents/FinnTechHub.html
6. **Federal Trade Commission.** (n.d.). *Fintech*. Retrieved from https://www.ftc.gov/business-guidance/credit-finance/fintech
7. **Deloitte Luxembourg.** (n.d.). *Regtech Universe*. Retrieved from https://www.deloitte.com/lu/en/Industries/technology/analysis/regtech-companies-compliance.html
8. **Apiax.** (n.d.). *The Ultimate RegTech Guide*. Retrieved from https://www.apiax.com/resources/guides/regtech-guide/
9. **Ascent RegTech.** (n.d.). *Homepage*. Retrieved from https://www.ascentregtech.com/
10. **CUBE Global.** (n.d.). *Homepage*. Retrieved from https://cube.global/
11. **ACA Group.** (n.d.). *ComplianceAlpha RegTech Solutions*. Retrieved from https://www.acaglobal.com/our-solutions/compliancealpha
12. **FinTech Futures.** (n.d.). *RegTech Archives*. Retrieved from https://www.fintechfutures.com/category/regtech/

<table>
<tr><td>Course Code: CSE3423</td><td colspan="4">Course Title: Go Programming
Type of Course: Theory Only Course</td><td>L- T-P- C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Version No.</td><td></td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td></td><td colspan="8">CSE1502 Problem Solving using C</td></tr>
<tr><td>Anti-requisites</td><td></td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td></td><td colspan="8">Go, an open-source programming language developed by Google, is recognized for its expressiveness, conciseness, cleanliness, and efficiency. Its robust concurrency mechanisms simplify building programs that effectively utilize multicore and networked machines. Go compiles quickly to machine code, offering the convenience of garbage collection and the power of run-time reflection. It's a fast, statically typed, compiled language that provides the dynamic feel of an interpreted language. Its adoption is rapidly growing across industries, with prominent users like Dropbox and Uber. This course provides an introduction to the Go programming essentials through lectures and demonstrations. Topics covered include Go program structure; data types and control statements; composite types (arrays, slices, strings, runes, bytes, hash maps); functions; methods; garbage collection essentials (pointers, structs, interfaces); error handling; concurrency (goroutines and channels); packages (importing and creating custom packages); and applications of Go, including basic web development and statistical computations.</td></tr>
<tr><td>Course Objective</td><td></td><td colspan="8">The primary objective of this course is to familiarize learners with the core concepts of Go Programming and to enhance their employability skills by applying these concepts through practical problem-solving techniques.</td></tr>
<tr><td>Course Out Comes</td><td></td><td colspan="8">On successful completion of the course, students will be able to:
CO1: Identify primitive programming constructs in Go. (Knowledge)
CO2: Discuss composite data types and apply principles of modular programming. (Comprehension)
CO3: Implement garbage collection concepts using pointers, structs, interfaces, and modules in Go. (Application)
CO4: Apply concurrent programming techniques and testing methodologies to build various applications. (Application)</td></tr>
<tr><td>Course Content:</td><td></td><td colspan="8">---</td></tr>
<tr><td>Module 1</td><td colspan="2">Introduction to Go Programming Language</td><td>Assignment</td><td colspan="2">Data Collection/Interpretation</td><td></td><td colspan="3">10 Sessions</td></tr>
<tr><td colspan="10">Topics: [Knowledge]
Features and philosophy of the Go language; Installing and configuring the development environment (Go tools and Go Playground); Anatomy of a Go program; Basic types (numbers, booleans, strings, runes); Variables (declaration, zero values, naming conventions, type conversions, constants, multiple variable declarations); Introduction to packages and utilizing functions from standard libraries (e.g., fmt.Println); Reading user input; Control Structures (if, switch, for); Practical programming exercises applying control statements.</td></tr>
<tr><td>Module 2</td><td colspan="2">Composite types and functions</td><td>Assignment</td><td colspan="2">Data Collection/Interpretation</td><td></td><td colspan="3">9 Sessions</td></tr>
<tr><td colspan="10">Topics: [Comprehension]Composit
e types – arrays, slices (creation, manipulation, slices with overlapping storage), structs (definition, embedding, anonymous structs); Functions – declaring functions, defining parameters, returning multiple values, variadic functions; Programming exercises focused on designing and using composite types and functions effectively.</td></tr>
<tr><td>Module 3</td><td colspan="2">Pointers, Structs, Interfaces and modules</td><td>Quiz</td><td colspan="2">Case studies / Case let</td><td></td><td colspan="3">9 Sessions</td></tr>
</table>

Topics: **[Application]**
Pointers: understanding * (dereference) and & (address-of) operators, pointer types, passing pointers to functions; Go's Garbage Collector – historical overview and its role in Go's memory management; Methods (value and pointer receivers) and Interfaces (defining and implementing interfaces, type assertion, type switches); Modules and packages – importing standard and third-party packages, creating and managing custom packages; Programming exercises.

Module 4	**Concurrency and Applications**	Quiz	Case studies / Case let		**7 Sessions**

Topics: **[Application]**
Introduction to Concurrency using Goroutines; Managing multiple goroutines; Channels – creating channels, unbuffered vs. buffered channels, sending and receiving operations on channels, select statement; Testing in Go – writing unit tests, running tests with go test command; Overview of Core Packages for string manipulation, containers (e.g., list, ring), and lists; Introduction to Writing Web Applications using Go's net/http package (basic HTTP server, routing); Basic Statistical Computations (e.g., calculating mean, median); Basic concepts of histogram plotting; Fundamentals of Encryption and Decryption with Go.

Targeted Application & Tools that can be used:
Go Playground: https://go.dev/play/ (Online environment for quick Go code experimentation)
Go Installation: https://go.dev/doc/install (Official guide for installing Go on various platforms)

Project work/Assignment:

Students can undertake a significant project that requires them to apply and integrate various Go programming concepts learned throughout the course. This project could involve:

- Developing a **command-line utility** for data processing (e.g., CSV/JSON parser and analyzer).
- Building a **simple web service/API** using Go's standard library or a lightweight framework.
- Implementing a **concurrent application** (e.g., a multi-threaded file downloader, a simple chat application).
- Creating a program that performs **data analysis and basic visualization** using Go's statistical capabilities, potentially generating data for machine learning models.

The project will emphasize **clean code practices, modular design, robust error handling, and comprehensive testing**, fostering practical application of Go in real-world scenarios.

Text Book

T1: John Badner, "Learning Go: An Idiomatic Approach to Real World Go Programming", O'Reilly, California, 2021.

References

R1: Alan A.A. Donovan and Brian W. Kernighan, "The Go Programming Language", Pearson Education, India, 2016.

R2: Tsoukalos M. Mastering Go: Create Golang production applications using network libraries, concurrency, machine learning, and advanced data structures. Packt Publishing Ltd; 2019 Aug 29.

Web resources:
- Go Language Programs: https://www.golangprograms.com/go-language.html
- EBSCO database of Presidency University: https://puniversity.informaticsglobal.com/login
- Official Go Documentation: https://go.dev/doc/
- Go Playground: https://go.dev/play/
- Go Download and Install: https://go.dev/doc/install

Online tool for program execution:
- GO Play Ground - https://go.dev/play/
- Download and install: https://go.dev/doc/install

	Topics relevant to development of “Employability”: This course directly contributes to Skill Development by focusing on Go Programming basics for enhancing Employability Skills through robust Problem Solving methodologies. This is reinforced through the assessment components mentioned in the course handout and the detailed lab experiments.

List of Lab Experiments (15 Experiments)

Each experiment is designed with a **low-level problem** to solidify fundamental understanding and a **higher-level problem** to challenge students with more complex application scenarios, building upon the theoretical knowledge.

Experiment 1: Go Fundamentals - Setup & Basic I/O

- **Low Level:** Set up the Go development environment. Write a Go program to print "My first Go program!" to the console.
- **Higher Level:** Create a program that prompts the user for their name and favorite number, then prints a personalized message incorporating both inputs. Demonstrate basic type conversion for the number.

Experiment 2: Variables, Constants, and Basic Data Types

- **Low Level:** Declare and initialize variables of int, float64, bool, and string types. Print their values and demonstrate Go's type inference.
- Higher Level: Write a program that declares constants for mathematical values (e.g., Pi, e). Calculate the circumference of a circle and the area of a rectangle using user-provided dimensions, demonstrating type safety.

Experiment 3: Control Flow - if-else and switch

- **Low Level:** Implement a program to determine if a given integer is positive, negative, or zero using nested if-else statements.
- **Higher Level:** Create a program that simulates a simple grading system. Take a student's score (0-100) as input and assign a letter grade (A, B, C, D, F) using a switch statement with case ranges.

Experiment 4: Looping Constructs - for

- **Low Level:** Print all odd numbers from 1 to 20 using a for loop with a continue statement.
- **Higher Level:** Write a program that calculates the sum of digits of a given integer using a for loop. For example, if the input is 123, the output should be 6.

Experiment 5: Arrays and Slices

- **Low Level:** Declare an array of 5 floating-point numbers. Calculate and print their sum.
- **Higher Level:** Create a dynamic slice of strings. Implement functions to: 1) add a new string, 2) remove a string by its value, and 3) print the current elements, length, and capacity of the slice after each operation.

Experiment 6: Maps (Hash Maps)

- **Low Level:** Create a map to store the capital cities of three countries. Access and print the capital of a specific country.
- **Higher Level:** Develop a program that simulates a simple inventory system. Use a map to store product names as keys and their quantities as values. Implement functions to add new products, update quantities, and check stock levels.

Experiment 7: Functions - Multiple Returns & Variadic Functions

- **Low Level:** Write a function divide(numerator, denominator float64) that returns both the result of division and a boolean indicating if division was successful (to handle division by zero).
- **Higher Level:** Create a function processNumbers(operation string, numbers ...int) that takes an operation ("sum", "average", "max") and a variadic list of integers. The function should return the calculated result

based on the operation.

Experiment 8: Pointers and Memory Management

- **Low Level:** Declare an integer variable and a pointer that points to it. Modify the variable's value using the pointer and confirm the change in the original variable.
- **Higher Level:** Implement a function that takes a string as input and reverses it **in-place** using pointers, avoiding extra memory allocation where possible. Discuss the implications of Go's garbage collector.

Experiment 9: Structs and Methods

- **Low Level:** Define a Student struct with fields ID, Name, and Grade. Create an instance of Student and write a method DisplayStudentInfo() that prints its details.
- **Higher Level:** Design a Vector struct representing a 2D vector (X, Y coordinates). Implement methods for Add(other Vector), Subtract(other Vector), and Magnitude(). Demonstrate their usage.

Experiment 10: Interfaces

- **Low Level:** Define an interface GeometricShape with methods Area() float64 and Perimeter() float64. Implement this interface for Square and Circle structs.
- **Higher Level:** Create an interface DataProcessor with a method Process(data []float64) (float64, error). Implement DataProcessor for a MeanCalculator and a StandardDeviationCalculator, demonstrating polymorphism.

Experiment 11: Packages and Modules

- **Low Level:** Create a custom package named stringutils with a function Reverse(s string) string. Build and use this package from your main application.
- **Higher Level:** Develop a Go module for simple **matrix operations** (e.g., matrix addition, scalar multiplication). Structure it as a proper Go module, add basic tests, and demonstrate its usage in a separate project.

Experiment 12: Goroutines for Concurrency

- **Low Level:** Write a program that launches three separate goroutines. Each goroutine should print a unique message after a random short delay (e.g., using time.Sleep). Observe the non-deterministic output order.
- **Higher Level:** Implement a concurrent prime number checker. Launch multiple goroutines, each checking a range of numbers for primality. Use sync.WaitGroup to wait for all goroutines to complete.

Experiment 13: Channels for Goroutine Communication

- **Low Level:** Create a program where one goroutine generates a sequence of numbers and sends them through a channel, and another goroutine receives and prints them.
- **Higher Level:** Design a simple **pipeline** using channels. For example, one goroutine reads integers from a source, sends them to a channel. A second goroutine squares these numbers and sends them to another channel. A third goroutine prints the squared numbers.

Experiment 14: Error Handling and defer

- **Low Level:** Write a function readFileContent(filename string) that attempts to read the content of a file. If the file doesn't exist, return an error. Use defer to ensure the file is closed properly.
- **Higher Level:** Implement a program that simulates a **database connection**. Create functions for OpenConnection(), ExecuteQuery(), and CloseConnection(). Use defer to ensure the connection is always closed, even if errors occur during query execution.

Experiment 15: Go Applications - Web & Statistical Computing

	• **Low Level:** (Web) Create a basic HTTP server that listens on a specific port and responds with "Welcome to Go Web Server!" for all incoming requests. • **Low Level:** (Statistical) Write a Go program to calculate the **mode** of a given slice of integers. • **Higher Level:** (Web) Enhance the HTTP server to handle different routes (e.g., /hello, /api/data) and return different responses (e.g., JSON data for /api/data). • **Higher Level:** (Statistical) Implement a program to read numerical data from a CSV file, calculate its **mean and standard deviation**, and then generate a simple **ASCII-based histogram** or frequency distribution plot of the data.

