PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Program Regulations and Curriculum 2026-2030

BACHELOR OF TECHNOLOGY (B.Tech.) in

Information Science and Engineering

B. Tech. [ISE]
Based on Choice Based Credit System (CBCS) and Outcome Based Education (OBE)

Table of Contents

Clause No.	Contents	Page Number
PART A – PROGRAM REGULATIONS		
1.	Vision & Mission of the University and the School / Department	4
2.	Preamble to the Program Regulations and Curriculum	4
3.	Short Title and Applicability	5
4.	Definitions	5
5.	Program Description	6
6.	Minimum and Maximum Duration	7
7.	Programme Educational Objectives (PEO)	8
8.	Programme Outcomes (PO) and Programme Specific Outcomes (PSO)	9
9.	Admission Criteria (as per the concerned Statutory Body)	9
10.	Lateral Entry / Transfer Students requirements	10
11.	Change of Branch / Discipline / Specialization	12
12.	Specific Regulations regarding Assessment and Evaluation	13
13.	Additional clarifications - Rules and Guidelines for Transfer of Credits from MOOC, etc.	14
PART B: PROGRAM STRUCTURE		
14.	Structure / Component with Credit Requirements Course Baskets & Minimum Basket wise Credit Requirements	18
15.	Minimum Total Credit Requirements of Award of Degree	19
16.	Other Specific Requirements for Award of Degree, if any, as prescribed by the Statutory Bodies	19
PART C: CURRICULUM STRUCTURE		
17.	Curriculum Structure – Basket Wise Course List	20
18.	Practical / Skill based Courses – Internships / Thesis / Dissertation / Capstone Project Work / Portfolio / Mini project	23
19.	List of Elective Courses under various Specializations / Stream Basket	27
20.	List of Open Electives to be offered by the School / Department (Separately for ODD and EVEN Semesters).	32

21.	List of MOOC Courses	35
22.	Recommended Semester Wise Course Structure / Flow including the Program / Professional Elective Paths / Options	36
23.	Course Catalogue of all Courses Listed including the Courses Offered by other School / Department and Discipline / Program Electives	41

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 developing cutting-edge technology, towards enhancing Quality of Life.

1.4 Mission of Presidency School of Computer Science and Engineering

- Cultivate a practice-driven environment, with computing-based pedagogy, integrating theory and practice.
- Attract and nurture world-class faculty to excel in Teaching and Research, in the realm of Computing Sciences.
- Establish state-of-the-art computing facilities, for effective Teaching and Learning experiences.
- Promote Interdisciplinary Studies to nurture talent for global impact.
- Instill 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 CreditSystem (CBCS) with focus on Social Project Based Learning, Industrial Training, and Internship toenable the students to become eligible and fully equipped for employment in industries,choosehigherstudiesorentrepreneurship.

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, 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 2025.
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 2024-2025.

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, 2022-2026;*

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

gg. *"PSOE" means the Presidency School of Engineering;*

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;*

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 2022-2026 are subject to, and, pursuant to the Academic Regulations, 2021.These Program Regulations shall be applicable to the following ongoing Bachelor of Technology (B.Tech.) Degree Programs of 2022-2026 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);
15. B. Tech Robotics and Artificial Intelligence
16. B. Tech Artificial Intelligence and Data Science

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

5.2 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.

5.3 The effect of periodic amendments or changes in the Program Regulations, on the students admitted in earlier years, shall be dealt with appropriately and carefully, so as to ensure that those students are not subjected to any unfair situation whatsoever, although they are required to conform to these revised Program Regulations, without any undue favour or considerations

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 **Error! Reference source not found.** of 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.**Error! Reference source not found.** of Academic Regulations) in the prescribed maximum duration (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:

PEO1. Demonstrate as a Computer Engineering Professional.

PEO2. Engage in lifelong learning through research and professional development,

PEO3. Serve as a leader in the profession through consultancy, extension activities or entrepreneurship.

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 knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions.

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking 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:

PSO 1: Show a deep understanding of the principles of information science to explain essential concepts, solve computational problems using theory learnt, and create systems and applications that are both reliable and efficient.

PSO2: Utilize programming capabilities, software engineering tools, and subject knowledge to develop hands-on applications, supporting students in pursuing careers in software development, academic advancement, research, or entrepreneurial initiatives.

PSO3: Analyze and optimize information systems using appropriate methodologies to enhance performance and scalability.

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:

9.1 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.

9.2 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.

9.3 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.

9.4 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.

9.5 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.

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

9.7 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.

9.8 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 (5th and 6th 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 1st 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 3rd Semester (commencement of the 2nd Year) of the B.Tech. Program and culminating with the 8th Semester (end of the 4th 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 1st year (1st or 2nd 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 3rd Semester of the Program. i.e., the Program Structure and Curriculum from the 3rd to 8th 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 1st 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, 2024-2028, minus the number of Credits prescribed / accepted by the Equivalence Committee for the 1st Year (1st and 2nd 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. (Information Science and Engineering) is "N" Credits, and, if the total credits prescribed in the 1st Year (total credits of the 1st and 2nd Semesters) of the Program concerned is "M" Credits, then the *Minimum Credit Requirements* for the award of the B.Tech. in Information 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 1st Year (i.e., passed in all the Courses / Subjects prescribed for the 1st 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 University as per the rules and guidelines prescribed in the following Sub-Clauses:

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

10.2.1 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 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.

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

10.2.3 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.

10.2.4 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 12.5 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 **Error! Reference source not found.**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

S.No	Credit Structure	Type	CA		Mid-Term		End-term		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
		Marks	25	25	25	25	50	50	-	200	
3	1-0-4-3	Percentage	-	25%	5%	20%	10%	40%	-	100%	Mid-Term & End Term by School
		Marks	-	25	5	20	10	40	-	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 in School level
		Marks	-	50	-	-	-	-	50	100	
6	0-0-2-1	Percentage	-	100%	-	-	-	-	-	100%	Only CA in 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	

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 **Error! Reference source not found.** 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

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

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

b. 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 Clause12.6.1, 12.6.2) 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 **Error! Reference source not found.**) 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 Clause13.3 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 13.3 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 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**Error! Reference source not found.**.

Table 2: Durations and Credit Equivalence for Transfer of Creditsfrom 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

13.3.9 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 forthe award ofthe concernedDegree.

13.3.10 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.

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

The B.Tech. (Information Science and Engineering) Program Structure (2025-2029) totalling 162 credits. Table 7 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: B.Tech. (Information Science and Engineering) 2025-2029: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets

S.No.	Baskets	Credit Contribution
1	Humanities and Social Sciences including Management Courses (HSMC)	9
2	Basic Science Courses (BSC)	24
3	Engineering Science Courses (ESC)	23
4	Professional Core Courses (PCC)	64
5	Professional Elective Courses (PEC)	18
6	Open Elective Courses (OEC)	6
7	Project Work (PRW)	16
8	Mandatory Courses (MAC)	0
	Total Credits	**160 (Minimum)**

In the entire Program, the practical and skill based course component contribute to an extent of approximately 57% out of the total credits of 160 for B.Tech. (Information 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 a 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.

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*) (HSMC)
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	Credits	Contact Hours	Type of Skill	Pre-requisite
1	ENG1030	Fundamentals of Communication	1	0	2	2	3	S	-
2	ENG2030	Communication in Practice	1	0	2	2	3	S	
3	FIN1002	Essentials of Finance	3	0	0	3	3	S	-
4	APT4005	Aptitude for Employability	0	0	2	1	2	AT	-
5	PPS3018	Preparedness for Interview	0	0	2	1	2	AT	
Total			5	0	8	9	13		

Table 3.2 : List of Basic Science Courses (BSC)									
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	MAT1211	Linear Algebra, Differential Equations and Statistical Methods	3	1	0	4	4	F	-
2	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F	-
3	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F	-
4	MAT1212	Mathematical foundations for Machine Learning and Data Analysis	3	1	0	4	4	F	-
5	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S	-
6	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S	-
7	MAT2311	Vector Calculus, Numerical Methods and Sampling Distributions	3	1	0	4	4	EM	-
8	MAT2321	Discrete Mathematical Structures	3	1	0	4	4	F	-
Total			**18**	**4**	**4**	**24**	**26**		

Table 3.3 : List of Engineering Science Courses (ESC)									
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	MEC1006	Engineering Graphics	2	0	0	2	2	S	-
2	CSE1500	Computational Thinking using Python	2	0	2	3	4	S	-
3	ECE2022	Digital Design	2	0	0	2	2	F/S	-
4	ECE2052	Digital Design Lab	0	0	2	1	2	F/S	-
5	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S	-
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S	-
7	EEE1251	Basics of Electrical and	0	0	2	1	2	F/S	-

		Electronics Engineering Lab							
8	CSE2264	Essentials of AI	3	0	0	3	3	S/EM	-
9	CSE2265	Essentials of AI Lab	0	0	2	1	2	S/EM	
10	CSE2274	Competitive Programming and Problem Solving	0	0	4	2	4	S/EM	
2	ECE1512	Design Thinking in Practice	2	0	2	3	4	F	-
Total			16	0	14	23	30		

Table 3.4 : List of Program Core Courses

Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skills	Pre-requisites
1	CSE2200	Problem Solving using C	2	0	0	2	2	S	-
2	CSE2201	Problem Solving using C Lab	0	0	4	2	4	S	-
3	CSE2251	Data Communication and Computer Networks	3	0	0	3	3	S	-
4	CSE2252	Data Communication and Computer Networks Lab	0	0	2	1	2	S	-
5	CSE2253	Data Structures	3	0	0	3	3	S	-
6	CSE2254	Data Structures Lab	0	0	2	1	2	S	-
7	CSE2255	Object Oriented Programming Using Java	3	0	0	3	3	S/EM	-
8	CSE2256	Object Oriented Programming Using Java Lab	0	0	2	1	2	S/EM	-
9	CSE2257	Computer Organization and Architecture	3	0	0	3	3	S	-
10	CSE2260	Database Management Systems	3	0	0	3	3	S	-
11	CSE2261	Database Management Systems Lab	0	0	2	1	2	S	-
12	ISE2506	Machine Learning	3	0	0	3	3	S/EM	-
13	ISE2507	Machine Learning Lab	0	0	4	2	4	S/EM	-
14	CSE2287	Software Design and Development	3	0	0	3	3	S	-
15	CSE2262	Analysis of Algorithms	3	1	0	4	4	S	-
16	CSE2263	Analysis of Algorithms Lab	0	0	2	1	2	S	-
17	CSE2266	Theory of Computation	3	0	0	3	3	S	-
18	ISE2502	Information Retrieval	3	0	0	3	3	S	-
19	ISE2500	Software Testing and Quality Assurance	3	0	0	3	3	S/EM	-

20	ISE2501	Software Testing and Quality Assurance Lab	0	0	2	1	2	S/EM	-
21	CSE2258	Web Technologies	3	0	0	3	3	S/EM	-
22	CSE2259	Web Technologies Lab	0	0	2	1	2	S/EM	-
23	CSE2269	Operating Systems	3	0	0	3	3	S	-
24	CSE2270	Operating Systems Lab	0	0	2	1	2	S/EM	-
25	CSE2502	Cryptography and Network Security	3	0	0	3	3	S	-
26	ISE2503	Information Theory and Coding	3	0	0	3	3	S/EM	-
27	ISE2504	Image Processing and Computer Vision	3	0	0	3	3	S/EM	-
28	ISE2505	Image Processing and Computer Vision Lab	0	0	2	1	2	S/EM	-
		Total No. of Credits				**64**	**77**		

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

Table 3.5 : List of course in Project Work basket (PRW)

S.No	Course Code	Course Name	L	T	P	C
1	CSE7000	Internship	0	0	0	2
2	CSE7100	Mini Project	0	0	0	4
3	CSE7300	Capstone Project	0	0	0	10
		Total No. of Credits				**16**