<table>
<tr><td>Course Code:
CSE3424</td><td colspan="3">Course Title:Advanced DBMS
Type of Course: Core
Theory &Integrated Laboratory</td><td>L-T-P-C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">Database Management System (CSE3156)</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">The purpose of this course is to provide students with an in-depth understanding of advanced database concepts, moving beyond traditional Relational Database Management Systems (RDBMS). It begins by revisiting RDBMS transactions, focusing on their properties and control mechanisms. Subsequently, it introduces students to Distributed, Parallel, and NoSQL database concepts, including their main characteristics, advantages, disadvantages, and the critical differences among them. The course emphasizes the evolving need to transition from RDBMS to NoSQL for specific application scenarios. Key features of distributed, parallel, and NoSQL systems are considered and studied in detail. The integrated laboratory component provides valuable hands-on experience, allowing students to apply the theoretical concepts learned during this course, primarily utilizing MongoDB as a practical NoSQL database.</td></tr>
<tr><td>Course Objective</td><td colspan="8">This course is designed to improve the learners' EMPLOYABILITY SKILLS by providing practical experience in working with advanced database systems, particularly focusing on NoSQL databases like MongoDB, and understanding the architectural considerations for distributed and parallel databases.</td></tr>
<tr><td>Course Outcomes</td><td colspan="8">On successful completion of this course, students shall be able to:
CO1: Recall and apply the concepts of transactions in RDBMS, including ACID properties and concurrency control mechanisms. (Knowledge, Application)
CO2: Explain the advanced features and architectural models of Distributed, Parallel, and NoSQL databases. (Comprehension)
CO3: Illustrate and configure features specific to Distributed databases, such as replication and fragmentation. (Application)
CO4: Employ Parallel database concepts and NoSQL techniques in real-life applications and problem-solving scenarios. (Application)</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Transactions in RDBMS</td><td>Quiz</td><td colspan="2">Comprehension based Quizzes and assignments.</td><td colspan="4">06 Classes</td></tr>
<tr><td colspan="9">Topics:
RDBMS Transactions: Transaction control state diagram, properties of transactions (ACID: Atomicity, Consistency, Isolation, Durability).
Schedules in Transactions: Serial schedules, Non-Serial schedules, and Serializable schedules.
Serializability: Conflict Serializability and View Serializability.
Conflict Serializability Check: Using Precedence Graphs.
Concurrency Control Mechanisms:
Lock-Based Protocols: Two-Phase Locking (2PL), Strict 2PL.
Time-Stamp Based Protocols: Basic Time-Stamp Ordering, Thomas' Write Rule.</td></tr>
<tr><td>Module 2</td><td>NoSQL Databases</td><td>Programming and Mini Project</td><td colspan="2">Laboratory experiments and Mini Projects on NoSQL Topics using MongoDB/ Casandra.</td><td colspan="4">06 Classes</td></tr>
<tr><td colspan="9">Topics
NoSQL Introduction: Motivation for NoSQL (Scale Out, Commodity Hardware, limitations of RDBMS), Brief History.
Key Features of NoSQL: Non-Relational nature, Schema-Free/Flexible Schema, Simple APIs, Distributed</td></tr>
</table>

architecture.

- NoSQL Architectures/Data Models:

    - Document Databases: (e.g., MongoDB) - structure, use cases.
    - Columnar Databases: (e.g., Cassandra, HBase) - structure, use cases.
    - Key-Value Databases: (e.g., Redis, DynamoDB) - structure, use cases.
    - Graph Databases: (e.g., Neo4j) - structure, use cases.

- Transactions in NoSQL: BASE (Basically Available, Soft state, Eventually consistent) properties for reliable database transactions, contrast with ACID.
- Horizontal Scalability: Achieving with Database Sharding.
- CAP Theorem: Consistency, Availability, Partition Tolerance - understanding trade-offs.
- Case Studies: Detailed overview of MongoDB / Cassandra / AWS DynamoDB / HBase (focus on one or two prominent examples).

Module 3	**Distributed Databases**	Assignment	Assignment on main topics of Distributed Databases	**06 Classes**

Topics:

- **Characteristics of Distributed Databases (Loosely Coupled Systems):** Data distribution, transparency, autonomy.
- **Views in Distributed Databases:** Local and Global view of applications and data.
- **Distributed Processing:** Query processing, transaction management in a distributed environment.
- **Types of Distributed Databases:** Homogeneous and Heterogeneous distributed database systems.
- **Distributed Data Storage:**
    - **Replication:** Full replication, partial replication, advantages, and disadvantages.
    - **Fragmentation:**
        - **Horizontal Fragmentation:** Primary horizontal, derived horizontal.
        - **Vertical Fragmentation:**
        - Mixed fragmentation.
- **Difference between Centralized and Distributed Databases:** Key distinguishing features and implications.

Module 4	**Parallel Databases**	Assignment	Assignment on main topics of Parallel Databases	**06 Classes**

Topics:

- **Characteristics of Parallel Databases (Tightly Coupled Systems):** Shared-nothing, shared-disk, shared-memory architectures.
- **Features of Parallel Databases:** Enhanced performance through parallelism (I/O parallelism, inter-query parallelism, intra-query parallelism).
- **Parallel Architectures:**
    - **Shared Memory Systems:** Advantages and disadvantages.
    - **Shared Disk Systems:** Advantages and disadvantages.
    - **Shared Nothing Systems:** Advantages and disadvantages.
- **Advantages and Disadvantages of Parallel Databases:** Performance benefits, complexity, cost.
- **Differences between Parallel and Distributed Databases:** Architectural, management, and use-case distinctions.

Targeted Application & Tools that can be used:

- **MongoDB** is to be installed and used for all practical lab sessions.
- **Installation Guide:** [suspicious link removed] (or official MongoDB documentation).
- **Tools:** MongoDB Compass (GUI), Mongo Shell.

Project work/Assignment: Students, in self-selected batches, will undertake a mini-project applying advanced database concepts. The project will involve designing and implementing a database solution for a selected real-world application (e.g., Library Management, Banking System, Online Reservation System, Content Management System, Gaming Backend). The project should specifically demonstrate the application of NoSQL concepts, including:

- **CRUD operations** on collections.
- Supporting **ad-hoc queries** and complex filtering.
- Utilizing **indexing flexibility**.
- Implementing **replication** (e.g., replica sets in MongoDB).
- Demonstrating **sharding** for horizontal scalability (if feasible with local setup or conceptualized).
- Retrieving data from multiple documents (e.g., using aggregation pipelines in MongoDB).

Sample Mini Projects:

1. **Content Management System (CMS):**
    - Design a NoSQL database schema to store content assets (text, HTML, media metadata).
    - Implement features for storing, retrieving, updating, and deleting content.
    - Model user comments on blog posts or articles. Explore how MongoDB's flexible schema benefits content management.

2. **Gaming Project Backend:**
    - Design a NoSQL database to manage player profiles, game states, matchmaking data, leaderboards, and telemetry.
    - Implement high-throughput CRUD operations for frequent updates to player data.
    - Explore how MongoDB can handle rapidly changing data structures common in games (e.g., player inventory, quest progress).

Textbook(s):

- **T1:** Sadalage, P. & Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", 1st Edition, 2019 (Wiley Publications).
- **T2:** Stefano Ceri, Giuseppe Pelagatti, "Distributed Databases: Principles and Systems", 2017 (McGraw Hill Education).

References

- **R1:** Elmasri R and Navathe S B, "Fundamentals of Database Systems", 7th Edition, 2017 (Pearson Publication).
- **R2:** Pivert. "NoSQL Data Models: Trends and Challenges", 1st Edition (Wiley).

Topics relevant to SKILL DEVELOPMENT:

- **FOUNDATION:** In-depth understanding of Transaction management, CRUD Operations, Replication, and Sharding in modern database systems.
- **EMPLOYABILITY:** Hands-on project implementations in software development, fostering practical problem-solving, and improving communication skills through batch-wise presentations.
- **HUMAN VALUES AND PROFESSIONAL ETHICS:** Emphasis on **Team Dynamics** and collaborative problem-solving during mini-project development, promoting responsible and ethical practices in data management.

Experiments

Each experiment is designed with a Level 1 (Fundamental) and Level 2 (Advanced/Application) task to cater to varying levels of understanding and promote deeper exploration. All experiments are to be performed using MongoDB.

Experiment 1: Database and Collection Management with Basic CRUD

In this experiment, you'll practice fundamental database and collection operations in MongoDB, then perform basic data manipulation.

Setup:

- **Choose and create** one of the following databases: Employee, Student, University, Banking, or Online Shopping.
- **Delete** the database you just created.
- **Create a new collection** using the db.createCollection(name, option) command.
- **Delete** the collection you just created.

Level 1: Basic Data Manipulation (CRUD)

- Perform **Create, Read, Update, and Delete (CRUD)** operations on documents within the Student database. This includes inserting new student records, querying existing ones, updating student information, and deleting records.

Level 2: Advanced Text Search

- Conduct a **text search** using MongoDB's features on the Employee database to find relevant information within text fields.

Experiment 2: Leveraging MongoDB Operators

This experiment focuses on using various MongoDB operators to refine your queries and updates.

Level 1: Query and Projection Operators

- Construct complex queries using MongoDB's **Query Operators** (e.g., $gt, $lt, $in, $and, $or) and **Projection Operators** (e.g., to select specific fields) on the Student database.

Level 2: Update Operators

- Apply various MongoDB **Update Operators** (e.g., $set, $inc, $push, $pull, $unset) to modify documents in the Employee database.

Experiment 3: Exploring Query Modifiers

Learn how to use query modifiers to enhance the behavior of your MongoDB queries.

Level 1: Student Database Modifiers

- Apply different **query modifiers** (e.g., .sort(), .limit(), .skip()) to refine your data retrieval from the Student database.

Level 2: Employee Database Modifiers

- Experiment with various **query modifiers** on the Employee database to control result sets and performance.

Experiment 4: Introduction to Aggregation Framework

This experiment dives into MongoDB's powerful Aggregation Framework for data processing and analysis.

Level 1: Student Data Aggregation

- Implement fundamental **aggregation commands** (e.g., $match, $group, $project, $count) to process and analyze data within the Student database.

Level 2: Employee Data Aggregation

- Perform diverse and more complex **aggregation commands** (e.g., $unwind, $lookup, $sort within pipelines) to derive insights from the Employee database.

Experiment 5: Database Authentication

Secure your MongoDB databases by implementing user authentication.

Level 1: Student Database Authentication

- Practice using MongoDB **authentication commands** to create users and assign roles, then connect to the Student database with authenticated credentials.

Level 2: *[No higher-level task specified for this experiment.]*

- NIL

Experiment 6: Implementing Replication

Understand and configure MongoDB replication for high availability and data redundancy.

Level 1: Student Database Replication

- Execute and verify all key **replication commands** on the Student database. This includes initiating a replica set and checking its status.

Level 2: Employee Database Replication

- **Implement and configure replication commands** for the Employee database to set up a robust replica set, ensuring data redundancy and automatic fail over.

Experiment 7: Exploring Sharding for Scalability

This experiment introduces sharding, MongoDB's method for horizontal scaling.

Level 1: Student Database Sharding

- Explore and understand the basic **sharding commands** and concepts on the Student database, such as enabling sharding for a database and collection.

Level 2: Employee Database Sharding

- **Implement and configure sharding commands** on the Employee database, including setting up a sharded cluster and choosing an appropriate shard key, to demonstrate horizontal scalability.

rse Code: **3425**	**rse Title:** Programming in C# and .NET **e of Course:** Theory Integrated Lab		**L-T-P-C**	1	0	4	3
Version No.	**1.0**						
rse Pre-requisites	**CSE 1006 : Problem Solving using Java**						
i-requisites							
rse Description	**.NET** is a powerful software framework developed by Microsoft, enabling the creation of diverse applications including form-based desktop applications, web-based applications, and web services. While supporting a variety of programming languages (e.g., VB.NET, C#), **C#** stands out as one of the most popular and versatile languages within the .NET ecosystem. It is widely used to build robust applications for Windows, mobile platforms, and the web, offering extensive functionalities and adhering to industry standards. This course provides a foundational understanding of the **.NET framework**, exploring its architecture and key components. Furthermore, it delves into the features of the **C# programming language**, empowering students to design and implement console, desktop, and web-based applications.						
Course Objective	The objective of this course is to equip learners with the foundational knowledge and practical skills in C# and the .NET framework, thereby enhancing their employability skills in modern software development through hands-on application development.						
Course Outcomes	On successful completion of this course, students shall be able to: • **CO1:** Understand the fundamental concepts, benefits, and architectural components of the **.NET framework**. [Comprehension] • **CO2:** Illustrate and apply the **Object-Oriented Programming (OOP) paradigm** using the C# Language. [Knowledge, Application] • **CO3:** Develop various types of applications (console, desktop, web) by applying **C# programming concepts** and **database connectivity** using ADO.NET. [Application] • **CO4:** Demonstrate the effective use of **event handling mechanisms** in GUI-based applications and ASP.NET Web Forms. [Application]						
Course Content:							

lule 1	Concepts of .NET Technology and Architecture	z	ıonstration of Assembly, ɔduction to IDE	**+ Pr** **(6)**
Topics: Motivation behind the .NET platform; Common Language Infrastructure (CLI); Role of the Common Type System (CTS), Common Language Specification (CLS), and Common Language Runtime (CLR); Understanding Assemblies, Metadata, Namespaces, and Type distinction; Contrast between single-file and multi-file assemblies; Role of the Common Intermediate Language (CIL); Introduction to .NET Core (overview of its cross-platform nature and advantages).				
lule 2	C# Programming Constructs and OOP Concepts	Based Assignments,	ds on Session for the Concepts, Creating a Console application	**(15)**
Topics: C# program structure; Types and Variables (value types, reference types, boxing/unboxing); Expressions; Statements (selection, iteration, jump statements); Structs; Classes and Objects; Namespaces; Constructors and Destructors; Function Overloading; Inheritance (single, multi-level, hierarchical); Operator Overloading; Arrays (single-dimensional, multi-dimensional, jagged); Interfaces (defining and implementing); Access Modifiers (public, private, protected, internal); Working with Console Input & Output; Properties and Indexers; Enums (enumerations).				
lule 3	Event Handling and Database Connectivity (ADO.NET)	based Assignments	tice Exercises using Visual Studio.NET	**(9)**
Topics: Delegates and Events (event declaration, event handlers, raising events); Exception Handling (try-catch-finally, throwing custom exceptions); ADO.NET (Advantages of ADO.NET, ADO.NET Architecture); .NET Data Providers; Key ADO.NET Objects: Connection, Command, Data Reader, Data Set; Working with Connection-Oriented (DataReader) and Connection-Less (DataSet) approaches; Programming Windows Forms Applications: The notifier-subscriber paradigm for handling events; Utilizing the .NET framework for handling GUI events (mouse clicks, keyboard input, etc.).				

dule 4	Introduction to ASP.NET Web Forms	based Assignments	tice Exercises using Visual Studio.NET	**(6)**
Topics: Introduction to Web Forms architecture; Basic working principles of Web Forms (Page Life Cycle); Standard Web Form Controls; Connectivity with the database in ASP.NET Web Forms; Validation Controls (RequiredFieldValidator, RangeValidator, RegularExpressionValidator, CompareValidator, CustomValidator, ValidationSummary).				

List of Experiments

Experiment 1: [Module 1]

- **Level 1:** Set up your development environment and demonstrate how to write, compile, and run a basic **C# console application** using the **.NET framework** and **Visual Studio IDE**.
- **Level 2:** Use the **ILDasm tool** to inspect the Common Intermediate Language (CIL) generated from a simple C# assembly. Discuss the role of CIL and metadata in the .NET framework.

Experiment 2: [Module 2]

- **Level 1:** Write a C# program to accept an **employee's birth year**. Determine if it's a **leap year** (February 29th birth-date) and, if so, print a "Surprise Gift!" message.
- **Level 2:** Develop a C# program to check if a user's input number belongs to the **Fibonacci series**.

Experiment 3: [Module 2]

- **Level 1:** Create a C# program to manage student marks. Store 10 students' names and marks for 5 subjects. Calculate and display each student's total marks.
- **Level 2:** Write a program to store student names and roll numbers. Implement and use different **sorting techniques** (e.g., Bubble Sort, Selection Sort) to sort students by roll number.

Experiment 4: [Module 2]

- **Level 1:** Design a BankAccount class with data members (depositor name, account number, type, balance) and methods (AssignInitialValues, DepositAmount, WithdrawAmount with balance check, DisplayAccountInfo). Write a C# program to demonstrate its functionality.
- **Level 2:** Define a Person class (with name, age, default/parameterized constructors, input/output methods). Create an **array of 5** Person **objects**, populate their data, and call all class methods.

Experiment 5: [Module 2]

- **Level 1:** Write a C# program to demonstrate **single and multi-level inheritance** using relevant class examples.
- **Level 2:** Extend the Person class to create an Emp class (adding empno, position, constructors, input/output methods). Further extend Emp to Manager (adding bonus, overriding input/output methods). Create Manager objects in main to demonstrate the hierarchy.

Experiment 6: [Module 2]

- **Level 1:** Calculate the **area of different shapes** (e.g., circle, rectangle, triangle) using **method overloading**.
- **Level 2:** Write a C# program to **merge data from two separate groups** (e.g., student lists) into a single, consolidated group.

Experiment 7: [Module 2]

- **Level 1:** A teacher stores student marks in an array. Write a program to find the **highest and lowest marks** in the class, along with the count of students who scored those marks.
- **Level 2:** Develop a **currency converter application** that allows users to convert amounts between different currencies.

Experiment 8: [Module 3]

- **Level 1:** Create a C# application for voter registration. If the user's age is less than 18, the application should **raise a custom exception**.
- **Level 2:** Develop a **desktop-based application** to display employee salary and leave balance.

eriment 9: [Module 3]

- **Level 1:** Design a **login screen** that prompts for username and password. If the credentials are valid (hardcoded check), display a welcome message upon button click using **event handling**.
- **Level 2:** Create a functional **calculator application** using C# for basic arithmetic operations (+, -, *, /).

Experiment 10: [Module 3]

- **Level 1:** Connect to a database (e.g., SQL Server). Implement a C# application that allows an administrator to **insert, update, and modify CSE student data** from the database.
- **Level 2:** Develop a **Windows-based application** for an online quiz for CSE students, managing quiz questions and responses using database connectivity.

Experiment 11: [Module 4]

- **Level 1:** Design an **ASP.NET web page** to collect participant registration details for a cultural festival.
- **Level 2:** Design a **website** to display information about various university departments, using multiple linked web pages.

Experiment 12: [Module 4]

- **Level 1:** Design a **product feedback form** with **validation controls** (e.g., RequiredFieldValidator, RangeValidator) to ensure data integrity.
- **Level 2:** Enhance the feedback form from Level 1. Include **comparison charts** (conceptual or simple display based on aggregated data) to visualize product reviews across different months.

Targeted Application & Tools that can be used:

- Microsoft Visual Studio .NET 2022 (or a more recent version).
- Visual Studio Code (for cross-platform C# development).

Project work/Assignment:

- Problem Solving: Design algorithms and implement programs demonstrating proficiency in C# and .NET concepts.
- Programming: Implement a given scenario or application using C# and the .NET framework, potentially involving a console application, a desktop GUI application, or a simple web application with database connectivity.

tbook(s):

- T1: Herbert Schildt, "C# 4.0 The Complete Reference", Fourth Edition, TMH.
- T2: Matthew Macdonald, "ASP.NET: The Complete Reference", McGraw Hill Education.

erences:

- R1: Joseph Albahari and Ben Albahari, "C# 3.0/4.0 in NUTSHELL", O'REILLY.
- R2: Andrew Troelsen, "C# and the .NET Platform" 1st edition Apress.
- R3: Matthew Macdonald, "Beginning ASP.NET 4.5 in C#", Wiley India.

Online References

- C# Tutorial (W3Schools): https://www.w3schools.com/cs/
- Microsoft C# Tutorials: https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/
- Microsoft ASP.NET Tutorials: https://docs.microsoft.com/en-us/aspnet/tutorials

Topics relevant to SKILL DEVELOPMENT:

- **Employability Skills:** Developed through **Web Application development** and other practical programming assignments, emphasizing **Experiential Learning techniques**.
- This is attained through the assessment components mentioned in the course handout and the detailed lab experiments.

Course Code: CSE3426	Course Title: Front-end Full Stack Development Type of Course: Core Theory &Integrated Laboratory	L- T-P- C	2	0	2	3
Version No.	1.0					
Course Pre-requisites	CSE1504 Web Technologies					
Anti-requisites	**NIL**					
Course Description	This intermediate course focuses on equipping students with the skills necessary for front-end full-stack development, with a strong emphasis on employability. It covers key technologies, architectures, and methodologies that enable students to design, develop, and deploy modern web applications. On successful completion of this course, students will be well-prepared to pursue a career in full-stack development, demonstrating strong problem-solving abilities and practical expertise in front-end technologies.					
Course Objectives	The primary objective is to familiarize learners with the core concepts of **front-end full-stack development** and to enhance their **employability** through **experiential learning techniques**.					
Course Outcomes	On successful completion of this course, students will be able to: • **CO1:** Describe the fundamentals of **DevOps** principles and the overall landscape of **front-end full-stack development**. [Comprehension] • **CO2:** Illustrate and construct a basic web design using **HTML, CSS, and JavaScript**. [Application] • **CO3:** Illustrate the development of a **responsive web design** using modern frameworks and techniques. [Application] • **CO4:** Apply the concepts of **Angular.js** (or a similar framework) to develop a dynamic web front-end. [Application]					
Course Content:						

Module	Title	Type		Programming	Sessions
Module 1	Fundamentals of DevOps	Project		Programming	**04 Sessions**

Topics: Introduction to Agile Methodology; Scrum Fundamentals (Roles, Artifacts, and Rituals); DevOps – Architecture, Lifecycle, Workflow, and Principles; DevOps Tools Overview – Jenkins, Docker, Kubernetes; Review of GIT source control for collaborative development.

Module 2	Web Design & Development	Project		Programming	**03 Sessions**

Topics: HTML5 – Syntax, Attributes, Events, Web Forms 2.0, Web Storage, Canvas, Web Sockets; CSS3 – Colors, Gradients, Text, Transforms, Animations.
Assignment: Develop a static website for managing HR policies of a department, demonstrating HTML5 and CSS3 features.

Module 3	Responsive web design	Project		Programming	**08 Sessions**

Topics: Bootstrap for Responsive Web Design; JavaScript – Core syntax, HTML DOM manipulation, Objects, Classes, Asynchronous programming; Ajax and jQuery Introduction.
Assignment: Design and develop a website that can dynamically track entry-exit information of a housing society, incorporating responsive design principles and JavaScript for interactive elements.

Module 4	Fundamentals of Angular.js	Project		Programming	**15 Sessions**

Topics: Setting up Development & Build Environment: Node.js and NPM; Introduction to TypeScript; Working with OOP concepts with Typescript; Angular Fundamentals (Components, Modules, Data Binding); Angular CLI; Debugging Angular applications; Angular Directives; Using Services & Dependency Injection; Angular Routing; Observable; Handling Forms in Angular Apps; Output transformation using Pipes; Making HTTP Requests; Authentication & Route Protection; Dynamic Components; Angular Modules & Optimizing Angular Apps; Deploying an Angular App; Angular Animations; Adding Offline Capabilities with Service

	Workers; Unit Testing in Angular Apps (Jasmine, Karma); Overview of React.js (as a comparison). **Assignment**: Develop a software tool to do inventory management in a warehouse using Angular.js (or a similar framework).
	Targeted Application & Tools that can be used: • Application Area: Design and develop efficient and user-friendly front-end applications. • Professionally Used Software: GCC compiler (for understanding underlying compilation processes, though not directly used in front-end development).
	Text Book: **T1**. Fender, Young, "Front-end Fundamentals", Leanpub, 2015. **T2**. Northwood, Chris, "The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer", APress, 2018.
	References: **R1.** Flanagan D S, "Javascript : The Definitive Guide" 7th Edition. 7th ed. O'Reilly Media; 2020. **R2**. Alex Libby, Gaurav Gupta, and Asoj Talesra. "Responsive Web Design with HTML5 and CSS3 Essentials", Packt Publishing, 2016. **R3.** Duckett J Ruppert G Moore J. "Javascript & Jquery : Interactive Front-End Web Development."; Wiley; 2014. Web Reference: • https://www.youtube.com/watch?v=JGNTYXkVCVY&list=PLd3UqWTnYXOkTSBCBNyyhxo_jxlY_uTWA&index=2 • https://www.freecodecamp.org/news/frontend-web-developer-bootcamp/ • https://puniversity.informaticsglobal.com:2229/login.aspx?direct=true&db=nlebk&AN=2233842&site=ehost-live (EBSCO database link, access may be restricted) • https://nptel.ac.in/courses/106102064 (NPTEL course on web technologies
	Topics relevant to SKILL DEVELOPMENT: • **DevOps Tools Overview** – Jenkins, Docker, Kubernetes for developing Employability Skills through Experiential Learning techniques. This is attained through the assessment component mentioned in the course handout.
	Experiments ***Experiment 1: Setting up the Development Environment & Git Basics*** • **Level 1:** Install Node.js and NPM. Set up a basic HTML/CSS/JavaScript project. Initialize a Git repository, add files, commit changes. • **Level 2:** Create a GitHub repository. Push your local repository to GitHub. Explore branching and merging workflows with a simple example. ***Experiment 2: HTML5 Fundamentals*** • **Level 1:** Create a webpage using common HTML5 elements (headings, paragraphs, lists, links, images). Use semantic elements like <article>, <nav>, <footer>. • **Level 2:** Develop a more complex HTML5 form using various input types, including Web Forms 2.0 features. Implement basic client-side validation using HTML5 attributes. ***Experiment 3: CSS3 Styling and Layout*** • **Level 1:** Style the webpage from Experiment 2 using CSS3. Experiment with colors, fonts, text formatting, and basic box model properties. • **Level 2:** Implement a two-column or three-column layout using CSS3 Flexbox or Grid. Explore CSS3

transitions and transforms to add visual effects.

Experiment 4: JavaScript Fundamentals

- **Level 1:** Write JavaScript code to manipulate the DOM (change text, hide/show elements, add classes). Use event listeners to respond to user interactions (e.g., button clicks).
- **Level 2:** Implement a simple interactive feature on a webpage (e.g., an image slider, a basic calculator) using JavaScript.

Experiment 5: JavaScript - Objects and Classes

- **Level 1:** Create JavaScript objects with properties and methods. Demonstrate different ways to create objects (literal notation, constructor function).
- **Level 2:** Define JavaScript classes using the class syntax. Implement inheritance and polymorphism with a simple example.

Experiment 6: Asynchronous JavaScript and Fetch API

- **Level 1:** Use setTimeout and setInterval to create asynchronous behavior. Demonstrate the use of callbacks.
- **Level 2:** Make an HTTP request to a public API (e.g., a weather API) using the Fetch API. Parse the JSON response and display the data on a webpage.

Experiment 7: jQuery Fundamentals

- **Level 1:** Use jQuery to select elements, manipulate the DOM, and handle events. Compare and contrast with plain JavaScript.
- **Level 2:** Implement a dynamic feature using jQuery (e.g., a tabbed interface, an accordion).

Experiment 8: Responsive Web Design with Bootstrap

- **Level 1:** Create a webpage layout using Bootstrap's grid system. Use Bootstrap's CSS classes for styling and responsiveness.
- **Level 2:** Implement a responsive navigation bar and a responsive image gallery using Bootstrap components.

Experiment 9: Introduction to Angular - Components and Templates

- **Level 1:** Set up an Angular project using Angular CLI. Create a simple component with a template and display data using interpolation.
- **Level 2:** Create multiple components and demonstrate component communication using input and output properties.

Experiment 10: Angular - Data Binding and Directives

- **Level 1:** Implement one-way and two-way data binding in Angular using different binding syntaxes.
- **Level 2:** Use built-in Angular directives (e.g., *ngIf, *ngFor, *ngClass) to dynamically control the appearance and behavior of a component.

Experiment 11: Angular - Services and Dependency Injection

- **Level 1:** Create a simple service to share data between components. Inject the service into a component using dependency injection.
- **Level 2:** Implement a more complex service that fetches data from an API and makes it available to multiple components.