Table 3.6 : 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	3	0	0	3	3	S/ EM	CSE2267	PEC
3	CSE3502	Computational Optimization for	3	0	0	3	3	S	CSE2267	PEC

		Intelligent Systems.								
4	CSE3503	Reinforcement Learning for AI Systems	2	0	2	3	3	S/EM	CSE2267	PEC
5	CSE3504	Computational Linguistics Natural Language Processing	3	0	0	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	3	0	0	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	MAT1003	PEC
6	CSE3511	Predictive Analytics and Applications	2	0	2	3	3	S/EM	MAT1003	PEC
7	CSE3512	Data Mining	3	0	0	3	3	S	MAT1003	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	CSE3535	Cloud Data Engineering	2	0	2	3	3	S	CSE2285	PEC
2	CSE3516	Federated Learning	2	0	2	3	3	S	CSE2285	PCC
3	CSE3517	Edge Computing	2	0	2	3	3	S/EM	CSE2285	PCC
4	CSE3518	Network Security and Firewall Management	2	0	2	3	3	S/EM	CSE2502	PEC
5	CSE3519	Information Security and	3	0	0	3	3	S	CSE2502	PEC

		Management								
6	CSE3520	Network Intrusion Detection and Prevention	3	0	0	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	3	0	0	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	FIN1002	PEC
2	CSE3524	Banking Technology	2	0	2	3	3	S/EM	FIN1002	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	FIN1002	PEC
5	CSE3527	Financial and Capital Markets	2	0	2	3	3	S	FIN1002	PEC
6	CSE3528	Blockchain Development and Programming	2	0	2	3	3	S	FIN1002	PEC
7	CSE3529	Statistics and Data Analysis for Finance	2	0	2	3	3	S	FIN1002	PEC
8	CSE3530	Financial Regulations and Compliances	3	0	0	3	3	S	FIN1002	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	3	0	0	3	3	S/ EM	CSE1006	PEC
2	CSE3532	Advanced Database Management Systems	2	0	2	3	3	S	CSE2260	PEC
3	CSE3533	Programming in C# and .NET	1	0	4	3	3	S	CSE2260	PEC
4	CSE3534	Rust Programming	2	0	2	3	3	S/ EM	CSE1006	PEC
Track - 6 Special Track										
Sl.No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of**	**Prerequisite**	**Type of course**

								Skill		
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	MAT2605	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	CSE1006	PEC
6	CSE3428	.Net Full Stack Development *	2	0	2	3	3	S/ EM	CSE2258	PEC
7	CSE3450	Introduction to NVIDIA Accelerated Computing and AI	2	0	2	3	4	S/ EM	Nil	PEC
8	CSE3451	NVIDIA Applied Artificial Intelligence and Generative AI	2	0	2	3	4	S/ EM	Nil	PEC
9	CSE3452	Advanced AI Systems and Generative Intelligence using NVIDIA Platforms	2	0	2	3	4	S/ EM	Nil	PEC
Track -7 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	CHE1018	Environmental Science	1	0	2	0	3	F	Nil	MAC
2	LAW1007	Indian Constitution and Professional Ethics for Engineers	1	0	0	0	1	F	Nil	MAC
3	CIV7601	Universal Human Values and Ethics	1	0	0	0	1	S	Nil	MAC

4	APT4002	Introduction to Aptitude	0	0	2	0	2	AT	Nil	MAC
5	APT4004	Aptitude Training - Intermediate	0	0	2	0	2	AT	Nil	MAC
6	APT4006	Logical and Critical Thinking	0	0	2	0	2	AT	Nil	MAC
7	APT4026	Aptitude for Employability	0	0	2	0	2	AT	Nil	MAC
8	PPS4027	Preparedness for Interview	0	0	2	0	2	AT	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										

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, 2021, 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, 2021). The same shall be prescribed in the Course Handout.

18.1 Internship / In-plant Training / Skill-based Program / International Immersion (IM) / Recognition of Prior Learning (RPL)

A student may undergo an Internship / In-plant Training / Skill-based Program / IM/ RPL for a period minimum 04 weeks in an industry / company, government bodies, academic / research institution or recognized online platforms offering certified skill-based programsduring the semester break between 4thand 5th semesters, subject to the following conditions:

18.1.1 The Internship / In-plant Training / Skill-based Program / IM/ RPL shall be conducted in accordance with the Internship Policy prescribed by the University from time to time.

18.1.2 Internship: student shall undergo internship, either in industry / company, academic / research organizations, government bodies, or international institutions. The objective is to provide practical exposure, industry insights, and real-world experience relevant to the student's field of study.

18.1.3 In-plant Training: student shall undergo training / industrial exposure program aimed at providing with practical insights into real-world working environments. The training may be conducted by industries / companies on-campus or through student visits to industries / companies, government bodies / institutions, or technical organizations.

18.1.4 Skill-based Program: student shall undergo a certified skill-based program of 30 hours / 04

weeks. Skill-based program should cater to Skill-Enhancement, Practical Focus, and Career orientation, Complementary to Curriculum or Industry relevant.

18.1.5 International Immersion (IM): student shall undergo IM aimed at providing global exposure through collaborations with foreign universities, industries, or research institutions. The International Immersion may include industry visits, expert interactions, and cultural exchange activities, enhancing students' international outlook, communication skills, and professional readiness.

18.1.6 Recognition of Prior Learning (RPL): student who shall undergo any formally recognize relevant prior work experience, internships, or project-based learning that meet the internship learning outcomes, thereby allowing students to earn internship credit without repeating equivalent practical training.

18.1.7 The number of Internships available for the concerned Academic Term. Further, the available number of Internships / In-plant training / Skill-based Program / IM / RPL shall be awarded to the students by the University on the basis of merit using the CGPA secured by the student and as per the selection criteria. Provided further, the student fulfils the criteria, as applicable, specified by the industry / company, government bodies, academic / research or through certified courses.

18.1.8 A student may opt for Internship / In-plant Training / IM / RPL in an industry / company, government bodies / academic / research institution, international bodies of her / his choice, subject to the condition that the concerned student takes the responsibility to arrange the Internship / In-plant Training/ IM / RPL on her / his own. Provided further, that the industry / company, government bodies, academic / research institution national or international offering such Internship / Training confirms to the University that the Internship shall be conducted in accordance with the Program Regulations and Internship Policy of the University / Rubrics.

18.1.9 A student undergoing RPL must submit an application with supporting documents such as experience letters, project reports, employer feedback, certifications, a self-reflection report etc. Application must be submitted before the commencement of the internship semester.

18.1.10 A student selected for an Internship / In-plant Training / Skill-based Program / IM / RPL in an industry / company, government bodies, academic / research institution shall adhere to all the rules and guidelines prescribed in the Internship Policy of the University.

The performance will be assessed and grades awardedas per the university's academic grading policy. It may be included as a non-GPA course (Grade-based Satisfactorily Completed/Not Completed). The student must secure a minimum of 50% of the total marks to be declared as having successfully completed the course.

18.1 Mini Project Work

A student may opt to do a Mini Project Work for a period of 6-8 weeks in an Industry / Company or academic / research institution or the University Department(s) during the 5th / 6th / 7th Semester as applicable, subject to the following conditions:

18.2.1 The Mini 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 mini project work in an Industry / Company or academic / research institution of her / his choice subject to the above mentioned condition (Sub-Clause 18.2.1). Provided further, that the Industry / Company or academic / research institution offering such mini project work

confirms to the University that the mini project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.2 Capstone Project

A student may undergo a Capstone Project for a period of 12-14 weeks in an industry / company or academic / research institution in the 7th / 8th Semester as applicable, subject to the following conditions:

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

18.2.2 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;

18.2.3 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.

18.2.4A 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 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 ProjectPolicy of the University.

18.3.5A 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.

18.3 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:

18.3.1 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.

19. List of Professional Elective Courses under various Specialisations / Stream Basket

Type of Skill
F - Foundation
S - Skill Development
EM – Employability
EN – Entrepreneurship
SS-Soft Skills
AT-Aptitude Training

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

Table 3.6 : 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	3	0	0	3	3	S/ EM	CSE2267	PEC
3	CSE3502	Computational Optimization for Intelligent Systems.	3	0	0	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	3	0	0	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	3	0	0	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	MAT1003	PEC
6	CSE3511	Predictive Analytics and Applications	2	0	2	3	3	S/EM	MAT1003	PEC
7	CSE3512	Data Mining	3	0	0	3	3	S	MAT1003	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	CSE3535	Cloud Data Engineering	2	0	2	3	3	S	CSE2285	PEC
2	CSE3516	Federated Learning	2	0	2	3	3	S	CSE2285	PCC
3	CSE3517	Edge Computing	2	0	2	3	3	S/EM	CSE2285	PCC
4	CSE3518	Network Security and Firewall Management	2	0	2	3	3	S/EM	CSE2502	PEC
5	CSE3519	Information Security and Management	3	0	0	3	3	S	CSE2502	PEC
6	CSE3520	Network Intrusion Detection and Prevention	3	0	0	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	3	0	0	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**	**Prerequisite**	**Type of**

								Skill		course
1	CSE3523	Introduction to Fintech	2	0	2	3	3	S/EM	FIN1002	PEC
2	CSE3524	Banking Technology	2	0	2	3	3	S/EM	FIN1002	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	FIN1002	PEC
5	CSE3527	Financial and Capital Markets	2	0	2	3	3	S	FIN1002	PEC
6	CSE3528	Blockchain Development and Programming	2	0	2	3	3	S	FIN1002	PEC
7	CSE3529	Statistics and Data Analysis for Finance	2	0	2	3	3	S	FIN1002	PEC
8	CSE3530	Financial Regulations and Compliances	3	0	0	3	3	S	FIN1002	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	3	0	0	3	3	S/ EM	CSE1006	PEC
2	CSE3532	Advanced Database Management Systems	2	0	2	3	3	S	CSE2260	PEC
3	CSE3533	Programming in C# and .NET	1	0	4	3	3	S	CSE2260	PEC
4	CSE3534	Rust Programming	2	0	2	3	3	S/ EM	CSE1006	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	MAT2605	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	CSE1006	PEC
6	CSE3428	.Net Full Stack Development *	2	0	2	3	3	S/ EM	CSE2258	PEC
7	CSE3450	Introduction to NVIDIA Accelerated Computing and AI	2	0	2	3	4	S/ EM	Nil	PEC
8	CSE3451	NVIDIA Applied Artificial Intelligence and Generative AI	2	0	2	3	4	S/ EM	Nil	PEC
9	CSE3452	Advanced AI Systems and Generative Intelligence using NVIDIA Platforms	2	0	2	3	4	S/ EM	Nil	PEC

Track -7 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	CHE1018	Environmental Science	1	0	2	0	3	F	Nil	MAC
2	LAW1007	Indian Constitution and Professional Ethics for Engineers	1	0	0	0	1	F	Nil	MAC
3	CIV7601	Universal Human Values and Ethics	1	0	0	0	1	S	Nil	MAC
4	APT4002	Introduction to Aptitude	0	0	2	0	2	AT	Nil	MAC
5	APT4004	Aptitude Training - Intermediate	0	0	2	0	2	AT	Nil	MAC
6	APT4006	Logical and Critical Thinking	0	0	2	0	2	AT	Nil	MAC
7	APT4026	Aptitude for Employability	0	0	2	0	2	AT	Nil	MAC
8	PPS4027	Preparedness	0	0	2	0	2	AT	Nil	MAC

		for Interview								
*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										

Track -6 Mandatory Non-Credited Course (Offered for Lateral Entry students in higher semester wherever applicable in MOOC mode)**								
LAW7601	Indian Constitution **	1	0	0	0	1	F	Nil
CHE7601	Environmental Studies **	1	0	0	0	1	F	Nil
CIV7601	Universal Human Values and Ethics **	1	0	0	0	1	F	Nil
PPS1025	Industry Readiness Program - I	0	0	2	0	2	SS	Nil
PPS1026	Industry Readiness Program - II	0	0	2	0	2	SS	Nil
APT4002	Introduction to Aptitude	0	0	2	0	2	AT	Nil
APT4004	Aptitude Training - Intermediate	0	0	2	0	2	AT	Nil
APT4006	Logical and Critical Thinking	0	0	2	0	2	AT	Nil
*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								