Experiment 12: Angular - Routing

	Level 1: Set up basic routing in an Angular application with multiple routes and navigation links. **Level 2:** Implement parameterized routes and route guards to control access to different parts of the application. ***Experiment 13: Angular - Forms*** **Level 1:** Create a basic form in Angular using template-driven forms. Implement form validation. **Level 2:** Implement a more complex form using reactive forms. Use custom validators and handle form submission. ***Experiment 14: Angular - HTTP and Observables*** **Level 1:** Use the Angular HttpClient to make HTTP requests to a public API and display the data. **Level 2:** Implement error handling and loading indicators when making HTTP requests. Use Observables to manage asynchronous data streams. ***Experiment 15: Angular - Deployment and Testing*** **Level 1:** Build and deploy an Angular application to a simple hosting environment (e.g., Netlify, GitHub Pages). **Level 2:** Write unit tests for a simple Angular component using Jasmine and Karma.

Course Code: CSE3427	Course Title: Java Full Stack Development Type of Course: Theory Integrated Lab			L- T-P- C	2	0	2	3
Version No.		1.0						
Course Pre-requisites		**CSE1006 Problem Solving Using Java**						
Anti-requisites								
Course Description		This advanced-level course enables students to perform full-stack development using Java, with emphasis on employability skills. The key technologies used for Full Stack development are based on either Java technology or .NET technology. In this course, the focus is on using Java, and the related technologies/tools like Java EE, Java Persistence, Hibernate, Maven, Spring Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.						
Course Objectives		This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.						
Course Outcomes		**On successful completion of the course the students shall be able to:** 1] Practice the use of Java for full stack development [Application] 2] Show web applications using Java EE. [Application] 3] Solve simple applications using Java Persistence and Hibernate [Application] 4] Apply concepts of Spring to develop a Full Stack application. [Application] 5] Employ automation tools like Maven, Selenium for Full Stack development. [Application]						
Course Content:								

Module 1	Introduction	Project		Programming	**03 Sessions**
	Topics: Review of Java; Advanced concepts of Java (e.g., Lambda Expressions, Streams, Optional); Java Generics; Java IO; New Features of Java (relevant to recent LTS versions); Unit Testing tools (JUnit basics).				
Module 2	Java EE Web Applications	Project		Programming	**05 Sessions**
	Topics: Introduction to Eclipse & Tomcat; JSP Fundamentals; Reading HTML Form Data with JSP; State Management with JSP (Session, Cookies, Application Scope); JSP Standard Tag Library (JSTL) - Core & Function Tags; Servlet API Fundamentals; ServletContext, Session, Cookies; Request Redirection Techniques (Forward, Redirect); Building MVC App with Servlets & JSP; Complete App - Integrating JDBC with MVC App. **Assignment:** Develop an application for managing HR policies of a department (e.g., employee details, leave requests).				
Module 3	Java Persistence using JPA and Hibernate	Project		Programming	**06 Sessions**
	Topics: Fundamentals of Java Persistence with Hibernate; JPA for Object/Relational Mapping (ORM); Querying with JPA (JPQL and Criteria API); Caching (First & Second Level Caching), Performance and Concurrency considerations (Batch Fetching, Optimistic Locking & Versioning); Entity Relationships (One-to-One, One-to-Many, Many-to-Many), Inheritance Mapping & Polymorphic Queries. **Assignment:** Design and develop a website that can actively keep track of entry-exit information of a housing society (e.g., resident check-in/out, visitor logs).				
Module 4	Spring Core	Project		Programming	**10 Sessions**
	Topics: Understanding Spring Framework (IoC, DI); Spring Core (Beans, Context); Spring MVC (Dispatcher Servlet, Controllers, Views); Building a Database Web App with Spring and Hibernate (Integration); Spring AOP (Aspect-Oriented Programming) - Introduction and Use Cases; Implementing Spring Security (Authentication, Authorization); Developing Spring REST API; Using Spring Boot for Rapid Development				

	(Auto-configuration, Starter Dependencies, Embedded Servers). **Assignment**: Develop a software tool to do inventory management in a warehouse (e.g., product entry, stock levels, order processing).				
Module 5	Automation tools	Project		Programming	**06 Sessions**
	Topics: Introduction to Automation Tools; Apache Maven: Maven Fundamentals, Software Setup - Command-line and Eclipse Integration, pom.xml and Directory Structure, Multi-Module Project Creation, Scopes (Compile, Test, Provided, Runtime, System), Dependency Management, Profiles; Functional/BDD Testing using Selenium: Selenium Fundamentals and IDE, Selenium WebDriver, Installation and Configuration, Locating WebElements, Driver Commands, WebElement Commands, TestNG/JUnit integration for Selenium tests. **Assignment**: Illustrate the use of automation tools in the development of a small software project (e.g., automate build process, run basic UI tests).				
	Targeted Application & Tools that can be used: **Targeted Application Area:** Design and Analyzing the efficiency of Algorithms (While the course is practical, the underlying problem-solving skills align with algorithmic thinking). This fundamental course is used by all application developers. **Tools**: Google Colab (for general programming practice/examples, though not ideal for full Java EE/Spring setup), Eclipse, NetBeans, Hibernate, Selenium, Maven, GIT.				
	Project work/Assignment: Problem Solving: Design of Algorithms and implementation of programs. Programming: Implementation of given scenarios using Java.				
	Text Book: **T1**. Fender, Young, "Front-end Fundamentals", Leanpub, 2015				
	References **R1**. Soni, Ravi Kant. "Full Stack AngularJS for Java Developers: Build a Full-Featured Web Application from Scratch Using AngularJS with Spring RESTful.", Apress, 2017. **R2**. Mardan, Azat. "Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB.", Apress, 2015 Web References • Oracle Java Documentation: https://docs.oracle.com/en/java/ • Spring Framework Documentation: https://docs.spring.io/spring-framework/docs/current/reference/html/ • Hibernate Documentation: https://docs.jboss.org/hibernate/orm/5.6/userguide/html_single/Hibernate_User_Guide.html • Maven Documentation: https://maven.apache.org/guides/ • Selenium Documentation: https://www.selenium.dev/documentation/ • TutorialsPoint (Java, JSP, Servlets, Spring, Hibernate): https://www.tutorialspoint.com/ • Baeldung (Spring, Java EE, Hibernate tutorials): https://www.baeldung.com/				
	List of experiments Each experiment will have a Low-Level (basic understanding and implementation) and a High-Level (complex scenario, additional features, or optimizations) component. **Module 1: Introduction & Advanced Java**				

Experiment 1: Java Review & Generics

1. **Low Level:** Implement a simple Java program to demonstrate ArrayList usage with and without generics. Observe and explain type-safety benefits.
2. **High Level:** Create a generic utility class (e.g., a Pair class or a custom LinkedList) that can handle different data types and demonstrate its usage with various types. Implement a basic unit test for this utility using JUnit.

Experiment 2: Java I/O & New Features

1. **Low Level:** Write a Java program to read data from a text file and write it to another file. Demonstrate the use of try-with-resources.
2. **High Level:** Implement a program to process a CSV file containing student records. Use Java Streams to filter, sort, and collect specific data. Write the processed data to a new CSV file or display it on the console.

Module 2: Java EE Web Applications

Experiment 3: JSP Fundamentals

1. **Low Level:** Create a simple JSP page that displays "Hello World" and the current date/time.
2. **High Level:** Design a JSP page that takes user input (e.g., name, age) via an HTML form and displays it back to the user after submission.

Experiment 4: State Management with JSP & Servlets

1. **Low Level:** Develop a simple counter application using JSP and Servlet that increments a count and stores it in the session. Display the count on the JSP page.
2. **High Level:** Create a simple e-commerce like application where users can add items to a cart. Use session management to store cart items. Display the cart contents and total price. Implement login/logout functionality using sessions and cookies.

Experiment 5: MVC with Servlets & JSP

1. **Low Level:** Implement a basic MVC application for user registration where a Servlet acts as the controller, a JSP as the view for the form, and another JSP for displaying success/failure.
2. **High Level:** Extend the MVC application to perform CRUD operations (Create, Read, Update, Delete) on a simple entity (e.g., "Product" or "Task"). Integrate a simple JDBC connection to a database (e.g., H2 or MySQL) to persist the data.

Module 3: Java Persistence using JPA and Hibernate

Experiment 6: Basic JPA & Hibernate Setup

1. **Low Level:** Configure Hibernate and JPA to map a simple Java POJO (e.g., Student with ID and Name) to a database table. Perform a basic save operation.
2. **High Level:** Extend the previous experiment to include basic CRUD operations (create, read, update, delete) for the Student entity using EntityManager. Demonstrate find and merge methods.

Experiment 7: Entity Relationships

1. **Low Level:** Model a One-to-Many relationship (e.g., Department has many Employees) using JPA annotations. Perform basic operations to save and retrieve related entities.
2. **High Level:** Implement a Many-to-Many relationship (e.g., Student can enroll in many Courses, and Course has many Students). Demonstrate saving and retrieving entities from

both sides of the relationship.

Experiment 8: JPQL and Criteria API

1. **Low Level:** Write a few basic JPQL queries to retrieve data from your entities (e.g., select all students, select students by name).
2. **High Level:** Implement more complex queries using JPQL (e.g., joins, aggregation functions). Use the Criteria API to construct a dynamic query with multiple search criteria.

Module 4: Spring Core & Spring Boot

Experiment 9: Spring Core - IoC and DI

1. **Low Level:** Create a simple Spring application demonstrating Inversion of Control (IoC) and Dependency Injection (DI) using XML-based configuration (e.g., inject a MessageService into a Client class).
2. **High Level:** Refactor the previous experiment to use annotation-based configuration (@Component, @Autowired). Demonstrate constructor injection and setter injection.

Experiment 10: Spring MVC Basic Application

1. **Low Level:** Develop a simple Spring MVC application that displays a static message on a JSP page using a Controller and ViewResolver.
2. **High Level:** Create a Spring MVC application with a form that submits data to a controller. Validate the form data and display appropriate messages on the view.

Experiment 11: Spring Boot REST API

1. **Low Level:** Develop a simple Spring Boot application that exposes a "Hello World" REST endpoint.
2. **High Level:** Create a RESTful API for a simple resource (e.g., Book with ID, title, author). Implement GET (all, by ID), POST, PUT, and DELETE operations. Use an in-memory database like H2 for persistence.

Experiment 12: Spring Data JPA Integration with Spring Boot

1. **Low Level:** Integrate Spring Data JPA into a Spring Boot application. Create a simple UserRepository interface and use its default methods to perform CRUD operations on a User entity.
2. **High Level:** Extend the UserRepository with custom queries using @Query annotation. Implement pagination and sorting for retrieving data.

Module 5: Automation Tools

Experiment 13: Maven Build Automation

1. **Low Level:** Create a simple Maven project for a Java application. Compile, package, and run the application using Maven commands.
2. **High Level:** Convert an existing simple Java EE project (from Module 2) into a Maven project. Manage dependencies (e.g., Servlet API, JSP API, JDBC driver) using pom.xml. Demonstrate building a WAR file.

Experiment 14: Selenium WebDriver - Basic Interaction

1. **Low Level:** Write a Selenium WebDriver script to open a web browser, navigate to a specific URL (e.g., Google), find a search box, enter text, and click the search button.
2. **High Level:** Develop a Selenium script to automate a login process on a sample website

(e.g., a test login page). Capture screenshots on success/failure and verify the title of the page after successful login.

Experiment 15: Selenium with TestNG/JUnit for Functional Testing

1. **Low Level:** Integrate Selenium scripts with JUnit (or TestNG). Write a basic test case that opens a browser, navigates to a URL, and asserts an element's presence.
2. **High Level:** Design a suite of functional tests for a simple web application (e.g., the HR policy application or housing society application developed in earlier modules). Create test methods for different scenarios (e.g., adding a new record, editing, deleting, search). Generate a test report.

Course Code: CSE3428	Course Title: .NET Full Stack Development Type of Course: Theory Integrated Lab	L- T-P- C	2	0	2	3

Version No.		1.0
Course Pre-requisites		CSE1504 Web Technologies
Anti-requisites		**CSE3427 Java Full Stack Development**
Course Description		This advanced level course enables students to perform full-stack development using .NET, with emphasis on employability skills. The key technologies used for Full Stack development is based on either Java technology or .NET technology. In this course, the focus is on using .NET and the related technologies/tools like C#, ASP.NET, Entity Framework Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.
Course Objectives		This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies
Course Outcomes		**On successful completion of the course the students shall be able to:** 1] Practice the use of C# for developing a small application [Application] 2] Show web applications using Entity Framework. [Application] 3]Solve simple web applications that use SQL and ASP.NET [Application] 4] Apply concepts of ASP.NET to develop a Full Stack application. [Application]
Course Content:		

Module 1	C# Programming for Full Stack Development	Project		Programming	**10 Sessions**
	Topics: .NET Framework/Core Fundamentals, Visual Studio IDE Fundamentals, C# Language Features (variables, operators, expressions, decision and iteration statements), Working with Arrays and Collections (List, Dictionary), Managing Program Flow and Events, Object-Oriented Programming (OOP) concepts (Classes, Methods, Inheritance, Polymorphism, Encapsulation, Abstraction), Properties (including Auto-Implemented), Delegates, Anonymous Methods and Anonymous Types, Extension Methods, Sealed Classes/Methods, Partial Classes/Methods, Asynchronous programming and Threading (async/await), Data validation and working with data collections including LINQ (Language Integrated Query), Handling Errors and Exceptions (try-catch-finally), Working with Files and Streams, Unit Testing – NUnit framework. **Assignment:** Develop a small application for managing a library using C# (e.g., adding books, searching, borrowing/returning).				
Module 2	Entity Framework Core 2.0	Project		Programming	**06 Sessions**
	Topics: Introduction to Entity Framework Core (EF Core) and Object-Relational Mapping (ORM); EF Core Code First Approach (Model-First and Database-First overview); Introduction to EDM (Conceptual, Logical, Physical Model); Querying the EDM (LINQ to Entities, DbSet operations); Working With Stored Procedures (Calling and Mapping); Advanced Entity Framework - DbContext (Lifecycle, Change Tracking); Advanced Operations (Transactions, Concurrency, Lazy/Eager Loading); Performance Optimization techniques in EF Core; Data Access with ADO.NET (brief review for understanding underlying mechanisms). **Assignment:** Develop an application for managing HR policies of a department using EF Core (e.g., employee details, department assignments, leave requests).				
Module 3	ASP.NET	Project		Programming	**06 Sessions**
	Topics: Introduction to ASP.NET Core; ASP.NET Core 3.1 MVC (or latest stable version) Architecture;				