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

Baskets
HSMC - Humanities and Social Sciences (including Management courses) (HSMC)
BSC - Basic Science Courses ESC - Engineering Science Courses
PCC - Professional Core Course
PEC - Professional Elective Courses

OEC - Open Elective Courses
EEC - Employment Enhancement Courses
MAC - Mandatory Course

Table 3.8: 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
Chemistry Basket							
1	CHE1003	Fundamentals of Sensors	3	0	0	3	S
2	CHE1004	Smart materials for IOT	3	0	0	3	S
3	CHE1005	Computational Chemistry	2	0	0	2	S
4	CHE1006	Introduction to Nano technology	3	0	0	3	S
5	CHE1007	Biodegradable electronics	2	0	0	2	S
6	CHE1008	Energy and Sustainability	2	0	0	2	S
7	CHE1009	3D printing with Polymers	2	0	0	2	S
8	CHE1010	Bioinformatics and Healthcare IT	2	0	0	2	S
9	CHE1011	Chemical and Petrochemical catalysts	3	0	0	3	S
10	CHE1012	Introduction to Composite materials	2	0	0	2	S
11	CHE1013	Chemistry for Engineers	3	0	0	3	S
12	CHE1014	Surface and Coatings technology	3	0	0	3	S
13	CHE1015	Waste to Fuels	2	0	0	2	S
14	CHE1016	Forensic Science	3	0	0	3	S
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
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
3	CSE2004	Python Application Programming	2	0	2	3	S/ EM
4	CSE2005	Web design fundamentals	2	0	2	3	S/ EM/EN
5	CSE3111	Artificial Intelligence : Search Methods For Problem Solving	3	0	0	3	S/ EM/EN
6	CSE3112	Privacy And Security In Online Social Media	3	0	0	3	S/ EM/EN
7	CSE3113	Computational Complexity	3	0	0	3	S/ EM/EN
8	CSE3114	Deep Learning for Computer Vision	3	0	0	3	S/ EM/EN
9	CSE3115	Learning Analytics Tools	3	0	0	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
9	DES1004	Choices in Virtual Fashion	1	0	2	2	F
10	DES1005	Fashion Lifestyle and Product Diversity	1	0	2	2	F
11	DES1006	Colour in Everyday Life	1	0	2	2	F
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	-
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	-
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
2	DSA2002	Yoga for Health	2	0	0	2	S
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
2	FRL1005	Fundamentals of French	2	0	0	2	S
3	FRL1009	Mandarin Chinese for Beginners	3	0	0	3	S
Law Basket							
1	LAW1001	Introduction to Sociology	2	0	0	2	F
2	LAW2001	Indian Heritage and Culture	2	0	0	2	F
3	LAW2002	Introdcution to Law of Succession	2	0	0	2	F
4	LAW2003	Introduction to Company Law	2	0	0	2	F
5	LAW2004	Introduction to Contracts	2	0	0	2	F
6	LAW2005	Introduction to Copy Rights Law	2	0	0	2	F
7	LAW2006	Introduction to Criminal Law	2	0	0	2	F
8	LAW2007	Introduction to Insurance Law	2	0	0	2	F
9	LAW2008	Introduction to Labour Law	2	0	0	2	F
10	LAW2009	Introduction to Law of Marriages	2	0	0	2	F
11	LAW2010	Introduction to Patent Law	2	0	0	2	F
12	LAW2011	Introduction to Personal Income Tax	2	0	0	2	F
13	LAW2012	Introduction to Real Estate Law	2	0	0	2	F
14	LAW2013	Introduction to Trademark Law	2	0	0	2	F
15	LAW2014	Introduction to Competition Law	3	0	0	3	F
16	LAW2015	Cyber Law	3	0	0	3	F
17	LAW2016	Law on Sexual Harrassment	2	0	0	2	F

18	LAW2017	Media Laws and Ethics	2	0	0	2	F
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
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
8	MEC2005	Fundamentals of Aerospace Engineering	3	0	0	3	F
9	MEC2006	Safety Engineering	3	0	0	3	S/EM
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
13	MEC3071	Hybrid Electric Vehicle Design	3	0	0	3	S/EM
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
2	PET1012	Energy Sustainability Practices	3	0	0	3	FC
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
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
11	PHY2005	Science and Technology of Energy	3	0	0	3	FC
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

Management Basket- II							
1	MGT1001	Introduction to Psychology	3	0	0	3	F
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
5	MGT1005	Cross Cultural Communication	3	0	0	3	S/EM/ EN
6	MGT2001	Business Analytics	3	0	0	3	S/ EM/EN
7	MGT2002	Organizational Behaviour	3	0	0	3	F
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
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
18	MGT2014	Project Finance	3	0	0	3	EN / EM
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
22	MGT2019	Sales Techniques	3	0	0	3	S/EM/ EN
23	MGT2020	Marketing for Engineers	3	0	0	3	S/EM/ EN
24	MGT2021	Finance for Engineers	3	0	0	3	S/EM/ EN
25	MGT2022	Customer Relationship Management	3	0	0	3	S/EM/ EN
26	ECO1001	Introduction to Economics	3	0	0	3	S/EM/ EN
Media Studies Basket							
1	BAJ3050	Corporate Filmmaking and Film Business	0	0	4	2	EM
2	BAJ3051	Digital Photography	2	0	2	3	EM
3	BAJ3055	Introduction to News Anchoring and News Management	0	0	2	1	EM

21.List of MOOC Courses

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

21.1.2. 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.

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

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

21.1.5. 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.

21.1.6. 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.

21.1.7. The question paper shall be prepared by the MOOCs coordinator(s).

21.1.8. Students write the exams in online mode. These exams are scheduled and conducted by the School.

21.1.9. Results are evaluated by School and given to the Office of the Controller of Examinations (CoE).

21.1.10. 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

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

Semester Wise Course Grids/ Tables: First year - CYCLE 1

Sl. No.	Course Code	Course Name	L	T	P	C	Type of Skill/ Focus	Basket
Semester 1 - Chemistry Cycle			**14**	**1**	**10**	**19**		
1	MAT1211	Linear Algebra, Differential Equations and Statistical Methods	3	1	0	4	F	BSC
2	CSE1500	Computational Thinking using Python	2	0	2	3	S	ESC
3	CIV1200	Foundations of Integrated Engineering	2	0	0	2	S	ESC
4	CHE2501	Chemistry of Smart Materials	3	0	0	3	S	BSC
5	ENG1030	Fundamentals of Communication	1	0	2	2	S	HSMC
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	F/S	ESC

7	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	S	BSC
8	PPS1025	Industry Readiness Program – I	0	0	2	0	S	MAC
9	EEE1251	Basics of Electrical and Electronics Engineering Lab	0	0	2	1	F/S	ESC
							Total Contact Hrs: 25	
Semester 2 – Physics Cycle			**15**	**1**	**14**	**22**		
1	MAT1212	Mathematical foundations for Machine Learning and Data Analysis	3	1	0	4	F	BSC
2	CSE2200	Problem Solving using C	2	0	0	2	S	PCC
3	CSE2201	Problem Solving using C Lab	0	0	4	2	S	PCC
4	ENG2030	Communication in Practice	1	0	2	2	S	HSMC
5	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	F/S	BSC
6	MEC1006	Engineering Graphics	2	0	0	2	S	ESC
7	ECE2022	Digital Design	2	0	0	2	F/S	ESC
8	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	F/S	BSC
9	PPS1026	Industry Readiness Program – II	0	0	2	0	S	MAC
10	ECE2052	Digital Design Lab	0	0	2	1	F/S	ESC
11	ECE1512	Design Thinking in Practice	2	0	2	3	F	ESC
							Total Contact Hrs: 30	

Sl. No.	Course Code	Course Name	L	T	P	C	Contact hours	Type of Skill/ Focus	Course Caters to	Basket
Semester 3			19	1	10	23	30			
1	MAT2311	Vector Calculus, Numerical Methods and Sampling Distributions	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	CSE2264	Essentials of AI	3	0	0	3	3	S/EM	Nil	ESC
9	CSE2265	Essentials of AI Lab	0	0	2	1	2	S/EM	Nil	ESC
10	CSE2257	Computer Organization and Architecture	3	0	0	3	3	S	Nil	PCC
11	APT4002	Introduction to Aptitude	0	0	2	0	2	AT	Nil	MAC
12	CHE7601	Environmental Studies	1	0	0	0	1	S	-	MAC
Semester 4			**19**	**2**	**10**	**24**	**31**			
1	MAT2321	Discrete Mathematical Structures	3	1	0	4	4	F	Nil	BSC
2	CSE2260	Database Management Systems	3	0	0	3	3	S	Nil	PCC
3	CSE2261	Database Management Systems Lab	0	0	2	1	2	S	Nil	PCC
4	ISE2506	Machine Learning	3	0	0	3	3	S/EM	MAT1001	PCC
5	ISE2507	Machine Learning Lab	0	0	4	2	4	S/EM	MAT1001	PCC
6	CSE2287	Software Design and Development	3	0	0	3	3	S	Nil	PCC
7	CSE2262	Analysis of Algorithms	3	1	0	4	4	S	Nil	PCC
8	CSE2263	Analysis of Algorithms Lab	0	0	2	1	2	S	Nil	PCC

9	APT4004	Aptitude Training - Intermediate	0	0	2	0	2	AT	Nil	MAC
10	FIN1002	Essentials of Finance	3	0	0	3	3	S	Nil	HSMC
11	CIV7601	Universal Human Values and Ethics	1	0	0	0	1	MAC	12	MAC
MAC Courses offeringare subject to change										
Semester 5			19	0	8	23	27			
1	CSE2266	Theory of Computation	3	0	0	3	3	S	Nil	PCC
2	ISE2502	Information Retrieval	3	0	0	3	3	S	CSE2011	PCC
3	ISE2500	Software Testing and Quality Assurance	3	0	0	3	3	S/EM	CSE2024	PCC
4	ISE2501	Software Testing and Quality Assurance Lab	0	0	2	1	2	S/EM	CSE2024	PCC
5	CSE2258	Web Technologies	3	0	0	3	3	S/EM	Nil	PCC
6	CSE2259	Web Technologies Lab	0	0	2	1	2	S/EM	Nil	PCC
7	CSE2269	Operating Systems	3	0	0	3	3	S	Nil	PCC
8	CSE2270	Operating Systems Lab	0	0	2	1	2	S/EM	Nil	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	0	2	AT	Nil	MAC
12	LAW7601	Indian Constitution	1	0	0	0	1	F		MAC
Semester 6			**18**	**0**	**8**	**22**	**26**			
1	CSE2502	Cryptography and Network Security	3	0	0	3	3	S	CSE2251	PCC
2	ISE25	Information Theory and	3	0	0	3	3	S/EM	CSE2002	PCC

	03	Coding								
3	ISE2504	Image Processing and Computer Vision	3	0	0	3	3	S/EM	IST2503	PCC
4	ISE2505	Image Processing and Computer Vision Lab	0	0	2	1	2	S/EM	IST2503	PCC
5	CSE2274	Competitive Programming and Problem Solving	0	0	4	2	4	S/EM	Nil	ESC
6	CSEXXXX	Professional Elective – II	3	0	0	3	3	S/EM	Nil	PEC
7	CSEXXXX	Professional Elective – III	3	0	0	3	3	S/EM	Nil	PEC
8	XXXXXXX	Open Elective – I	3	0	0	3	3	S	Nil	OEC
9	APT4005	Aptitude for Employability	0	0	2	1	2	AT	Nil	HSMC
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	PPS3018	Preparedness for Interview	0	0	2	1	2	S	Nil	HSMC
6	CSE7100	Mini Project	0	0	0	4	0	S/EM	Nil	PRW
Semester 8			**0**	**0**	**0**	**10**	**0**			
1	CSE7300	Capstone Project	0	0	0	10	0	S/EM	Nil	PRW

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.

The Course Catalogues for the Courses offered in each basket are attached below:

Course Code: ENG1900	**Course Title: English for Technical Communication** **Type of Course: Theory (HSMC)**	**L-T-P-C**	**2**	**0**	**0**	**2**
VersionNo.	1.0					
CoursePre-requisites	NIL					
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, students shall be able to: **CO1:** Differentiate between general communication and technical communication. [Understand] **CO2:** Explain effective reading comprehension strategies for understanding technical texts. [Understand] **CO3:** Develop clear, concise, and well-structured technical reports and documents. [Apply] **CO4:** Deliver technical presentations and utilize peer feedback for continuous improvement. [Apply] **CO5:** Explain ethical practices in digital communication for professional contexts. [Understand]					
CourseContent:						
Module 1	**Technical communication**					**9 Sessions**
Topics: 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**					**12 Sessions**
Topics: Reading Comprehension. Note making & Notetaking. Content Analysis. **Activity:** Reading technical articles and answering comprehension questions, Note making techniques.						
Module 3	**Technical Writing**					**12 Sessions**
Topics: 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**	**12 Sessions**
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.		
TargetedApplication&Toolsthatcanbeused: YouTube,Instagram, Quill Bot, Grammarly, & Padlet.		
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/. https://ocw.mit.edu/.https://www.ted.com/talks.https://owl.purdue.edu/owl/subject_specific_writing/professional_technical_writing/digital_writing.html		
TopicsRelevantto"employability": TeamworkandCollaboration,CriticalThinkingandProblem- Solving **TopicsRelevantto"HumanValuesandProfessionalEthics":** Criticalreasoning,Inclusivityand Fairness		

Course Code: DES1146	**Course Title: Introduction to Design Thinking Type of Course: Theory (HSMC)**	**L-T-P-C**	**1**	**0**	**0**	**1**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	This course is designed to develop and familiarize the learners with the concepts of creating thinking and attain Entrepreneurship by using Participative Learning techniques.					

Course Outcomes	On successful completion of the course, students shall be able to: **CO1:** Explain the concept and significance of Design Thinking. [Understand] **CO2:** Differentiate between traditional problem-solving approaches and Design Thinking. [Understand] **CO3:** Identify the key stages involved in the Design Thinking process. [Remember]	
Course Content:	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.	
Module 1	**Introduction to Design Thinking**	**3 Sessions**
Topic 1) Definition and Introduction to Design Thinking 2) Understand the Design Thinking Process		
Module 2	**Design Thinking in Action**	**12 Sessions**
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.		
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		
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		
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: ENG2501	Course Title: Advanced English Type of Course: Theory (HSMC)	L- T- P- C	2	0	0	2
Version No.	2.0					
Course Pre-requisites	ENG1900 - English for Technical Communication					
Anti-requisites	NIL					
Course 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.					
Course Out Come	On successful completion of the course, students shall be able to: **CO1:** Identify the key elements of interpersonal and cross-cultural communication to effectively address communication challenges. [Understand] **CO2:** Demonstrate the ability to deliver both structured and impromptu speeches using effective speaking techniques. [Apply] **CO3:** Interpret textual and visual content using critical reading strategies to evaluate arguments, logic, and persuasion. [Analyze] **CO4:** Develop persuasive and analytical essays using effective argumentation and structured writing techniques. [Apply]					

Course Content

Module 1	**Foundations of Effective Communication**	**12 Sessions**

Topics:

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.

Module 2	**Mastering Speech Delivery**	**12 Sessions**

Topics:

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

Activities: Speech Writing, Impromptu Speech.

Module 3	**Critical Reading and Logical Analysis**	**12 Sessions**

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.		
Module 4	**Writing Effective Arguments**	**9 Sessions**
Topics: Understanding Critical Writing, Building Arguments (Pathos, Ethos, Logos), Techniques for Persuasion. **Activities:** Causes or Effects/Appeal Mash-Up/Debates on Controversial Topics, Opinion Writing.		
Targeted Application & Tools that can be used: Quizziz, Chatgpt, Gemini, Youtube, Instagram, Quillbot, Grammarly, Padlet		
References 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		

Course Code: FIN1002	**Course Title: Essentials of Finance** **Type of Course: Theory Course (HSMC)**	**L-T-P-C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a comprehensive understanding of financial management and taxation by enabling students to understand forms of business organization and their financial implications, analyze financial principles influencing decision-making, interpret financial statements to assess organizational performance, and determine tax liability under various heads of income as per the Income Tax Act, 1961.					

Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the basic concepts of finance, financial markets, and financial institutions. [Understand] **CO2:** Apply financial concepts and tools for effective business decision-making. [Apply] **CO3:** Analyze and interpret financial statements to assess the financial performance of an organization. [Analyze] **CO4:** Identify various heads of income and applicable deductions under the Income Tax Act, 1961. [Understand] **CO5:** Compute taxable income and determine tax liability based on relevant financial data and regulations. [Apply]	
Course Content:		
Module 1	**Introduction to Finance**	**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.		
Module 2	**Financial Management**	**18 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.		
Module 3	**Taxation**	**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.		
Targeted Application & Tools that can be used: **Textbooks, PPT, Spreadsheet Software (e.g., Microsoft Excel), Official Website of Income Tax**		

Department.
Project 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.
Text 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.
Reference 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.):** 1. https://presidencyuniversity.linways.com 2. https://onlinecourses.nptel.ac.in/noc24_ec01/preview 3. 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.

Course ode: APT4005	**Course Title: Aptitude For Employability Type of Course: Practical Only (HSMC)**	**L- T-P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
CoursePre-requisites	StudentsshouldhavethebasicconceptsofQuantitativeaptitude, Verbal ability along with its applications in real life problems.					
Anti-requisites	NIL					
CourseDescription	Thiscourseisdesignedtoenablethestudentstoenhancetheirskillsin quantitative aptitude and verbal ability skills.					
CourseObjective	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.					
CourseOutcomes	On successful completion of the course, students shall be able to: **CO1:** Recall fundamental mathematical concepts. [Remember] **CO2:** Identify the appropriate principle required to solve a given problem. [Understand] **CO3:** Apply suitable concepts to solve quantitative and logical reasoning problems. [Apply]					
CourseContent:						
Module1	**QuantitativeAbility**					**20 Sessions**

Topics: Number System, Percentage, Ratio andProportion,Average,MixtureandAllegation,Time andWork, Profit and Loss, Time Speed and Distance, Simple Interest and Compound Interest, Probability, Permutation and Combination.		
Module2	**VerbalAbility**	**10 Sessions**
Topics:-PartsofSpeech,SubjectVerbAgreement,SpottingError,ClozeTest,VerbalAnalogies, ReadingComprehension,Idioms&Phrases,ParaJumbles.		
TargetedApplication&Toolsthatcanbeused: Applicationarea:PlacementactivitiesandCompetitiveexaminations. Tools: LMS		
Evaluation	**ContinuousEvaluation** Topicwise evaluation	
TextBook 1. FasttrackobjectivebyRajesh Verma 2. R S Aggarwal 3. S.PBakshi		
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 DevelopmentthroughProblemsolvingTechniques.Thisisattainedthroughassessment componentmentionedincoursehandout.		

Course Code: PPS3018	**Course Title: Preparedness for Interview** **Type of Course: Practical Only Course (HSMC)**	**L- T- P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	StudentsareexpectedtounderstandBasicEnglish. Studentsshouldhavedesireandenthusiasmtoinvolve,participateandlearn.					
Anti-requisites	NIL					
Course Description	Thiscourseisdesignedtoenablestudentstounderstandsoftskillsconceptsto be corporate ready. The modules are set to improve self-confidence, communicate effectively and Prepare for the Interview to assist in employability.Ithelpsthestudentstogetaglimpseoftheacceptablecorporate 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,flippedclassrooms,continuousfeedback,role-playandmentoring.					
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, students shall be able to: **CO1:** Develop professional resumes. [Apply] **CO2:** Illustrate and present resumes effectively. [Understand] **CO3:** Apply acquired skills and knowledge for effective participation in group discussions and interviews. [Apply]					
Course Content:						

Module 1	**ResumeBuilding**	**10 Sessions**
Topics: Resume structure,use of templates, Do's and Don'ts, ATS methods, Cover Letter and Video Resume. **Activity:**Realworldscenarios		
Module 2	**GroupDiscussion**	**9 Sessions**
Topics:-Groupdiscussionas aplacement process,GDtechniqueslike Keyword.SPELT&POVofaffected parties.Do&Don'tofGD,Case-letsandtopicsforGD,practicesessionandevaluation **Activity:-**Realworldscenarios		
Module 3	**PersonalInterview**	**9 Sessions**
Topics: Placementprocess,Differentinterviewrounds,HRinterviews,Interviewquestionsanddesired answers, Different types of interviews, Do's and Don'ts. **Activity:-**RolePlay&Real-worldscenario		
Module 4	**Recap/Revision/FeedbackSession**	**2 Sessions**
TargetedApplication&Toolsthatcanbe used: 1. TED Talks 2. YouTube Links 3. RolePlayactivities		
Projectwork/Assignment:MentiontheTypeofProject/Assignmentproposedforthiscourse		
ContinuousIndividualAssessment		
TheTopicsrelatedtoSkillDevelopment: ArtOfPresentationandGroupDiscussionfor**SkillDevelopment**tthrough**ParticipativeLearning** Techniques. This is attained through assessment Component mentioned in course handout.		

Course Code: MAT2301	**Course Title: Calculus and Differential Equations** **Type of Course : BSC**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course 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.					
Course 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.					

Course Outcomess	On successful completion of the course the students shall be able to: CO1- Apply differentiation techniques (product rule, quotient rule, chain rule) to solve engineering problems. CO2- Evaluate indeterminate forms using L'Hôpital's Rule. CO3- Compute definite and indefinite integrals using standard techniques (substitution, integration by parts, partial fractions). CO4- Analyze functions of multiple variables, including limits, continuity, and partial derivatives. CO5- Solve higher-order linear ODEs with constant coefficients (homogeneous & non-homogeneous). CO6- Analyze real-world engineering problems (mechanical vibrations, electrical circuits, heat transfer) using ODEs.	
Course Content:		
Module 1	**Calculus**	**(13 Sessions)**
Introduction, Application, Limit, continuity, Rolle's theorem, Mean value theorems, Indeterminate forms and L'Hospital's rule; Definite and indefinite integrals, Reduction formulae (without proof) of $\int \sin^n x\,dx$, $\int \cos^n x\,dx$ and $\int \sin^m x \cos^n x\,dx$ (m and n are positive integers), evaluation of these integral with standard limits (0 to $\frac{\pi}{2}$), Beta and Gamma functions and their properties.		
Module 2	**Multivariable Calculus**	**(9 Sessions)**
Partial derivatives, total derivative, Taylor's and Maclaurin (self-study) theorems, Euler's Theorem, Jacobians, Maxima, minima and saddle points; Method of Lagrange multipliers (self-study)		
Module 3	**Multivariable Integrals**	**(9 Sessions)**
Double integrals (Cartesian), change of order of integration in double integrals, Change of variables (Cartesian to polar), Triple integrals (Cartesian); Applications: areas, volume -Center of mass and Gravity (constant and variable densities), orthogonal curvilinear coordinates, Simple applications involving cubes, sphere and rectangular parallelepipeds (self study)		
Module 4	**Differential Equations**	**(14 Sessions)**
Introduction, types of differential equation–variable separable (self study), Homogeneous, Exact, linear, Bernoulli'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, $e^{ax}v(x)$- Method of variation of parameters.		
Targeted Application & Tools that can be used: Differential 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. Differential 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.		