	ASP.NET Core Middleware and Request Pipeline; Review of SQL using MS SQL (basic DDL and DML operations); Working With Data In ASP.NET Core (Model Binding); Razor View Engine (Syntax, Layouts, Partials); State Management In ASP.NET Core MVC (Session, TempData, ViewData, ViewBag, Hidden Fields, Query Strings, Cookies); Layouts and Partial Views. **Assignment**: Develop a web application to mark entry/exit of guests in a building (e.g., guest registration, timestamping entry/exit).				
Module 4	ASP.NET	Project		Programming	**08 Sessions**
	Topics: Introduction To Models (Data Transfer Objects, View Models); Validations In ASP.NET Core MVC (Data Annotations, Custom Validations); Authentication and Authorization In ASP.NET Core MVC (Identity Framework basics, Role-based authorization); Advanced ASP.NET Core MVC - AJAX Action Link and AJAX Forms in MVC (Unobtrusive Ajax); Microsoft Testing Framework (MSTest) / NUnit for Unit Testing the .NET Application (focus on Controller and Service Layer testing). **Assignment**: Develop a software tool to do inventory management in a warehouse using ASP.NET Core MVC (e.g., product entry, stock levels, order processing, with user authentication).				
	Targeted Application & Tools that can be used: Targeted Application Area: Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers. Tools: Visual Studio IDE (Community/Professional), SQL Server Management Studio (SSMS) or Azure Data Studio, Git. While Google Colab is mentioned, it's not directly applicable for .NET development.				
	Project work/Assignment: Problem Solving: Design of Algorithms and implementation of programs. Programming: Implementation of given scenarios using .NET.				
	Text Book: **T1**. Fender, Young, "Front-end Fundamentals", Leanpub, 2015 **T2**. Valerio De Sanctis, "ASP.NET Core 5 and Angular: Full-stack web development with .NET 5 and Angular 11", 4th Edition, Packt, 2021.				
	References **R1.** Benjamin Perkins, Jon D. Reid, "Beginning C# and .NET", Wiley, 2021. **R2.** Piotr Gankiewicz, "Full Stack .NET Web Development", Packt Publishing, 2017. **R3.** Tamir Dresher, Amir Zuker, Shay Friedman, "Hands-On Full-Stack Web Development with ASP.NET Core", Packt Publishing, 2018. **R4.** Dustin Metzgar, "Exploring .NET core with microservices, ASP.NET core, and Entity Framework Core", Manning, 2017. Web References • Microsoft .NET Documentation: https://docs.microsoft.com/en-us/dotnet/ • ASP.NET Core Documentation: https://docs.microsoft.com/en-us/aspnet/core/ • Entity Framework Core Documentation: https://docs.microsoft.com/en-us/ef/core/ • C# Documentation: https://docs.microsoft.com/en-us/dotnet/csharp/ • Tutorials for C#, ASP.NET Core, EF Core: https://www.tutorialspoint.com/ • Microsoft Learn (Interactive learning paths): https://learn.microsoft.com/en-us/				

List of Experiments

Each experiment will have a Low-Level (basic understanding and implementation) and a High-Level (complex scenario, additional features, or optimizations) component.

Module 1: C# Programming for Full Stack Development

Experiment 1: C# Fundamentals & Console Application

1. **Low Level:** Write a C# console application to calculate the area and perimeter of a rectangle. Use basic variables, input/output, and arithmetic operators.
2. **High Level:** Develop a C# console application that simulates a simple ATM. Implement features like deposit, withdrawal, and balance inquiry. Use if-else or switch statements, loops, and basic error handling for invalid input.

Experiment 2: OOP Concepts in C#

1. **Low Level:** Create a Book class with properties (Title, Author, ISBN) and a method to display book details. Create objects and demonstrate property access.
2. **High Level:** Design a class hierarchy for Shape (abstract class) with derived classes like Circle and Rectangle. Implement methods to calculate area and perimeter for each shape. Demonstrate polymorphism by creating a list of Shape objects and iterating through them to calculate properties.

Experiment 3: Collections and LINQ

1. **Low Level:** Use List<T> to store a collection of strings and perform basic operations like adding, removing, and iterating.
2. **High Level:** Create a List<Student> objects (with properties like Name, Age, Grade). Use LINQ queries to:
    1. Filter students based on grade.
    2. Order students by name.
    3. Find the average age of students.
    4. Group students by grade.

Experiment 4: Asynchronous Programming & Error Handling

1. **Low Level:** Implement a simple try-catch-finally block to handle DivideByZeroException.
2. **High Level:** Create an asynchronous method using async/await that simulates a long-running operation (e.g., fetching data from a remote server with a delay). Demonstrate how to call this method without blocking the UI/console. Implement custom exception handling for specific scenarios.

Module 2: Entity Framework Core

Experiment 5: EF Core Code First - Basic CRUD

1. **Low Level:** Create a simple .NET Core Console Application. Define a Product entity (Id, Name, Price) and configure EF Core using Code First. Perform Add, SaveChanges, and Read (all products) operations.
2. **High Level:** Extend the previous experiment to include Update and Delete operations for the Product entity. Implement a simple menu-driven console application to interact with these CRUD operations.

Experiment 6: EF Core Relationships (One-to-Many)

1. **Low Level:** Define two entities, Department and Employee, with a one-to-many relationship. Add employees to a department and save them to the database. Retrieve departments and their associated employees.
2. **High Level:** Implement Lazy and Eager loading for the Department-Employee relationship. Compare the SQL queries generated by EF Core for both loading strategies using a debugger or logging.

Experiment 7: EF Core Querying & Advanced Features

1. **Low Level:** Write LINQ to Entities queries to filter products by price range and order them by name.
2. **High Level:** Implement a complex LINQ query involving multiple tables (e.g., Order, OrderItem, Product) to retrieve specific sales information (e.g., total sales for a product, orders placed by a specific customer). Explore transactions in EF Core for atomic operations.

Module 3: ASP.NET Core MVC Fundamentals

Experiment 8: Basic ASP.NET Core MVC Application

1. **Low Level:** Create a new ASP.NET Core MVC project. Develop a simple controller that returns a static "Hello from MVC" message to a Razor View.
2. **High Level:** Create a form on a Razor View for user input (e.g., Name, Email). Implement a controller action to receive and display the submitted data on another view

Experiment 9: State Management in ASP.NET Core MVC

1. **Low Level:** Implement a simple page counter using Session state. Display the count on the view.
2. **High Level:** Develop a multi-step form (e.g., a simple registration wizard). Use TempData to pass data between steps and Session to store accumulated data across the wizard.

Experiment 10: Integrating EF Core with ASP.NET Core MVC

1. **Low Level:** Build a simple ASP.NET Core MVC application to display a list of products from the database using EF Core.
2. **High Level:** Extend the application to implement full CRUD operations (Create, Read, Update, Delete) for the Product entity. Use scaffolded views as a starting point and customize them.

Module 4: Advanced ASP.NET Core MVC & Testing

Experiment 11: Data Validation & View Models

1. **Low Level:** Apply Data Annotations (e.g., [Required], [StringLength], [Range]) to a simple model (e.g., Customer with Name, Email, Age) and demonstrate client-side and server-side validation in an MVC form.
2. **High Level:** Create a dedicated View Model for a complex form (e.g., OrderViewModel for an order form that includes customer details and order items). Implement custom validation logic for specific business rules.

Experiment 12: Authentication & Authorization

1. **Low Level:** Implement basic user login/logout functionality in an ASP.NET Core MVC application using ASP.NET Core Identity. Configure a simple role (e.g., "Admin").
2. **High Level:** Secure specific actions or controllers based on roles (e.g., only "Admin" can

access "Create Product" page). Implement a custom authorization policy.

Experiment 13: AJAX Integration in MVC

1. **Low Level:** Use @Ajax.ActionLink or basic JavaScript to make an AJAX call to a controller action that returns partial view data, updating a specific div on the page without a full page reload.
2. **High Level:** Implement an AJAX-powered search feature for products. As the user types in a search box, dynamically update a list of matching products on the page using AJAX and partial views.

Experiment 14: Unit Testing Controllers and Services

1. **Low Level:** Write a unit test for a simple controller action using MSTest (or NUnit). Use mocking frameworks (e.g., Moq) to mock dependencies like DbContext.
2. **High Level:** Write unit tests for a service layer class that performs business logic (e.g., ProductService that validates product data before saving). Ensure coverage for success and failure scenarios, including exception handling.

Experiment 15: Full Stack Project Integration

1. **Low Level:** Integrate an existing ASP.NET Core MVC application (from previous modules) with EF Core for data persistence. Ensure basic CRUD operations are functional.
2. **High Level:** Take one of the assignment applications (e.g., Inventory Management, HR Policy) and develop it as a complete full-stack solution. This should include:
    1. A well-structured ASP.NET Core MVC project.
    2. Robust EF Core integration for all data operations.
    3. User authentication and authorization.
    4. Client-side validation and potentially some AJAX features.
    5. A set of unit tests for key components (controllers, services).
    6. Deployment to a local IIS Express or simple publish to folder (optional: Azure App Service).

Course Code: CSE3429	**Course Rust Programming** **Type of Course: Core** **Theory &Integrated Laboratory**	**L-T-P-C**	2	0	2	3

Version No.	**1.0**
Course Pre-requisites	**CSE1004 Problem Solving using C**
Anti-requisites	**NIL**
Course Description	Rust is an excellent language for students and anyone curious about systems programming. It's helped many learn about complex topics like operating system development. Rust empowers learners to write faster, more reliable software.Traditionally, programming languages force a trade-off between high-level ease of use and low-level control. Rust defies this by offering both. It strikes a balance between powerful technical capabilities and a great developer experience, letting you manage low-level details like memory usage without the usual headaches. The accompanying lab sessions offer practical, hands-on experience with the concepts you'll learn in the course.
Course Objective	This course is designed to significantly improve the learners' EMPLOYABILITY SKILLS by mastering the Rust Programming language.
Course Outcomes	Upon the successful completion of this course students will be able to: • Create a Rust project, including proper Cargo configuration.[Application] • Translate a design into a working Rust program.[Application] • Explain the causes of, and effectively remedy, type and lifetime errors encountered during Rust programming.[Analysis] • Strategically use structs, enums, and traits in the construction of Rust programs.[Application] • Apply advanced memory management concepts, including references, boxes, cells, and reference counting, in Rust programming..[Application] • Organize a Rust crate into multiple source files using the module system..[Application]
Course Content:	

Module 1	Introduction to Rust Programming	Quiz		**10 sessions**

Topics: Reasons to Adopt Rust (Performance, Memory Safety, Concurrency, Developer Productivity, Community, Use Cases: OS, WebAssembly, CLI tools, Networking, Games, Blockchain, etc.), Career Opportunities, Key Language Features Overview, Advantages of Rust; Installation of Rust Toolchain (rustup), Using Cargo (build system, package manager), Your First Rust Program ("Hello World"); Rust Data Types (Integers, Floats, Booleans, Characters, Tuples, Arrays), Variables and Mutability (let, mut, const, Shadowing); Strings (&str, String, common operations); Operators (Arithmetic, Comparison, Logical, Bitwise); Control Flow (Conditional: if, else if, else; Looping: loop, while, for, break, continue, Loop Labels).

Module 2	Features of Rust	Programming and Mini Project		**12 sessions**

Topics: Unique Features of Rust revisited; Common Types (Option<T>, Result<T, E>) for handling presence/absence and errors; Arrays and Slices; Ownership, Borrowing, and the Stack vs. Heap distinction; Structures (struct fundamentals); Enums (enum for custom types, pattern matching with match); Collections (Vec<T>, String, HashMap<K, V>); Modules (basics of mod, pub, use for organizing code); Error Handling (panic vs. Result, error propagation); Basic Input/Output (reading from stdin, writing to stdout/stderr).

Module 3	**Ownership**	Assignment		**11 sessions**

Topics: Deep dive into Ownership rules and move semantics; Detailed understanding of Borrowing (References: immutable & and mutable &mut); Lifetimes (Lifetime annotations, generic lifetimes, 'static lifetime); Smart Pointers (Box<T>, Rc<T> for multiple ownership, Arc<T> for thread-safe multiple ownership, Cell<T> and RefCell<T> for interior mutability); Modularizing Projects (complex module structures, super, self, crate, external crates); Documenting Code (/// documentation comments, cargo doc); Unit Testing (#[test] attribute, assert!, assert_eq!, assert_ne!, #[should_panic]); Documentation Testing (running code examples in docs).

Module 4	Generics and Concurrency	Assignment		**12 sessions**

Topics: Generics (generic types, functions, structs, enums; Monomorphization); Traits (defining and implementing traits, trait bounds, trait objects for dynamic dispatch, blanket implementations); Static vs. Dynamic Dispatch (performance implications); Closures (defining and using closures, capturing environment); Function Types (fn, FnOnce, FnMut, Fn traits); Macros (declarative macros macro_rules!, procedural macros - brief introduction); Fearless Concurrency (Rust's approach to concurrency safety); Threads (creating and joining threads); Synchronization Primitives (Mutex<T>, RwLock<T>, Message Passing with mpsc channels); Current state and future trends of Rust development.

List of Experiments

Experiment No. 1: Getting Started with Cargo and Basic Conversion

This experiment introduces you to **Cargo**, Rust's build system and package manager, and helps you write a simple conversion program.

- **Level 1: Cargo Project Setup & "Hello World"**
  - **Objective:** Learn to initialize, build, and run a new Rust project using Cargo conventions.
  - **Assignment:** Install Rust and Cargo. Create a new Cargo project. Write a "Hello, <Your Name>!" program that takes your name as input from the command line (if possible, otherwise hardcode for simplicity).
- **Level 2: Fahrenheit to Celsius Converter**
  - **Objective:** Apply basic input/output and arithmetic operations within a Cargo project.
  - **Assignment:** Enhance the project to convert a temperature from Fahrenheit to Celsius. The program should take the Fahrenheit temperature as input and print the Celsius equivalent. Handle numeric input.