Tools Used: Python.
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. Erwin Kreyzig (2015), Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition 2. B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.
References: 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
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.

Course Code: PHY2501	**Course Title:** Optoelectronics and Quantum Physics **Type of Course: BSC**	**L-T-P-C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	**NIL**					
Course Description	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.					
Course Outcomess	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.	
Course Objective	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.	
Course Content:		
Module 1	**Electrical Conductivity of Solids and Semiconducting Devices**	**12 Sessions**
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		
Module 2	**Quantum Mechanics**	**12 Sessions**
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		
Module 3	**Quantum Computing**	**11 Sessions**
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.		
Module 4	**Lasers And Optical Fibers**	**10 Sessions**
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.		
Targeted 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. 2. Origin, excel and Mat lab soft wares for programming and data analysis.		
Project 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 1. Prepare a comprehensive report on non-conventional energy resources in Karnataka and their pros		

and cons.
2. Write a report on importance of quantum entanglement in supercomputers.

Text 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.

This is attained through the Assignment/ Presentation as mentioned in the assessment component in course handout.

Course Code: **MAT2402**	**Course Title: Probability and Statistics** **Type of Course: BSC**	**L-T- P- C**	**3**	1	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
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 Outcomess	On successful completion of the course the students shall be able to: CO1 – Use probability concepts and rules to determine outcomes of random experiments. CO2 – Apply random variable concepts and probability distributions to describe data					

	variations. CO3 – Employ the principle of least squares to determine best-fit models for experimental and engineering data. CO4 – Apply appropriate measures of central tendency and dispersion to analyze data sets. CO5- Apply concepts of sampling and hypothesis testing to make decisions about data. CO6- Employ test of significance procedures to solve data-based problems.

Course Content:

Module 1	**Introduction to Probability**	**15 Sessions**

Permutations, Combinations, Random Experiments, Types of events, Probability of an Event, Addition law, Conditional probability, Multiplication law, Bayes's Theorem and Problems.

Module 2	**Random Variables and Bivariate Distributions**	**8 Sessions**

Random Variables (discrete and continuous), Probability Mass/Density Functions, Mathematical Expectations, discrete probability distributions - Binomial distribution, Poisson distribution, continuous probability distributions - exponential distribution, normal distribution and problems.

Module 3	**Curve Fitting & Descriptive Statistics**	**10 Sessions**

Random Variables (discrete and continuous), Probability Mass/Density Functions, Mathematical Expectations, discrete probability distributions - Binomial distribution, Poisson distribution, continuous probability distributions - exponential distribution, normal distribution and problems.

Principle of Least Square. 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, Measure of Dispersion, 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	**Sampling Distributions**	**12 Sessions**

Random sampling, sampling distributions, various types of Sampling, 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, Small sample 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.

Course Code: PHY2504	**Course Title: Optoelectronics and Quantum Physics Lab** **Type of Course: BSC**	**L-T-P-C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course 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.**					
Course Outcomess	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.					
Course 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.					

List of Laboratory Tasks: **30 Sessions**

Experiment No. 1: Experimental errors and uncertainty using excel

Level 1: Calculation of accuracy and precision of a given data

Level 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.

Level 1: Determination of Wavelength of Laser
Level 2: Finding the particle size of lycopodium powder.

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

Level 1: To study I –V characteristics of the given Zener diode in reverse bias and to determine break down voltage.
Level 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.

Level 1: To determine the input resistance of a given transistor.
Level 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.

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

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

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

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

Experiment No. 11: Determination of dielectric constant of given materials.

Level 1: Determination of Stefan's constant
Level 2: compare the obtain results with other materials

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

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

Project 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

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

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	Nil					
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: Demonstrate the application of predicates, quantifiers, and logical connectives in formulating logical expressions. CO2: Apply counting techniques to tackle combinatorial problems CO3: Employ set-theoretic operations and relations to solve structured problems. CO4: Apply generating functions to transform recurrence relations into algebraic form.					

	CO5: Apply graph-theoretic and tree-based concepts to solve discrete mathematical problems. CO6: Apply shortest-path and minimum spanning tree algorithms to solve network problems.	
Course Content:		
Module 1	**Fundamentals of Logic**	**10 Sessions**
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**	**15 Sessions**
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 Sessions**
Cartesian Products and Relations, Functions,One-to-One, Onto Functions.The Pigeon-hole Principle, FunctionComposition 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 Sessions**
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**	**15 Sessions**
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&uniid=EBSCO95_30102024_54588		

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&uniid=EBSCO95_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.

Course Code: CHE2501	**Course Title: Chemistry of Smart Materials Type of Course: Theory (BSC)**	**L-T- P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	Theobjective 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 materialsand 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					
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using Participative learning techniques.					
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:**Apply the knowledge of computational chemistry for molecular structure, geometry and properties prediction. [Understand] **CO2:** Explain the role of organic, polymeric and hybrid materials in memory systems and photo and electroactive materials in display systems. [Understand] **CO3:** Explain the properties of Nanomaterials associated with sensors and smart devices. [Understand] **CO4:** Apply the knowledge of fundamentals of green chemistry and sustainable chemistry for E- waste management. [Apply] **CO5:**Analyze the principles of electrochemistry for performance evaluation of batteries, supercapacitors and photovolatics for sustainable technologies. [Analyze]					

Course Content:

Module 1	**Computational Chemistry**	**09 Sessions**

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

Module 2	**Materials for Memory and Display Systems**	**09 Sessions**

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 electroactivematerials , 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

Module 3	**Nanomaterials based Smart Sensors and Devices**	**09 Sessions**

Topics: Nanomaterials- Introduction,classificationbasedondimensionality,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

Module 4	**Sustainable Materials and Development**	**09 Sessio**

Topics: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 toBiomaterials- PLA , polymers in bio-compatible andbio-degradable materials - Poly(3-hydroxybutyrate-co-3-hydroxyvalerate) PHBV, synthesis and applications in drug delivery.

Self-learning topics:circular economy- case studies.

Module 5	**Energy Science**	**09 Sessions**

Topics:Battery technology: Fundamentalsof 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

Targeted Application & Tools that can be used:
Application areas are Data storage and analysis, logistics, Biomedicine, Energy, Environment and sustainability
Tools: Molview, chemdraw, excel etc

Project work/Assignment:

Assessment 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

Text Book

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

Reference Books

1. 1. Essentials of computational chemistry: theories and models, Christopher J Cramer, 2013, John Wiley& Sons. ISBN: 978-0-470-09182-1.
2. F. Jensen, Introduction to Computational Chemistry, 3rd edition, Wiley, 2017.
3. Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press,ISBN: 978-036-727-510-5.
4. Electrical and electronic devices, circuits and materials: Technological challenges and solutions.Tripathi, S. L., Alvi, P. A., & Subramaniam, U, 2021, John Wiley & Sons, ISBN: 978-0367564261.
5. E-waste recycling and management: present scenarios and environmental issues, Khan, Anish, andAbdullah M. Asiri. 2019, Springer, Vol. 33. ISBN: 978-3-030-14186-8.
6. Energy storage and conversion devices: Supercapacitors, batteries and hydroelectric cells, AnuragGaur, A. L. Sharma, Anil Arya. 2021, CRC press, 1st edition, ISBN: 978-1-003-14176-1.
7. Fundamentals of analytical chemistry: An introduction, Douglas A. Skoogetetal., 2004 Thomson Asiapte Ltd., 8th, ISBN: 978-0-495-55828-6.

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

Skill Sets

All topics in theory components are relevant to skill development through Participative Learning techniques. This is attained through assessment components mentioned in course plan.

Course Code: CHE2502	**Course Title: Chemistry of Smart Materials Lab** **Type of Course: Laboratory (BSC)**	**L-T-P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	Before undertaking this Chemistry of Smart Materials Lab course, students are expected to possess foundational knowledge of chemistry, including an understanding of acids and bases, metals and metal ions, oxidizing and reducing agents, various types of instrumental analysis, and the proper use of laboratory glassware. Additionally, students should be familiar with handling chemicals and glassware safely and adhering to essential laboratory safety precautions.					
Anti-requisites	NIL					
Course Description	The laboratory course aims to develop experimental skills and apply fundamental chemical principles to address chemistry-related problems in engineering. The experiments are carefully designed to complement the theoretical concepts covered in lectures, providing hands-on experience to deepen understanding and reinforce learning. This course is designed to cater to Environment and Sustainability.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Chemistry of Smart Materials Lab" and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques.					
Course Outcomes	On successful completion of the course, students shall be able to: **CO1:**Determine the amount of acids in domestic and industrial waste using conductometric and pH sensors. [Remember] **CO2:**Calculate the amount of metal ions in electronic waste using potentiometric and optical sensors. [Apply] **CO3:** Examine the viscoelastic property of the organic solvents using viscometrymethod. [Analyze] **CO4:**Apply the appropriate titration method to quantify the metal ions in industrial waste. [Apply] **CO5:**Explain the preparation, identification and recovery of industrially important materials using relevant techniques [Understand]					
Course Content: **Total 30 sessions**						
Experiment 1						
Determination of strength of strong acid in battery electrolyte using conductometric sensors.						
Experiment 2						
Determination of pKa of organic acid of battery electrolyte using pH sensor.						
Experiment 3						

Estimation of copper from PCBs by using colorimeter (Optical Sensor).
Experiment 4
Determination of viscosity coefficient of a given organic liquid using Ostwald's Viscometer (viscoelastic property).
Experiment 5
Estimation of iron in electronic devices using Std. Potassium permanganate solution.
Experiment 6
Conductometric estimation of mixture of acids in recycling process of E- waste.
Experiment 7
Estimation of iron from e-waste using Electrochemical sensors.
Experiment 8
Estimation of valuable metals (copper) from e- waste by Iodometric titration.
Experiment 9
Flame photometric estimation of sodium (battery recycling- Optical Sensor).
Experiment 10
Synthesis of conducting polyaniline for gas sensor applications (Demonstration experiment).
Experiment 11
Green synthesis of nanomaterials (Demonstration experiment).
Experiment 12
Recovery of valuable metals from e- waste by eletroless method (Demonstration experiment).
Any 10 experiments will be conducted out of 12
Continuous Internal Assessment: • Midterm exam • Experimental Evaluation • Viva-voce • Endterm exam
Text Book 1. Lab manual for Engineering chemistry by B. Ramadevi and P. Aparna, S. Chand Publications, New Delhi (2022) 2. Vogel's text book of practical organic chemistry 5th edition 3. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications. 4. College Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd. New Delhi
References 1. Engineering Chemistry Laboratory Manual (English, Paperback, Dr Manoj Kumar Solanki), Educreation

Publishing

E-resources:

1. https://books-library.net/files/download-pdf-ebooks.org-kupd-679.pdf

Video Links:

1. https://www.youtube.com/watch?v=gd1YQr-74sw
2. https://www.youtube.com/watch?v=wVJ8WQax0rQ
3. https://www.youtube.com/watch?v=aWwEGCNtKwk
4. https://www.youtube.com/watch?v=JhBs_8DrPYo
5. https://www.youtube.com/watch?v=5bFAx2b_6A8
6. https://www.youtube.com/watch?v=_IVVZnAFfrM
7. https://www.youtube.com/watch?v=BBhuXOh9vOM
8. https://www.youtube.com/watch?v=j-nW3Jhc794

Topics relevant to SKILL DEVELOPMENT: All the experiments are relevant to Skill Development through Experiential Learning Techniques.

This is attained through assessment component mentioned in course plan.