Experiment No. 2: Bitwise Operations, Patterns, and Input Handling

This experiment delves into bitwise operations, character patterns, and robust user input handling.

- **Level 1: Binary Equivalence & Basic Bitwise Operation**
  - **Objective:** Understand how to display the binary representation of an integer and perform a simple bitwise operation.
  - **Assignment:** Create a project that takes an integer as input. Display its binary equivalent. Then, perform and display the result of a bitwise AND operation with a fixed number (e.g., 5).
- **Level 2: Comprehensive Bitwise Operations, Dollar Pattern & Guessing Game with Validation**
  - **Objective:** Explore more bitwise operations, practice pattern generation, and implement robust user input validation.
  - **Assignment (Part A):** Extend the previous task to perform and display the results of bitwise OR, XOR, Left Shift, and Right Shift operations on the input integer.
  - **Assignment (Part B):** Create a program that prints a dollar sign ($) pattern (e.g., a square or triangle) to the console using nested loops.
  - **Assignment (Part C):** Build a guessing game where the program generates a random number. The user tries to guess the number. The program must prompt for user input, gracefully handle non-numeric or out-of-range input using Result and match (or expect/unwrap for simpler cases), and provide hints (e.g., "Too high!", "Too low!").

Experiment No. 3: Complex Numbers and Prime Number Operations

This experiment focuses on mathematical operations, including complex numbers and prime number calculations.

- **Level 1: Complex Number Struct & Magnitude**
    - **Objective:** Define a struct for complex numbers and calculate their magnitude.
    - **Assignment:** Create a project that defines a ComplexNumber struct (with real and imaginary parts as floating-point numbers). Implement an associated function to calculate and display the magnitude (absolute value) of a given complex number.
- **Level 2: Complex Number Angle & Comprehensive Prime Number Utilities**
    - **Objective:** Extend complex number operations to include angle calculation and develop a suite of prime number utility functions.
    - **Assignment (Part A):** Add another associated function to your ComplexNumber struct to calculate and display its angle (argument) in radians or degrees.
    - **Assignment (Part B):** Create a project with the following functions related to prime numbers:
        - A function is_prime(n: u64) to check if a given number n is prime.
        - A function count_primes(start: u64, end: u64) to count all prime numbers within a specified range.
        - A function sum_primes(start: u64, end: u64) to calculate the sum of all prime numbers within a specified range.
        - A function display_primes(start: u64, end: u64) to print all prime numbers within a specified range.

Experiment No. 4: Structs, Enums, and Associated Functions

This experiment explores the fundamental concepts of structs, enums, and associated functions for object-like behavior in Rust.

- **Level 1: Simple Structs, Enums, and Associated Functions**
    - **Objective:** Practice defining and using basic structs, enums, and implementing associated functions.
    - **Assignment:** Create a simple program demonstrating the use of a struct (e.g., Car with make, model, year), an enum (e.g., FuelType with Petrol, Diesel, Electric), and an associated function for the struct (e.g., Car::display_details()).
- **Level 2: Turtle Graphics Builder with State Management**
    - **Objective:** Implement a builder pattern using a struct to construct an object, demonstrating methods for state modification and basic geometric transformations.
    - **Assignment:** Implement a Turtle struct with fields for its current x, y coordinates, and angle. Create associated functions Turtle::new() (constructor), Turtle::rotate(angle_degrees: f64), Turtle::move_forward(distance: f64), and Turtle::move_backward(distance: f64). Each movement/rotation should update the turtle's internal state. Print the turtle's position after each operation.

Experiment No. 5: Command-Line Arguments and Data Processing

This experiment focuses on accepting command-line arguments and processing file data using collections.

- **Level 1: Quadratic Equation Solver via Command Line**
    - **Objective:** Learn to parse and use basic numeric command-line arguments in a Rust program.
    - **Assignment:** Write a program to solve a quadratic equation (ax2+bx+c=0). The coefficients a, b,

and c should be passed as command-line arguments. For simplicity, assume valid numeric input for now.

- **Level 2: Robust Quadratic Solver & Word Histogram from File**
    - **Objective:** Enhance command-line argument handling with error management and process text data from files using HashMap.
    - **Assignment (Part A):** Refactor the quadratic equation solver to gracefully handle incorrect numbers of arguments (e.g., not providing 3 coefficients) and non-numeric input for the coefficients, providing informative error messages.
    - **Assignment (Part B):** Write a program that accepts a file path as a command-line argument. Read the content of the file, count the occurrences of each word using a HashMap<String, u32>, and then display the word counts sorted alphabetically by word.

Experiment No. 6: Cellular Automata and Modular Design

This experiment introduces simulating complex systems and organizing your code into reusable modules.

- **Level 1: Basic Conway's Game of Life (Single Generation)**
    - **Objective:** Implement the core logic of Conway's Game of Life for a single generation.
    - **Assignment:** Create a program that initializes a small 2D grid representing a Game of Life board (e.g., 10x10). Define an initial pattern of live and dead cells. Implement the logic to calculate the state of the next generation based on the rules. Print both the initial and next generation grids. (No thread::sleep needed for this level).
- **Level 2: Full Game of Life Simulation & Modular Turtle Library**
    - **Objective:** Implement the full Game of Life simulation over multiple generations and practice organizing code into reusable library modules.
    - **Assignment (Part A):** Extend the Game of Life program to simulate multiple generations. Use std::thread::sleep to introduce a delay between generations for visualization. Allow the user to specify the number of generations or run indefinitely until a certain condition is met.
    - **Assignment (Part B):** Take your Turtle type and its associated functions from Experiment 4 (Level 2) and place them into a separate library module (e.g., turtle_lib) within your Cargo project. Modify your main binary to depend on and use this newly created turtle_lib module.

Experiment No. 7: Robust Error Handling and Generic Data Structures

This experiment focuses on making your programs more robust by implementing proper error handling and building generic data structures.

- **Level 1: Function Returning** Result
    - **Objective:** Understand how to define and return Result types from functions to indicate success or failure.
    - **Assignment:** Create a simple function that attempts to parse a string into an integer. If the parsing is successful, return Ok(integer_value); otherwise, return Err(error_description). Demonstrate calling this function and handling both Ok and Err variants using a match statement.
- **Level 2: Main Function Returning** Result **& Generic Stack/Queue Library**
    - **Objective:** Learn to make the main function return a Result type for consistent error propagation, and design/implement reusable generic data structures.
    - **Assignment (Part A):** Refactor one of your previous projects (e.g., the quadratic equation solver or word histogram) so that the main function returns a Result<(), Box<dyn std::error::Error>>. Ensure all potential errors throughout the program are propagated back to main using the ? operator or explicit error handling.
    - **Assignment (Part B):** Create a new library project named generics_ds. Inside this library, implement generic Stack<T> and Queue<T> data structures. Ensure they support common

operations like push, pop, peek, and is_empty. Create a separate binary project that depends on your generics_ds library and demonstrates the usage of both the Stack and Queue with at least two different data types (e.g., integers and strings).

Targeted Application & Tools that can be used:

Targeted Application Area: Designing and analyzing the efficiency of algorithms. This fundamental course cultivates skills applicable to all application developers, especially in systems programming, high-performance computing, and reliable software development.

Tools:

Rust Toolchain (installed locally: rustup, cargo)
Visual Studio Code (recommended IDE with Rust-analyzer extension)
Command-line interface (Terminal/Command Prompt)
Google Colab: Limited applicability for full Rust development environment. Primarily for conceptual exploration or showcasing small, self-contained Rust snippets if integration is available. The primary environment will be local Rust installation.

Official Rust Website: https://www.rust-lang.org

Project work/Assignment:

Problem Solving: Design of Algorithms and implementation of programs addressing specific computational problems using Rust's capabilities.
Programming: Implementation of given scenarios using Rust, ranging from command-line utilities to small systems-level components.
Batch-wise Presentations: Students will present their project implementations, fostering communication and presentation skills.

Textbook(s):

T1. Klabnik, Steve, and Carol Nichols. The Rust Programming Language. No Starch Press, 2023. Publisher: William Pollock. (Often referred to as "The Book," available online for free).

References

R1. Jim Blandey, Jason Orendorff and Leonora F.S. Tindall, *Programming Rust – Fast, Safe System Development*. 2nd Edition, O'Reilly Publication, 2021.
R2. *Rust Book (Online Version)*: https://rustbook.cs.brown.edu (and also https://doc.rust-lang.org/book/)

Web References

- The Rust Programming Language Official Website: https://www.rust-lang.org
- Rust Documentation: https://doc.rust-lang.org/
- Rust by Example: https://doc.rust-lang.org/rust-by-example/
- Rust Playground (online Rust compiler): https://play.rust-lang.org/
- Crater (Rust package registry): https://crates.io/

<table>
<tr><td>Course Code: CSE3426</td><td colspan="3">Course Title: Front-end Full Stack Development</td><td>L- T-P- C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">CSE2260</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This intermediate course enables students to perform front-end full stack development, with emphasis on employability skills. The course covers key technologies and architectures that enables the student to design and implement front-end. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.</td></tr>
<tr><td>Course Objectives</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Front-end Full Stack Development and attain Employability Skills through Experiential Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="8">On successful completion of the course the students shall be able to:
CO1. Design and develop static web pages using HTML5 elements and CSS3 [Apply]
CO2.Develop responsive web pages using CSS, JavaScript and bootstrap. [Apply]
CO3.Demonstrate the concepts of Angular.js to develop a web front-end. [Apply]
CO4.Illustrate the concepts of React.js to develop a web front-end. [Apply]</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td>Introduction to web technology</td><td>Project</td><td colspan="5">Programming</td><td>15Sessions [7L +8P]</td></tr>
<tr><td colspan="9">Topics:
HTML5 – Syntax,Attributes, Events, Web Forms 2.0, Web Storage, Canvas, Web Sockets;
CSS3 – Colors, Gradients, Text, Transform.</td></tr>
<tr><td>Module 2</td><td>Responsive web design</td><td>Project</td><td colspan="5">Programming</td><td>15 Sessions [7L+8P]</td></tr>
<tr><td colspan="9">Topics:
BootStrap for Responsive Web Design; JavaScript – Core syntax,JavaScript – Core syntax,HTML DOM, objects, classes,HTML DOM, objects, classes, Async; Ajax ,jQuery Introduction.
Assignment: Design and develop a website that can actively keep track of entry-exit information of a housing society.</td></tr>
<tr><td>Module 3</td><td>JavaScript Frameworks</td><td>Project</td><td colspan="5">Programming</td><td>20Sessions [10L+10P]</td></tr>
<tr><td colspan="9">Topics:
Setting up Development & Build Environment: Node.js and NPM; Introduction to TypeScript;
Working with OOP concepts with TypeScript; Angular Fundamentals; Angular CLI; Introduction to TypeScript; Debugging Angular applications;Components& Databinding in Depth; Angular Directives; Using Services & Dependency Injection; Angular Routing; Observables; Handling Forms in Angular Apps; Output transformation using Pipes; Making Http Requests; Authentication & Route Protection; Dynamic Components;Angular Modules & Optimizing Angular Apps;AngularAnimations;Adding Offline Capabilities with Service Workers; React.js ; Developing single page application

Assignment: Develop a software tool to do inventory management in a warehouse.</td></tr>
<tr><td>Module 4</td><td>Fundamentals of DevOps and Project Management</td><td>Project</td><td colspan="5">Programming</td><td>10 Sessions [6L+4P]</td></tr>
</table>

Topics:
Introduction to Agile Methodology; Scrum Fundamentals; Scrum Roles, Artifacts and Rituals; Scrum Fundamentals; Scrum Roles, Artifacts and Rituals;DevOps – Architecture, Lifecycle, Workflow & Principles; DevOps Tools Overview – Jenkins, Docker, Kubernetes. Review of GIT source control. Deploying an Angular/React App; Unit Testing in Angular Apps (Jasmine, Karma).

Assignment: Develop a web-based application to book movies/events (like bookmyshow).

List of Laboratory Tasks:(7 X 2= 14 Sessions)
Experiment No. 1: [3 + 1 Practical Sessions]
Level 1: Familiarization of HTML and CSS basics.
Level 2: Create an HTML webpage showcasing biodata with CSS styling.

Experiment No. 2: [3 + 1 Practical Sessions]
Level 1: Design an interactive web page for a new restaurant using CSS3 features.
Level 2: Create a simple web form to gather user information.

Experiment No. 3: [4 + 1 Practical Sessions]
Level 1: Practice basic JavaScript exercises, including creating a canvas drawing application.
Level 2: Implement JavaScript exercises for form validation.

Experiment No. 4 [3 + 1 Practical Sessions]
Level 1: Create a student registration form using JavaScript.
Level 2: Design an RSVP form using Bootstrap form controls.

Experiment No. 5 [3 + 1 Practical Sessions]
Level 1: Create a responsive image grid using Bootstrap 5.
Level 2: Write a JavaScript program using AJAX to dynamically load content and implement jQuery effects like fading.
Experiment No. 6 [3 + 1 Practical Sessions]
Level 1: Create an AngularJS application module and controller in app.js.
Level 2: Design an "AngularJS Solar System Explorer" for planet data visualization.

Targeted Application & Tools that can be used:
Application Area is to Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers.
Professionally Used Software: Replit

Project work/Assignment:

Problem Solving: Design of Algorithms and implementation of programs.
Programming: Implementation of given scenario using Java.

Text Book:
T1. Fender, Young, "Front-end Fundamentals",Leanpub, 2015
T2. Northwood, Chris, "The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer",APress, 2018

References:
R1. Flanagan D S, "Javascript : The Definitive Guide" 7th Edition. 7th ed. O'Reilly Media; 2020.
R2. Alex Libby, Gaurav Gupta, and AsojTalesra. "Responsive Web Design with HTML5 and CSS3 Essentials",Packt Publishing, 2016
R3. Duckett J Ruppert G Moore J. "Javascript&Jquery : Interactive Front-End Web Development."; Wiley; 2014.
R4. Greg Sidelnikov, "React.js Book_ Learning React JavaScript Library", 1 edition, Scratch-River Tigris LLC 2016
R5. Web Reference:
https://www.youtube.com/watch?v=JGNTYXkVCVY&list=PLd3UqWTnYXOkTSBCBNyyhxo_jxlY_uTWA&index=2

Course Code: CSE3427	Course Title: Java Full Stack Development	L- P- C	2	2	3
Version No.	1.0				
Course Pre-requisites	Nil				
Anti-requisites	CSE2260				
Course Description	This advanced level course enables students to perform full stack development using Java, with emphasis on employability skills. The key technologies used for Full Stack development is based on either Java technology or .NET technology. In this course, the focus is on using Java, and the related technologies/tools like Java EE, Java Persistence, Hibernate, Maven, Spring Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.				
Course Objectives	This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.				
Course Outcomes	On successful completion of the course the students shall be able to: 1] Practice the use of Java for full stack development [Application] 2] Show web applications using Java EE. [Application] 3] Solve simple applications using Java Persistence and Hibernate [Application] 4] Apply concepts of Spring to develop a Full Stack application. [Application] 5] Employ automation tools like Maven, Selenium for Full Stack development. [Application]				
Course Content:					

Module 1	Introduction	Project	Programming	03 Sessions
Topics: Review of Java; Advanced concepts of Java; Java generics; Java IO; New Features of Java. Unit Testing tools.				
Module 2	Java EE Web Applications	Project	Programming	05 Sessions
Topics: Introduction to Eclipse & Tomcat; JSP Fundamentals; Reading HTML form Data with JSP; State Management with JSP; JSP Standard Tag Library - Core & Function Tags; Servlet API Fundamentals; ServletContext, Session, Cookies; Request Redirection Techniques; Building MVC App with Servlets & JSP; Complete App - Integrating JDBC with MVC App Assignment: Develop an application for managing HR policies of a department.				
Module 3	Java Persistence using JPA and Hibernate	Project	Programming	06 Sessions
Topics: Fundamentals of Java Persistence with Hibernate; JPA for Object/Relational Mapping, Querying, Caching,				

Performance and Concurrency; First & Second Level Caching, Batch Fetching, Optimistic Locking & Versioning; Entity Relationships, Inheritance Mapping & Polymorphic Queries; Querying database using JPQL and Criteria API (JPA) Assignment: Design and develop a website that can actively keep track of entry-exit information of a housing society..				
Module 4	Spring Core	Project	Programming	10 Sessions
Topics: Spring Core, Spring MVC, Spring Boot REST API; Understanding Spring Framework; Using Spring MVC; Building a Database Web App with Spring and Hibernate o Spring AOP (Aspect Oriented Programming); Implementing Spring Security; Developing Spring REST API; Using Spring Boot for Rapid Development Assignment: Develop a software tool to do inventory management in a warehouse.				
Module 5	Automation tools	Project	Programming	06 Sessions
Topics: Introduction to Automation Tools; Apache Maven: Maven Fundamentals, Software Setup - Commandline and Eclipse, pom.xml and Directory Structure, Multi-Module Project Creation, Scopes, Dependency Management, Profiles; Functional/BDD Testing using Selenium, Selenium Fundamentals and IDE, Selenium WebDriver, Installation and Configuration, Locating WebElements, Driver Commands, WebElement Commands Assignment: Illustrate the use of automation tools in the development of a small software project.				
Targeted Application & Tools that can be used: Application Area is to Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers. Professionally Used Software: Eclipse, NetBeans, Hibernate, Selenium, Maven, GIT.				
Project work/Assignment:				
Problem Solving: Design of Algorithms and implementation of programs. Programming: Implementation of given scenario using Java.				
Text Book: T1. Fender, Young, “Front-end Fundamentals”, Leanpub, 2015				
References R1. Soni, Ravi Kant. “Full Stack AngularJS for Java Developers: Build a Full-Featured Web Application from Scratch Using AngularJS with Spring RESTful.” , Apress, 2017. R2. Mardan, Azat. “Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB.”, Apress, 2015				

Course Code: CSE3428	Course Title: .NET Full Stack Development	L- P- C	2	2	3
Version No.	1.0				
Course Pre-requisites	Nil				
Anti-requisites	CSE2260				

<table>
<tr><td>Course Description</td><td colspan="4">This advanced level course enables students to perform full stack development using .NET, with emphasis on employability skills. The key technologies used for Full Stack development is based on either Java technology or .NET technology. In this course, the focus is on using .NET and the related technologies/tools like C#, ASP.NET, Entity Framework Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.</td></tr>
<tr><td>Course Objectives</td><td colspan="4">This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.</td></tr>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of the course the students shall be able to:
1] Practice the use of C# for developing a small application [Application]
2] Show web applications using Entity Framework. [Application]
3]Solve simple web applications that use SQL and ASP.NET [Application]
4] Apply concepts of ASP.NET to develop a Full Stack application. [Application]</td></tr>
<tr><td>Course Content:</td><td colspan="4"></td></tr>
<tr><td>Module 1</td><td>C# Programming for Full Stack Development</td><td>Project</td><td>Programming</td><td>10 Sessions</td></tr>
<tr><td colspan="5">Topics:
.NET Framework Fundamentals, Visual Studio IDE Fundamentals, C# Language Features, Working with arrays and collections, Working with variables, operators, and expressions, Decision and iteration statements, Managing program flow and events, Working with classes and methods, OOP concepts, Properties, Auto Implemented, Delegates, Anonymous Methods and Anonymous Types, Extension methods, Sealed Classes/Methods, Partial Classes/Methods, Asynchronous programming and threading, Data validation and working with data collections including LINQ, Handling errors and exceptions, Working with Files, Unit Testing – Nunit framework
Assignment: Develop a small application for managing library using C#.</td></tr>
<tr><td>Module 2</td><td>Entity Framework Core 2.0</td><td>Project</td><td>Programming</td><td>06 Sessions</td></tr>
<tr><td colspan="5">Topics:
Entity Framework Core 2.0 Code First Approach; Introduction To Entity Framework and EDM; Querying the EDM; Working With Stored Procedures; Advanced Entity Framework - DbContext [EF6]; Advanced Operations; Performance Optimization; Data Access with ADO.NET
Assignment: Develop an application for managing HR policies of a department.</td></tr>
<tr><td>Module 3</td><td>ASP.NET</td><td>Project</td><td>Programming</td><td>06 Sessions</td></tr>
<tr><td colspan="5">Topics:
ASP.NET Core, ASP.Net Core 3.1 MVC, ASP.NET Core Middleware and Request pipeline, Review of SQL using MS SQL, Working With Data In Asp.Net, Razor View Engine, State Management In Asp. Net MVC & Layouts;

Assignment: Develop a web application to mark entry/exit of guests in a building.</td></tr>
<tr><td>Module 4</td><td>ASP.NET</td><td>Project</td><td>Programming</td><td>08 Sessions</td></tr>
<tr><td colspan="5">Topics:
Introduction To Models, Validations In Asp.Net MVC, Authentication and Authorization In Asp.Net MVC, Advanced Asp. Net MVC - Ajax Action Link In MVC, Advanced Asp.Net MVC - Ajax Forms In MVC, Microsoft Testing Framework – Unit Testing the .NET Application
Assignment: Develop a software tool to do inventory management in a warehouse.</td></tr>
<tr><td colspan="5">Targeted Application & Tools that can be used:</td></tr>
</table>

Application Area is to Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers. Professionally Used Software: Visual Studio
Project work/Assignment:
Problem Solving: Design of Algorithms and implementation of programs. Programming: Implementation of given scenario using .NET.
Text Book: T1. Fender, Young, "Front-end Fundamentals", Leanpub, 2015 T2. Valerio De Sanctis, "ASP.NET Core 5 and Angular: Full-stack web development with .NET 5 and Angular 11", 4th Edition, Packt, 2021.
References R1. Benjamin Perkins, Jon D. Reid, "Beginning C# and .NET", Wiley, 2021 Reid, 2021. R2. Piotr Gankiewicz, "Full Stack .NET Web Development", Packt Publishing, 2017. R3. Tamir Dresher, Amir Zuker, Shay Friedman, "Hands-On Full-Stack Web Development with ASP.NET Core", Packt Publishing, 2018. R4. Dustin Metzgar, "Exploring .NET core with microservices, ASP.NET core, and Entity Framework Core", Manning, 2017.

Course Code: CAI3427	Course Title: Language Models for Text Mining		L-T-P-C	2	0	2	3
Version No.	1.0						
Course Pre-requisites	CSE2264						
Anti-requisites	NIL						
Course Description	This course introduces the basics of Text Mining and Natural Language Processing. The course will teach students different concepts such as text mining, NLP, Sequence Labeling, etc. Topics: Text Mining, NLP, Tokenization, Lemmatization, Stemming, One-hot encoding, Language modelling, Bag-of-words, Term-document Matrix, Cosine similarity, Viterbi Algorithm, etc.						
Course Objectives	The objective of the course is EMPLOYBILITY of student by using EXPERIENTIAL LEARNING techniques.						
Course Out Comes	On successful completion of this course the students shall be able to: Process text data to derive information from text. [Apply] Apply insights from textual information to real-world business. [Apply] Develop solutions for a particular NLP problem using different machine learning and deep learning techniques. [Apply] Utilize different NLP tools and packages. [Apply]						
Course Content:							

Module 1	Text Mining	Adversarial Quiz Tests	Module Tests	No. of Sessions: 09

Introduction to Text Mining. Text Mining vs. NLP. Text Mining Algorithms. Steps in Text Mining - Extraction, Preprocessing, Analysis and Evaluation. Lexical Resource Creation (NEW). Data collection. String Manipulation to Clean Data. Natural Language Processing. Research Paradigms in NLP. Sequential Data. Sequence Labeling (NEW). Viterbi Algorithm (NEW). Corpus. Building a HMM using a Corpus (NEW). Unknown word handling (NEW).				
Module 2	Text Preprocessing	Adversarial Quiz Tests	Module Tests	No. of sessions: 06
Introduction to Preprocessing. Tokenization. Stop Words Removal. Lemmatization and Stemming. PoS Tagging. Integer Encoding. Padding. One-Hot Encoding.				
Module 3	Text Representations	Adversarial Quiz Tests	Module Tests	No. of sessions: 08
Language Modeling. N-Gram Language Model. Bag-of-Words Model. Term-Document Matrix. Term Frequency. Inverse Document Frequency. TF-IDF. Cosine Similarity. Naive Bayes Classifier using Bag-of-Words. Topic Modeling. Latent Semantic Analysis. Singular Value Decomposition. Truncated SVD and Topic Vector. LDA Algorithm.				
Module 4	Natural Language Processing with Keras	Adversarial Quiz Tests	Module Tests	No. of Sessions: 06
Word Embeddings vs. One-Hot Encoding. Contextual Bag of Words (CBOW). Skipgram. Deep Learning for Document Classification.				

List of Laboratory Tasks:
Experiment No. 1: File Handling
Level 1: Read text files using Python and extract meaningful content.
Level 2: Parse text files using Python to preprocess the data for NLP tasks.

Experiment No. 2: Introduction to NLP Tools
Level 1: Install and use NLTK for basic text processing.
Level 2: Install and use SpaCy for tokenization, PoS tagging, and Named Entity Recognition.

Experiment No. 3: Corpus Cleaning Techniques
Level 1: Use NLTK for corpus cleaning techniques such as tokenization, stopword removal, and stemming.
Level 2: Prepare cleaned text data for downstream NLP tasks like classification or translation.

Experiment No. 4: Word Vector Usage
Level 1: Download and use pre-trained word vectors (e.g., Word2Vec, GloVe, or FastText).
Level 2: Compute similarity between two words, find the most similar word, and complete word analogies (e.g., king - man + woman = queen).

Experiment No. 5 & 6: Language Identification
Level 1: Build a simple language identifier using Bag-of-Words (BoW) features.
Level 2: Predict the language of a given text using the trained model.

Experiment No. 7 & 8: Lexical Simplification
Level 1: Implement a lexical simplifier to replace complex words with simpler alternatives.
Level 2: Generate a simplified version of a given word or sentence while preserving meaning.

Experiment No. 9 & 10: Sentiment Analysis
Level 1: Implement a basic sentiment classifier using a lexicon-based or machine learning approach.
Level 2: Compare the performance of an existing sentiment classifier (e.g., VADER, TextBlob, or a pre-trained Transformer model).

Experiment No. 11: Named Entity Recognition (NER)
Level 1: Extract named entities from a text using NLTK.
Level 2: Extract named entities using SpaCy and compare results.