Course Code: MAT2303	**Course Title: Linear Algebra and Vector Calculus** **Type of Course: BSC**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	MAT2301 Calculus and Differential Equations					
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 Outcomess	On successful completion of the course the students shall be able to: CO1: Implement matrix operations and elementary row transformations to solve systems of linear equations using Gaussian elimination, Cramer's rule, and LU-					

	decomposition. CO2: Apply concepts of vector spaces, linear independence, basis, and dimension to represent and analyze vectors in multi-dimensional spaces. CO3: Employ linear transformations and their matrix representations to compute null space, range, and perform change of basis. CO4: Utilize eigenvalues and eigenvectors to diagonalize matrices and solve related problems. CO5: Execute vector calculus operations such as gradient, divergence, and curl to analyze scalar and vector fields. CO6: Carry out line and surface integrals using Green's, Gauss's, and Stokes' theorems.	
Course Content:		
Module 1	**Systems of Linear Equations**	**6 Sessions**
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**	**9 Sessions**
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 Sessions**
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**	**15 Sessions**
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.

Course Code: MEC1006	**Course Title: Engineering Graphics Type of Course: School Core & Theory Only**	**L- T-P- C**	**2**	**0**	**0**	**2**
VersionNo.	1.2					
CoursePre-requisites	**NIL**					
Anti-requisites	**NIL**					
CourseDescription	Thecourseisdesignedwiththeobjectiveofgivinganoverviewofengineering graphics.Itisintroductoryinnatureandacquaintsthestudentswiththetechniquesusedtocreateengineeringdrawings. The course emphasizes on projection ofpoints,lines,planesandsolids and isometricprojections.					

CourseObjective	The objective of the course is to familiarize the learners with the concepts of "Engineering Graphics" and attain SKILL DEVELOPMENT through Problem solving methodologies.	
CourseOutcomes	Onsuccessfulcompletionofthiscoursethestudentsshallbeableto: 1. Demonstratecompetencyof Engineering Graphics asperBISconventionsandstandards. 2. ComprehendthetheoryofprojectionfordrawingprojectionsofPoints,LinesandPlanesunder differentconditions. 3. PreparemultivieworthographicprojectionsofSolidsbyvisualizingthemindifferent positions. 4. Preparepictorialdrawingsusingtheprinciplesofisometricprojectionstovisualizeobjectsinthreedimensions.	
CourseContent:		
Module1	**Introduction toDrawing**	**02Sessions**
Topics: Introduction,drawinginstrumentsandtheiruses,relevantBISconventionsandstandards,Lettering,Lineconventions,dimensioning,Selectionofdrawing sheetsizeandscale.		
Module2	**Orthographic projectionsof Points,Straight LinesandPlane Surfaces**	**10Sessions**
Topics: Introduction, Definitions – Elements of projection and methods of projection, Planes of projection,referencelineandconventionsadopted.FirstangleandthirdangleprojectionsProjectionofPointsinall4 quadrants. Projections of Straight Lines (located in first quadrant/first angle projection only): True and apparentlengths, true and apparent Inclinations to reference planes. (No application problems). Projection ofPlane 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 methodonly.		
Module3	**OrthographicProjectionsof Solids**	**10Sessions**
Topics: Introduction,Projectionofrightregularprisms,pyramids,cone,hexahedronandtetrahedronindifferentpositions(ProblemsrestingonHPonly andFirstangleprojection).		
Module4	**IsometricProjections ofSolids (Usingisometricscale only)**	**8Sessions**
Topics: Introduction,Isometricscale,Isometricprojectionsofrightregularprisms,cylinders,pyramids, cones and theirfrustums, spheres and hemispheres, hexahedron (cube), andcombination of 2 solids, conversion of orthographic view to isometric projection of simpleobjects.		

TextBook: 1.N.D.Bhatt,"EngineeringDrawing:PlaneandSolidGeometry,"CharotarPublishingHousePvt.Ltd.
References: **1.** K.R.Gopalakrishna,"EngineeringGraphics",SubhashPublishers,Bangalore. **2.** D.M.Kulkarni,A.P.Rastogi,A.K.Sarkar, "EngineeringGraphicswithAutoCAD,"PrenticeHall. **3.** D.A.Jolhe,"EngineeringDrawingwithIntroductiontoAutoCAD,"TataMcGrawHill. **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.

Course Code: CSE1500	**Course Title: Computational Thinking using Python** **Type of Course: Lab Integrated Course (ESC)**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	Nil					
Anti-requisites	NIL					
Course 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.					
Course Objective	To develop computational thinking and problem-solving skills using Python by enabling students to design algorithms, implement programs using core programming constructs and build efficient solutions using data structures and file handling techniques for real-world applications.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Apply computational thinking principles to design algorithms for solving basic and intermediate computing problems. [Understand] **CO2:** Implement Python programs using data types, operators, and expressions to perform computational tasks. [Apply] **CO3:** Develop programs using control structures, functions, and recursion to solve structured problems. [Apply] **CO4:** Utilize data structures such as lists, tuples, and dictionaries for efficient data					

	representation and manipulation. [Apply] **CO5:** Design and implement Python programs for file handling and exception management to handle real-world data. [Apply] **CO6:** Build modular and reusable programs using functions and standard Python libraries to enhance code efficiency and readability. [Apply]

Course Content:

Module 1	**Computational Thinking And Problem Solving**	**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

Module 2	**Datatypes, Expressions, Statements**	**6 Sessions**

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.

Module 3	**Control flow, Functions, Strings**	**6 Sessions**

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.

Module 4	**Lists, Tuples, Dictionaries**	**6 Sessions**

Topics:
Lists: 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.

Module 5	**Files**	**6 Sessions**

Files 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).

List of Practical Tasks **30 Sessions**

WEEK 1: Computational Thinking & Algorithms

Lab Sheet 1: Problem Solving & Algorithm Design

Level 1: Write an algorithm and Python program to find the minimum element in a list of numbers.

Level 2: Write an algorithm and Python program to insert an element into a sorted list at the correct position.

Lab Sheet 2: Algorithm Strategies (Iteration & Recursion)

Level 1: Write a Python program to guess a number within a given range using iteration.

Level 2: Write a Python program to solve the Tower of Hanoi problem using recursion.

WEEK 2: Data Types & Expressions

Lab Sheet 3: Data Types & Variables

Level 1: Write a Python program to demonstrate different data types (int, float, boolean, string, list).

Level 2: Write a Python program to perform type conversions and display results.

Lab Sheet 4: Expressions & Operators

Level 1: Write a Python program to swap two variables using tuple assignment.

Level 2: Write a Python program to circulate the values of n variables.

WEEK 3: Expressions & Problem Solving

Lab Sheet 5: Mathematical Expressions

Level 1: Write a Python program to calculate the distance between two points.

Level 2: Write a Python program to evaluate complex arithmetic expressions with operator precedence.

Lab Sheet 6: Debugging & Interactive Mode

Level 1: Write a Python program with intentional errors and debug it.

Level 2: Write a Python program to demonstrate interactive mode operations.

WEEK 4: Conditional Statements

Lab Sheet 7: Decision Making

Level 1: Write a Python program to check whether a number is positive, negative, or zero.

Level 2: Write a Python program to find the largest of three numbers using if-elif-else.

Lab Sheet 8: Advanced Conditions

Level 1: Write a Python program to check whether a number is even or odd.

Level 2: Write a Python program to implement a grading system using chained conditions.

WEEK 5: Iteration (Loops)

Lab Sheet 9: Looping Basics

Level 1: Write a Python program to print numbers from 1 to N using a loop.

Level 2: Write a Python program to compute factorial using loops.

Lab Sheet 10: Loop Control

Level 1: Write a Python program demonstrating break and continue.

Level 2: Write a Python program to generate number patterns using nested loops.

WEEK 6: Functions

Lab Sheet 11: Functions Basics

Level 1: Write a Python function to calculate square root of a number.

Level 2: Write a Python function to find GCD of two numbers.

Lab Sheet 12: Recursion & Scope

Level 1: Write a recursive function to calculate exponentiation.

Level 2: Write a Python program demonstrating local and global variables.

WEEK 7: Strings

Lab Sheet 13: String Operations

Level 1: Write a Python program to demonstrate string slicing.

Level 2: Write a Python program to check whether a string is a palindrome.

Lab Sheet 14: String Methods

Level 1: Write a Python program using string functions (upper, lower, replace).

Level 2: Write a Python program to count vowels and consonants in a string.

WEEK 8: Lists

Lab Sheet 15: List Basics

Level 1: Write a Python program to create and display a list.

Level 2: Write a Python program to find the sum of elements in a list.

Lab Sheet 16: List Operations

Level 1: Write a Python program to demonstrate list slicing.

Level 2: Write a Python program to perform linear search on a list.

WEEK 9: Advanced Lists

Lab Sheet 17: List Methods

Level 1: Write a Python program to demonstrate list methods (append, remove).

Level 2: Write a Python program to sort a list.

Lab Sheet 18: List Comprehension

Level 1: Write a Python program using list comprehension to generate squares.

Level 2: Write a Python program to create a histogram of list values.

WEEK 10: Tuples

Lab Sheet 19: Tuple Basics

Level 1: Write a Python program to create and access tuple elements.

Level 2: Write a Python program to demonstrate tuple assignment.

Lab Sheet 20: Tuples as Return Values

Level 1: Write a Python function returning multiple values using tuple.

Level 2: Write a Python program to swap values using tuples.

WEEK 11: Dictionaries

Lab Sheet 21: Dictionary Basics

Level 1: Write a Python program to create and access dictionary elements.

Level 2: Write a Python program to count frequency of elements using dictionary.

Lab Sheet 22: Dictionary Applications

Level 1: Write a Python program to store student marks using dictionary.

Level 2: Write a Python program to generate a retail bill using dictionary.

WEEK 12: Advanced List Processing

Lab Sheet 23: Sorting & Searching

Level 1: Write a Python program to implement simple sorting.

Level 2: Write a Python program to implement binary search.

Lab Sheet 24: Data Processing

Level 1: Write a Python program to generate student marks statement.

Level 2: Write a Python program to analyze list data (min, max, average).

WEEK 13: File Handling Basics

Lab Sheet 25: File Operations

Level 1: Write a Python program to create and write to a text file.

Level 2: Write a Python program to read and display file contents.

Lab Sheet 26: File Processing

Level 1: Write a Python program to count words in a file.

Level 2: Write a Python program to copy contents from one file to another.

WEEK 14: Exceptions & Validation

Lab Sheet 27: Exception Handling

Level 1: Write a Python program using try-except.

Level 2: Write a Python program to handle multiple exceptions.

Lab Sheet 28: Validation Programs

Level 1: Write a Python program to validate voter's age.

Level 2: Write a Python program to validate marks range (0–100).

WEEK 15: Modules & Advanced Concepts

Lab Sheet 29: Modules & Packages

Level 1: Write a Python program to create and use a module.

Level 2: Write a Python program to use built-in modules.

Lab Sheet 30: Command Line Arguments

Level 1: Write a Python program to accept command-line arguments.

Level 2: Write a Python program to perform operations using command-line inputs

Project 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

1. **Topics relevant to development of "Employability":** Data structures using python.
2. **Topics relevant to "PROFESSIONAL ETHICS":** Naming and coding convention for simple programs using python.

Course Code: ECE2022	**Course Title: DIGITAL DESIGN** **Type of Course: School Core [Theory Only]**	**L-T-P-C**	**2**	**0**	**0**	**2**
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	The purpose of this course is to enable the students to appreciate the fundamentals of digital logic circuits and Boolean algebra. Further to focusing on combinational and sequential digital logic circuits. This course is analytical in nature and needs a fundamental knowledge on logical computation with Boolean algebra. The focus of the course will be to discuss the minimization techniques for making low-cost canonical digital circuit implementations. In this course, we emphasize design procedures of digital electronic circuits. Additionally, this course will create a foundation for future courses that includes Computer Architecture, Microprocessors, Microcontrollers and Embedded Systems etc.					
Course Objectives	The objective of the course is to familiarize the learners with Design and Analysis of Digital Logic Circuits concepts and to improve the learner's **SKILL DEVELOPMENT** by using **PARTICIPATIVE LEARNING TECHNIQUES** through Assessments**.**					
Course Outcomess	On successful completion of this course the students shall be able to: **CO1. Explain** the fundamental concepts of Number system, Boolean algebra and Logic gates **CO2. Determine** the minimum literal Boolean Expression using K-Map minimization techniques **CO3. Make use of**relevant logic gates for constructing the combinational circuit. **CO4. Produce** standard and real world combinational logic circuits using minimum number of logic gates **CO5. Demonstrate** the working principles of flip-flops and latches within the					

	context of sequential logic circuits. **CO6. Design** synchronous and asynchronous logic circuit using Flip-flops	
Module 1	**Number System, Logic Gates & Boolean function simplification**	**10 Sessions**
Review of Number systems -Overview of Boolean Algebra- Boolean Laws, Rules & Theorems. Boolean function Standard form and Canonical form. Review of logic gates -Basic gates, Universal gates & Special Gates- Boolean functions simplifications using Algebraic functions & Karnaugh-map. SOP and POS simplifications & Implementations.		
Module 2	**Combinational Logic circuits**	**10 Sessions**
Digital logic circuit Introduction –Classification of circuit- Analysis and Design procedure of Combinational Logic circuits - Binary Adder and Subtractor- Magnitude comparator- , Multiplexers - DeMultiplexers, Decoders, Encoders and Priority Encoders-Real world scenario based application design using logic gates.		
Module 3	**Sequential logic circuits**	**10 Sessions**
Introduction to sequential circuits- Storage elements: latches and flip flops- Characteristic tables and equations, excitation table- Analysis of clocked sequential circuits- state diagram-state table and design of finite state machines - Registers & Counters.		
•		
Targeted Application & Tools that can be used: **NA**		
Project Based Learning proposed for this course **NA**		

Text Books:

1. **T1:** Thomas L Flyod, *"Digital Fundamentals"*, Eleventh Edition, Pearson Education. ISBN-10: 132737965. (2014).eBook-[PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org
2. **T2.**Mano, M. Morris and Ciletti Michael D., *"Digital Design"*, Pearson Education, 6th edition ., [PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download (studymaterialz.in)

References:

R1 Jain, R. P., *"Modern Digital Electronics",* McGraw Hill Education (India).

R2 Roth, Charles H., Jr and Kinney Larry L., *"Fundamentals of logic Design"*, Cengage Learning.

E-Resources:

4. **eBook1**: Mano, M. Morris and Ciletti Michael D., *"Digital Design"*, Pearson Education.
5. {[PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download
}
6. **eBook2:**Floyd "DIGITAL LOGIC DESIGN" fourth edition- ePub, eBook- [PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org.
7. NPTEL Course- NPTEL :: Electrical Engineering - NOC:Digital Electronic Circuits

8. Digital Logic Design PPT Slide 1 (iare.ac.in)

Course Code: ECE2052	**Course Title: Digital Design Lab** **Type of Course: School Core [Lab Only]**	**L-T-P-C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This course gives knowledge through the simulation software for design and implementation of various digital logic circuit. The theoretical concepts and working principles of combinational and sequential logic circuits are verified through implementation using openCV. This course provides necessary learning for fundamental concepts of computer and communication engineering. The purpose of this course is to support the students to exhibit the design and analysis skills. The course enhances the Design and Implementation abilities through laboratory tasks. Additionally, it provides an opportunity to verify the theoretical knowledge.					
Course Objectives	The objective of the course is to familiarize the learners with the design concepts of digital logic circuit and attain Skill Development through Experiential Learning by simulation-based laboratory Assessments.					
Course Outcomess	On successful completion of the course the students shall be able to CO1. **Interpret** Boolean laws and the logic gates functionality using available circuit elements in simulator platforms. CO2. **Apply** the fundamental laws and rules of Boolean algebra for simplified circuit simulation using any simulator tool. CO3. **Develop** combinational logic circuits by connecting offered gates for basic arithmetic operation CO4. **Make use of** circuit elements for ensuring the digital logic circuit's truth table perfection. CO5. **Examine** the performance of sequential logic circuit simulation using flip-flop and latch elements CO6. **Demonstrate** the operation of clocked and edge-triggered sequential components in an open-source simulation environment					

LIST OF EXPERIMENTS **30 Sessions**

S. No	Name of the Experiment/Exercise
1	Logic Gates
2	Boolean Algebra
3	Adder

4	Subtractor
5	Magnitude Comparator
6	Design of Combinational Circuits
7	Multiplexer, Demultiplexer & Applications of MUX
8	Flip-Flops
9	Shift Registers
10	Counter Design

Targeted Application & Tools that can be used: Targeted Applications: This course is contributed for placement in core companies, research & development work and also useful to know the existing & developing communications.

Professionally Used Software: CIRCUIT VERSE, HDL,VHDL

Project Based Learning proposed for this course

NA

Text Books:

T1. Mano, M. Morris and Ciletti Michael D., "Digital Design", Pearson Education, 6th edition

T2. Thomas L. Floyd "DIGITAL LOGIC DESIGN" ,Pearson Education, fourth edition.

References:

R1. Jain, R. P., "Modern Digital Electronics", McGraw Hill Education (India), Fourth Edition

R2. Roth, Charles H., Jr and Kinney Larry L., "Fundamentals of logic Design", Cengage Learning, Seventh Edition

- Presidency Lab Manual- for ECE2052 DIGITAL DESIGN LAB

E-Resources:

1. eBook1: Mano, M. Morris and Ciletti Michael D., *"Digital Design"*, Pearson Education. Book Free Download (studymaterialz.in)

[PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download

2. eBook2:Floyd "DIGITAL LOGIC DESIGN" fourth edition- ePub, eBook- [PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org.
3. NPTEL Course- NPTEL :: Electrical Engineering - NOC:Digital Electronic Circuits
4. Digital Logic Design PPT Slide 1 (iare.ac.in)
5. 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

<table>
<tr><td>Course Code:
CIV1200</td><td>Course Title: Foundations of Integrated Engineering
Type of Course: ESC</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="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">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.</td></tr>
<tr><td>Course Objective</td><td colspan="6">The objective of the course is skill development of student by using Participative 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] 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.</td></tr>
<tr><td>Course Content:</td><td colspan="6"></td></tr>
<tr><td>Module 1</td><td colspan="4">Foundations of Engineering Practice</td><td colspan="2">6 Sessions</td></tr>
<tr><td colspan="7">Real-world problem-solving using data logic and practical applications, Collaboration and Innovation through multi-domain project, Engineering Ethics & Environmental Impact
Emerging Fields: Automation, and Introduction to bioinformatics and its application
Sustainability & Safety: Circular economy principles, carbon footprint analysis.</td></tr>
<tr><td>Module 2</td><td colspan="4">Civil Engineering & Geomatics</td><td colspan="2">6 Sessions</td></tr>
</table>

Smart Infrastructure & Geomatics: GIS mapping, LiDAR, drone surveys for urban planning, Geospatial data analysis for disaster management.
Sustainable Construction: 3D-printed structures, self-healing concrete, Digital twins for infrastructure monitoring.
Green Innovations: Net-zero energy buildings, rainwater harvesting systems.

Module 3	**Mechanical Engineering in Action**	**6 Sessions**

Advanced Manufacturing: Collaborative robots (cobots), additive manufacturing and 3D printing, Reverse engineering and prototyping.
Energy Systems: Solar/wind energy harvesting, piezoelectric applications.
Biomechanics: Prosthetics design, ergonomic product lifecycle.

Module 4	**Electrical & Electronics Engineering**	**6 Sessions**

Smart Devices & Systems: Embedded systems, Wearable technology, Edge computing and hardware platforms
Energy Innovations: EV charging infrastructure, wireless power transfer, Smart grid integration with renewables.

Module 5	**Fundamentals of IT**	**6 Sessions**

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

Targeted Application & Tools that can be used:

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

Text 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

References

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

Charles J. Brooks, Christopher Grow, & Philip Craig, “Cybersecurity Essentials", Wiley, 2nd Edition, 2021

Web-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

Course Code: EEE1200	**Course Title: Basics of Electrical and Electronics Engineering. Type of Course: Theory - ESC**	**L-T-P-C**	3	0	0	3
Version No.	2.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course 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.					
Course 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.					
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1: Apply** basic law**s** of Electrical Engineering to compute voltage, currents and other parameters in the circuits. **CO2: Discuss** various fundamental parameters appearing in the characteristics of semiconductor devices and their applications. **CO3: Summarize** the operations of different biasing configurations of BJTs and amplifiers. **CO4: Discuss** the performance characteristics and applications of various electrical Machines.					
Course Content:						
Module 1	**Introduction to Electrical Circuits**					**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.

Module 2	Semiconductor and Diode applications	10 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.

Module 3	Transistors and its Applications	10 Sessions

Transistor 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.

Module 4	Fundamentals of Electrical Machines	10 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.

Self-Learning Topics:

Clipping 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.

Targeted 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

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

Project 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

Text Book(s):

3. Kothari D. P. & Nagrath I. J., "Basic Electrical and Electronics Engineering", Tata McGraw-Hill
4. Education
5. 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.
6. A.P.Malvino, Electronic Principles,7thEdition, Tata McGraw Hill,2007
7. J. Millman, C. C. Halkias and C. D. Parikh, "Millman's Integrated Electronics", McGraw Hill Education, 2nd Edition.

8. Basics of Electrical & Electronics Laboratory Manual.

Reference 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.

Course Code EEE1250	**Course Title: Basics of Electrical and Electronics Engineering Lab** **Type of Course: Laboratory - ESC**	**L-T- P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course 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.					
Course 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.					
Basic 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.					
Course Outcomess	**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					
Course Content:						

List of Laboratory Tasks: **30 Sessions**

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.

Experiment 6: Study of PN-Junction Diode Characteristics in Forward and Reverse Bias Conditions.
Level 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.
Level 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.
Experiment No. 7: To observe the output waveform of half wave and full wave rectifier circuit and compute ripple factor and efficiency
Level 1: Identify the components required for a rectifier circuit, rig up the circuit, and sketch the output waveforms without filter.
Level 2: Rig up the rectifier circuit with RC filter, observe the output waveforms, determine the efficiency and ripple factor.
Experiment 8: To construct clipping and clamping circuits for different reference voltages and to verify the responses.
Level 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.
Level 2: Given a sinusoidal input of 10 V p-p, implement a positive / negative clipper with output clipped at 2 V.
Experiment 9: To calculate various parameters of emitter follower circuit using BJT
Level 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.
Level 2: Determine the values of Z_{in} input impedance and Z_{out} output impedance for Emitter Follower.
Experiment 10: To Implement RC Coupled amplifier using a BJT and sketch the frequency response.
Level 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.

Targeted 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

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

Course Material

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

Text Book:

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

Reference Books:

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 Learning Resources:

1. https://presidencyuniversity.linways.com
2. https://www.digimat.in/nptel/courses/video/108105112/L01 "Fundamentals of Electrical Engineering-Basic Concepts, Examples"
3. 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.

Course Code: ECE1511	**Course Title: Design Workshop** **Course Type : ESC**	**L- T-P- C**	**1**	**0**	**2**	**2**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	The objective of the course is **Employability Skills** of student by using **PARTICIPATIVE LEARNING** techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to** **CO1:** Explain the main features of the Arduino & the Raspberry Pi prototype board. **CO2:** Demonstrate the hardware interfacing of the peripherals to Arduino and Raspberry Pi system. **CO3:** Understand the types of sensors and its functions **CO4:** Demonstrate the functioning of live projects carried out using Arduino and Raspberry Pi system.					