Experiment No. 12 & 13: Implement a Hidden Markov Model (HMM)

Level 1: Implement a generic HMM for sequence prediction. Level 2: Calculate the forward probability of a given sequence using HMM. Experiment No. 14: Linguistic HMM Level 1: Develop a Hidden Markov Model (HMM) for NLP tasks such as PoS tagging. Level 2: Evaluate the performance of the HMM on a specific NLP task (e.g., Named Entity Recognition or Chunking). Experiment No. 15: Machine Translation Level 1: Implement Machine Translation (MT) using a pre-trained model from Hugging Face Transformers. Level 2: Evaluate the quality of MT output via Round-Trip Translation (translate text to another language and back to check accuracy).
Targeted Application & Tools that can be used: Google Colab Python IDEs like PyCharm
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course Group project on some NLP Task like text classification (Creating a Simple Text Classifier: Use Scikit-learn to classify positive vs. negative reviews from a dataset), sentiment analysis, etc.
Textbook(s): Daniel Jurafsky, James H. Martin. “Speech and Language Processing: An Introduction to Natural Language Processing”, Computational Linguistics and Speech, Pearson Publication, 2025 (3rd Edition Draft). Aditya Joshi, Pushpak Bhattacharyya. “Natural Language Processing”, Wiley Publication, 2023 (1st Edition).
References: R1. Chris Manning and Hinrich Schutze, “Foundations of Statistical Natural Language Processing”, 1st Edition, MIT Press. 1999. R2. Pawan Goyal. “Natural Language Processing”. 1st Edition, 2016. Weblinks W1. E-Book link or R2: https://drive.google.com/file/d/10nbwAJd-dv6htOOZVBgAvLd1WscI0RqC/view W2. Web Resource for T1: https://web.stanford.edu/~jurafsky/slp3/ - VERY VERY IMPORTANT!!! W3. NPTEL Courses: https://nptel.ac.in/courses/106106211 CMI), https://nptel.ac.in/courses/106105158 (IIT Kgp), https://nptel.ac.in/courses/106101007 (IITB), https://nptel.ac.in/courses/106105572 (IIT Kgp - NEW)

Course Code: CAI3428	Course Title: Practical Deep Learning with TensorFlow	L- T-P- C	2	0	2	3
Version No.	1.0					
Course Pre-requisites	CSE2264					
Anti-requisites	NIL					
Course Description	This course introduces students to the concepts of deep neural networks and state of the art approaches to develop deep learning models. In this course students will be given an exposure to the details of neural networks as well as deep learning architectures and to develop end-to-end models for such tasks. It will help to design and develop an application-specific deep learning models and also provide the practical knowledge handling and analyzing end user realistic applications.					
Course Objective	This course is designed to improve the learners EMPLOYABILITY SKILLS by using EXPERIENTIAL LEARNING techniques.					

<table>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of this course the students shall be able to:
Implement backpropagation and gradient descent techniques to train neural networks effectively. (Apply)
Build and train deep learning models using Python libraries such as TensorFlow and Keras for real-world applications. (Apply)
Utilize deep learning techniques for image classification, object detection, sentiment analysis, and language modeling. (Apply)</td></tr>
<tr><td colspan="5">Course Content:</td></tr>
<tr><td>Module 1</td><td>Basics of Neural Networks</td><td>Assignment</td><td></td><td>18[8L+10P] Sessions</td></tr>
<tr><td colspan="5">Topics:
Understanding Perceptron with Excel, Understanding Multilayer Perceptron with Excel, From Multilayer Perceptron to Deep Learning, Error Backpropagation and Gradient Descent to reduce errors, Activation Functions, Deep Learning, Problems with Deep Learning with solutions.</td></tr>
<tr><td>Module 2</td><td>TensorFlow Basics</td><td>Assignment</td><td></td><td>14[7L+7P] Sessions</td></tr>
<tr><td colspan="5">Topics:
Introduction to TensorFlow, TensorFlow dataset, Machine Learning with TensorFlow</td></tr>
<tr><td>Module 3</td><td>Deep Learning methods with Tensor Flow and Keras</td><td>Assignment</td><td></td><td>14[6L+8P] Sessions</td></tr>
<tr><td colspan="5">Topics:
Main Features of TensorFlow, Keras basics, AI with Keras.</td></tr>
<tr><td colspan="5">Project work/Assignment:</td></tr>
<tr><td colspan="5">Assignment 1 on (Module 1 and Module 2)
Assignment 2 on (Module 3)</td></tr>
<tr><td colspan="5">List of Laboratory Tasks:

Lab 1: Working with Deep Learning Frameworks
Objective: Explore various Deep Learning Frameworks
Tasks: Identify deep learning frameworks (Keras, Tensorflow, Matplotlib, etc)
Activity: Practice with various methods available in DL Frameworks to develop a Model.

Lab 2: Build a Basic Artificial Neural Network
Objective: Create a ANN with DL frameworks.
Task: Identify suitable ANN Layers using Keras and Tensorflow.
Activity: Design a basic Artificial Neural Networks using Keras with TensorFlow (pima-indians-diabetes)

Lab 3: Build a MultiLayer Perceptron
Objective: Create a MLP for classification task.
Task: Identify suitable model for house price prediction.
Activity: Design a MLP for implementing classification and fine-tuning using House price.csv

Lab 4: Create a Tensor in TensorFlow using List or Numpy array.
Objective: To understand how to create a tensor in TensorFlow using a Python list or NumPy array
Task: Create a simple tensor using both a Python list and a NumPy array in TensorFlow.
Activity: Create a tensor using a Python list and Numpy array

Lab 5: Apply math operations on tensor using various mathematical functions.</td></tr>
</table>

Objective: To learn how to apply mathematical operations on tensors using various TensorFlow mathematical functions.
Task: Perform basic mathematical operations (addition, subtraction, multiplication, division) and advanced functions (square, square root, exponential) on tensors.
Activity: Perform basic math operations: Add, Subtract, Multiply, Divide and Apply advanced math functions: Square, Square root, Exponential.

Lab 6: Connecting two tensors in dataset.
Objective: Combine two tensors using concatenation and stacking operations in TensorFlow.
Task: Combine two tensors using concatenation and stacking operations in TensorFlow
Activity: Concatenate them along a specific axis and Stack them along a new axis.

Lab 7: Building dataset from a file stored in a local drive
Objective: To learn how to build a dataset in TensorFlow from a file stored in a local drive.
Task: Load a dataset from a CSV file stored on the local drive and process it using TensorFlow
Activity: Load the file using TensorFlow's tf.data API and Process the dataset (e.g., convert it into tensors)

Lab 8: Loading Dataset from TensorFlow.dataset Library
Objective: To learn how to load a dataset from the tensorflow_datasets library and use it in machine learning models.
Task: Load a dataset from TensorFlow Datasets (tfds), preprocess it, and display sample data
Activity: Load a dataset (e.g., MNIST, CIFAR-10, IMDB Reviews) and Split the dataset into training and testing sets.

Lab 9: Build a Convolutional Neural Network
Objective: Create a CNN model.
Task: Build CNN architecture for Dog-Cat classification problem.
Activity: Implement a Convolution Neural Network (CNN) for dog/cat classification problem using keras.

Lab 10: Build a Time-Series Model
Objective: Create a RNN and LSTM Model
Task: Build RNN/LSTM Model for predicting time series data.
Activity Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU notes.

REFERENCE MATERIALS:

TEXTBOOKS

François Chollet, "Deep Learning with Python", 2nd Edition, Manning Publications, 2022
Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2017.

REFERENCES

Amlan Chakrabarti Amit Kumar Das, Saptarsi Goswami, Pabitra Mitra , "Deep Learning", Pearson Publication, 2021.
David Foster, "Generative Deep Learning" O'Reilly Publishers, 2020.
John D Kellehar, "Deep Learning", MIT Press, 2020.

JOURNALS/MAGAZINES

IEEE Transactions on Neural Networks and Learning Systems
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385 IEEE Transactions on Pattern Analysis and Machine Intelligence
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34http://ijaerd.com/papers/special_papers/IT032.pdf International Journal of Intelligent Systems https://onlinelibrary.wiley.com/journal/1098111x

SWAYAM/NPTEL/MOOCs:

Swayam Nptel – Deep Learning – IIT Ropar https://onlinecourses.nptel.ac.in/noc21_cs35/preview
Coursera – Neural Networks and Deep Learning Andrew Ng
Coursera - Neural Networks for Machine Learning by Geoffrey Hinton in Coursera

Course Code: CAI3429	Course Title: Deep Learning Techniques for Computer Vision	L-T- P-C	2	0	2	3
Version No.	1.0					
Course Pre-requisites	MAT2402					
Anti-requisites	NIL					
Course Description	This course covers the fundamentals and advanced concepts of deep learning for computer vision applications. Students will explore convolutional neural networks (CNNs), object detection, image segmentation, and generative models. Hands-on lab experiments will reinforce theoretical concepts using frameworks like TensorFlow and PyTorch.					
Course Out Comes	On successful completion of the course the students shall be able to: Understand the Fundamentals of Deep Learning for Vision Explain the core concepts of neural networks and deep learning architectures for image processing. Implement and optimize convolutional neural networks (CNNs) for classification tasks. Apply Object Detection and Image Segmentation Techniques Implement and analyze state-of-the-art object detection algorithms such as YOLO, Faster R-CNN, and SSD. Develop and evaluate image segmentation models like U-Net and Mask R-CNN. Explore Advanced Deep Learning Techniques for Vision Utilize Vision Transformers (ViTs) and attention mechanisms for image classification. Generate and manipulate images using Generative Adversarial Networks (GANs). Deploy and Optimize Deep Learning Models for Real-World Applications					
Course Content:						

Module 1	Fundamentals of Deep Learning for Vision	Assignment	Practical	No. of Classes:8
Introduction to Deep Learning & Neural Networks, Convolutional Neural Networks (CNNs) Architecture Backpropagation & Optimization in CNNs, Transfer Learning & Pretrained Models.				
Module 2	Object Detection & Image Segmentation	Assignment	Practical	No. of Classes:14
Introduction to Object Detection (R-CNN, SSD, YOLO), Region Proposal Networks (Faster R-CNN) Semantic & Instance Segmentation (U-Net, Mask R-CNN), Real-time Object Detection Applications				
Module 3	Advanced Topics in Vision	Assignment	Practical	No. of Classes:8
Attention Mechanisms & Vision Transformers (ViTs), Generative Adversarial Networks (GANs) for Image Generation, Self-supervised Learning for Vision, Multi-modal Learning (CLIP, DALL·E)				
Module 4	Applications & Deployment	Assignment	Practical	No. of Classes:8
Edge AI & Mobile Deployment (TensorFlow Lite, ONNX), Adversarial Attacks & Robustness in Vision Models, Explainability & Interpretability of Vision Models, Case Studies & Industry Applications				
Lab Experiments are to be conducted on the following topics:- Lab Sheet 1: Keras Sequential API model Read in the data and explore				

Define a Sequential API model
Define the hyperparameters and optimizer
Train the model and visualize the history
Testing
Keras Functional API model:
Define a Functional API model
Train the model and visualize the history

Lab Sheet 2:
Softmax regression with Keras
Read in the data and prepare
Define a Sequential API model
Define the hyperparameters and optimizer
Train the model and visualize the history
Testing

Lab Sheet 3:
Convolutional Neural Network with Keras (grayscale images)
Read in the data:
Visualize the data:
Prepare the data:
Define a CNN model:
Define the hyperparameters and optimizer:
Train the model and visualize the history:
Testing:
Lab Sheet 4:
Convolutional Neural Network with Keras (color images):
Read in the data:
Visualize the data:
Prepare the data:
Define a CNN model:
Define the hyperparameters and optimizer:
Train the model and visualize the history:
Testing:
Lab Sheet 5:
Time series and prediction:
Read in the data and explore:
Apply the exponential smoothing method and predict
Recurrent neural network (RNN):
Pre-processing:
Do the necessary definitions: (Hyper parameters, Model,
Train the model:
Predict the future:
Lab Sheet 6:
Document classification with LSTM network:
Read in the data:
Explore the data:
Data preprocessing:
Define the model:
Define the optimizer and compile:
Train the model and visualize the history:
Testing:
Lab Sheet 7:

Document classification with LSTM network (Binary):
Read in the data:
Explore the data:
Data preprocessing:
Define the model:
Define the optimizer and compile:
Train the model and visualize the history:
Testing:
Lab Sheet 8:
Document classification with LSTM + CNN network (Binary):
Read in the data:
Explore the data:
Data preprocessing:
Define the model:
Define the optimizer and compile:
Train the model and visualize the history:
Testing:
Lab Sheet 9:
Softmax regression to recognize the handswritten digits:
Download the MNIST data:
Take a look at the dataset:
Do the necessary definitions:
Training and Testing:
Multi-layer neural network to recognize the handswritten digits:
Download the MNIST data:
Take a look at the dataset:
Do the necessary definitions:
Training and Testing:

Lab Sheet 10:
Object Detection using YOLOv5
Lab Sheet 11:
Image Segmentation using U-Net
Custom Object Detection using Faster R-CNN
Lab Sheet 12:
Implementing Vision Transformers for Image Classification
Generating Images using GANs (DCGAN, StyleGAN)
(Group Project)
Object Detection and Recognition:
Haar cascade object detection (e.g., face detection or object detection using pre-trained classifiers).
Feature-based object detection using techniques like Speeded-Up Robust Features (SURF) or Scale-Invariant Feature Transform (SIFT).
Deep learning-based object detection using Convolutional Neural Networks (CNNs) or You Only Look Once (YOLO) algorithm.
Optical Character Recognition (OCR):
Preprocessing of text images (e.g., binarization, noise removal, or skew correction).
Text localization using techniques like connected component analysis or Stroke Width Transform (SWT).
Character recognition using machine learning algorithms like Support Vector Machines (SVM) or Convolutional Neural Networks (CNNs).
Gesture Recognition:
Hand segmentation using techniques like background subtraction or skin color detection.
Feature extraction from hand regions (e.g., finger counting, hand shape descriptors).
Classification of gestures using machine learning algorithms (e.g., k-Nearest Neighbors or Support Vector Machines).

Tools/Software Required : OpenCV 4 Python 3.7 MATLAB
Text Books "Deep Learning for Computer Vision Image Classification, Object Detection and Face Recognition in Python" Jason Brownlee (2019) "Deep Learning for Computer Vision with python" Adrian Rosebrock (2017)
References Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. A foundational book covering deep learning principles, including CNNs, optimization, and generative models. Raschka, S., & Mirjalili, V. (2022). Machine Learning with PyTorch and Scikit-Learn. Packt Publishing. Covers practical deep learning techniques using PyTorch, including CNNs and transfer learning. Geron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd Edition). O'Reilly Media. Provides hands-on implementations of deep learning for computer vision using TensorFlow and Keras. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into Deep Learning. Available online (https://d2l.ai). Open-access book covering CNNs, object detection, and advanced vision techniques with PyTorch and TensorFlow. Chollet, F. (2021). Deep Learning with Python (2nd Edition). Manning Publications. Explains deep learning fundamentals and applications with Keras, including image classification and segmentation. Ballé, J., Laparra, V., & Simoncelli, E. P. (2017). Deep Learning for Computer Vision: A Brief Introduction. A concise introduction to CNNs, object detection, and generative models.

Rajanukunte, Yelahanka, Bengaluru 560 119