Course Content:		
Module 1	**Basic concepts of Microcontrollers**	**3 Sessions**

Topics:

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.

Module 2	**Sensory Devices**	**3 Sessions**

Arduino Sensors: Humidity Sensor, Temperature Sensor, Water Detector / Sensor, PIR Sensor, Ultrasonic Sensor, Connecting Switches and actuators, sensor interface with Arduino.
Introduction to 3D Printer: 3D Printer technology and its working Principles, Applications. Introduction to online Simulators: Working with AutoCAD/Fusion 360 Simulator.

Module 3	**Introduction to Micro python**	**4 Sessions**

Topics:
Introduction to MicroPython, Comparison with other programming languages, Setting up the MicroPython development environment, Basics of MicroPython syntax and structure.

Module 4	**Working with Raspberry-pi**	**5 Sessions**

Introduction 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.

Lab: Name of the Experiments: **30 Sessions**

1. **Introduction Lab 1:**
   Level 1: Overview on Arduino based Micro-controller, and sensors.
   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**

Targeted Application & Tools that can be used:

Application Area:

Home 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 RaspberryPi combined with the wide range of sensors available, allow for endless possibilities in creating innovative projects.

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

Project 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 .

3. 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

1. Textbook(s):
2. Monk Simon "Programming Arduino: Getting Started with Sketches", Mc Graw Hill Publications Second Edition
3. Monk Simon "Raspberry Pi Cookbook: Software and Hardware Problems and Solutions", Publisher(s): O'Reilly Media, Inc. ISBN: 9781098130923 fourth Edition.
4. References
5. Reference Book(s)
6. 1. Neerparaj Rai "Arduino Projects for Engineers" BPB publishers,first edition, 2016.
7. 2. Ryan Turner "ArduinoProgramming "Nelly B.L. International Consulting Ltd. first edition,2019.
8. 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
9. 4. Stewart Watkiss "Learn Electronics with Raspberry Pi "Apress Berkeley, CA . second edition,2020. ISBN978-1-4842-6348-8
10. 5. Jo Prusa, "Basic of 3D printing", Prusa Research, 3rd edition.
11. 6. Volker Ziemann, "A Hands-On Course in Sensors Using the Arduino and Raspberry Pi (Series in Sensors)", CRC Press, 1st Edition. 2018.
12. Online Resources (e-books, notes, ppts, video lectures etc.):
13. Arduino trending Projects <https://www.https://projecthub.arduino.cc/>
14. Introduction to Arduino <https://onlinecourses.swayam2.ac.in/aic20_sp04/preview>
15. Case studies on Wearable technology<https://www.htciitm.org/wearables>
16. Raspberry-pi Projects <https://magpi.raspberrypi.com/articles/category/tutorials/>
17. Introduction to internet of things<https://nptel.ac.in/courses/106105166>
18. E-content:

19. Cattle Health Monitoring System Using Arduino and IOT (April 2021| IJIRT | Volume 7 Issue 11 | ISSN: 2349-6002)
20. 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.
21. 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.
22. 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.
23. Basil, Eliza Sawant, S.D. "IoT based traffic light control system using Raspberry Pi " DOI 10.1109/ICECDS.2017.8389604
24. Supriya S, 2Dr. Aravinda " Green leaf disease detection and identification using Raspberry Pi https://www.irjet.net/archives/V9/i8/IRJET-V9I847.
25. Dr. E.N. Ganesh., "Health Monitoring System using Raspberry Pi and IOT" DOI : http://dx.doi.org/10.13005/ojcst12.01.03

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

Course Code: CSE2264	**Course Title: Essentials of AI** **Type of Course: -Theory Course (ESC)**	**L-T-P-C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
CourseDescription	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.					
Course Objectives	To develop a foundational understanding of Artificial Intelligence by enabling students to explore search strategies and problem-solving techniques, study knowledge representation and reasoning and analyze real-world AI applications.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain various search techniques, problem-solving methods, and inference mechanisms in Artificial Intelligence. [Understand] **CO2:** Explain graphical and adversarial search algorithms used in AI. [Understand] **CO3:** Apply first-order logic and resolution techniques to solve reasoning problems. [Apply] **CO4:** Apply Hidden Markov Models (HMM) to solve sequence labeling problems. [Apply] **CO5:** Apply deep learning techniques to analyze and solve AI-based problems. [Apply]					

CourseContent:

Module 1	**Search Methods for Problem-Solving**	**12 Sessions**

Topics:
Introduction – 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.

Module 2	**Advanced Search Methods**	**12 Sessions**

Topics:
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

Module 3	**Knowledge-Based Logic Representation**	**12 Sessions**

Topics:
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

Module 4	**Uncertainty in AI**	**09 Sessions**

Topics:
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.

Project 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).

Textbook(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.

References:

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.

Weblinks

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.

Course Code: CSE2265	**Course Title: Essentials of AI Lab Type of Course: Laboratory Course (ESC)**	**L-T-P-C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
CourseDescription	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).					
Course Objectives	To develop practical skills in Artificial Intelligence by enabling students to implement basic AI algorithms, apply search and optimization techniques, develop simple intelligent systems, and analyze AI-based problem-solving approaches.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain various search, reasoning, and problem-solving methods in Artificial Intelligence. [Understand] **CO2:** Implement graphical and adversarial search algorithms to solve AI problems. [Apply] **CO3:** Apply first-order logic and resolution techniques for logical inference. [Apply **CO4:** Implement probabilistic models such as Hidden Markov Models (HMM) for sequence labeling tasks. [Apply] **CO5:** Analyze and evaluate the performance of AI algorithms in solving real-world problems. [Analyze]					
CourseContent:						

List of Laboratory Tasks:30 Sessions

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

Week 2, 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.

Week 3 & 4, Experiment No. 3: 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

Week 5, Experiment No. 4: 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.

Week 6 & 7, Experiment No. 5: Implementation of Adversarial Search
Level 1: Implement a Game Tree
Level 2: Perform Alpha-Beta Pruning and Ideal Ordering

Week 8 & 9, Experiment No. 6: Implementation of a CSP Solver
Level 1: Implement a CSP solver to solve a cryptarithmetic problem using backtracking
Level 2: Implement a CSP solver for map colouring

Week 10 & 11, Experiment No. 7: Using Python Packages for CSP
Level 1: Implement a CSP solver for Sudoku
Level 2: Implement a CSP solver for Addoku

Week 12, Experiment No. 8: Implement a Probabilistic Decision Maker
Level 1: Implement a Minesweeper solver
Level 2: Implement a Battleship solver

Week 13 & 14, Experiment No. 9: 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.

Week 15, Experiment No. 10: Perform Classification Using Scikit Learn
Level 1: Use a Naïve Bayes Classifier for a classification problem.
Level 2: Use a Logistic Regression Classifier for a classification problem.

Targeted Application & Tools that can be used:

1. Google Colab
2. 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.

Textbook(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.

References:

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.

Course Code: CSE2274	**Course Title: Competitive Programming and Problem Solving Type Of Course: Laboratory Course (ESC)**	**L-T-P-C**	**0**	**0**	**4**	**2**
VersionNo.	1.0					
CoursePre-requisites	NIL					
Anti-requisites	NIL					
Course Description	TheCompetitive Programming and Problem Solvingcourse equips students with efficient problem-solving skills for coding competitions and real-world challenges. Startingwithbrute-forcesolutions,studentslearntooptimizetimeandspacecomplexity using advanced techniques like dynamic programming, greedy algorithms, and backtracking.Hands-onpracticeonplatformslikeCodeChefandCodeforceshelpstackle 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.					
Course Objective	Theobjectiveofthecourseistofamiliarizethelearnerswiththeconcepts ofCompetitive Programming and Problem Solving and attain Skill Development through Experiential Learning techniques.					
Course Outcomes	Onsuccessfulcompletionofthecoursethestudentsshallbeableto: **CO1:**Understandthechallengesofonlinecompetitiveprogrammingplatforms,interpret problemconstraints andunderstandbrute-forcesolutionapproaches usedincompetitive programming. [Understand] **CO2:**Analyzethetimeandspacecomplexityofnumber-theoreticsolutionstodetermine the need for optimization under competitive programming constraints. [Analyze] **CO3:** Apply appropriate algorithmic paradigms and data structures to solve problems within defined time and memory constraints. [Apply] **CO4:**Analyze execution failures and identify algorithmic inefficiencies in solutions involvingadvanceddatastructureconceptstoredesignoptimizedsolutions withingiven time and memory constraints. [Analyze] **CO5:**Analyze advanced problem-solving techniques using dynamic programming to evaluate their effectiveness in large-scale competitive programming problems. [Analyze]					

Course Content:

Module 1	**IntroductiontoCompetitiveProgramming**	**NA**

Topics:

Overview of Efficient Codingfor ProblemSolvingand CP: Introductiontocompetitiveprogramming(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 asTLE; use of

STL.
CPProblemSolvingusingAdvancedTopics:conceptofdisjointsetsandtheirefficientrepresentation,algorithmic approachessuch asGreedy,Backtracking,Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

Module 2:	**NumberTheoryforProblem-Solving**	**NA**

Topics:
UseofNumberTheoryforproblem-solving:reducingtime/spacecomplexityofbruteforcecodingsolutionofSieveMethod,InverseModule,EuclidianMethodoffactorization;efficientcoding for Permutation Combination; XORing based and pattern-based solutions.

Module 3:	**OptimizingTime&SpaceUsingSequentialStorage**	**NA**

Topics:
CodingforOptimizingtimeandSpaceusingSequentialStorage:twopointerapproach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string matchingalgorithms;Kadane'salgo,stacks,priority-queuesandhashingbasedefficientcoding;median based problems and alternate solutions.

Module 4:	**Non-LinearDataStructures**	**NA**

Topics:
ApplyingNon-LinearDataStructuresforreal-lifeproblems:designofefficientsolutionsfor problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solvingusingtreesandbinarytrees,Catalannumbers,applicationsofgraphs,spanningtreeandpath algos for CP problems with reduced time/space complexity.

Module 5:	**ProblemSolvingusingAdvancedTopics**	**NA**

Topics:
CPProblemSolvingusingAdvancedTopics:conceptofdisjointsetsandtheirefficientrepresentation,algorithmic approachessuch asGreedy,Backtracking,Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

List of Practical Tasks **60 Sessions**

WEEK 1

Lab Sheet 1: Sorting & Ranking

Level 1: Write a program to find the runner who finished in the third position from the given finishing times of N runners.

Level 2: Write a program to find the top 10 runners along with their bib numbers efficiently for large datasets.

Lab Sheet 2: Searching Techniques

Level 1: Write a program to check whether a given book ID exists in a library database.

Level 2: Design and implement a system to search books efficiently by title using an appropriate data structure.

WEEK 2

Lab Sheet 3: Cost Calculation

Level 1: Write a program to calculate the total cost of products in a shopping cart.

Level 2: Design an algorithm to apply multiple discount rules and compute the final bill.

Lab Sheet 4: Modular Arithmetic

Level 1: Write a program to compute (a^m \mod p).

Level 2: Implement binary exponentiation for efficient computation of large powers modulo p.

WEEK 3

Lab Sheet 5: Combinatorics

Level 1: Write a program to calculate the number of ways to choose K cards from N cards.

Level 2: Design an algorithm to calculate probabilities of drawing specific card combinations.

Lab Sheet 6: Graph Connectivity

Level 1: Write a program to determine if a path exists between two nodes in a network.

Level 2: Design an algorithm to detect anomalies in a network using XOR-based comparison techniques.

WEEK 4

Lab Sheet 7: Array Processing

Level 1: Write a program to find the minimum time required for cars to pass a point based on their speeds.

Level 2: Implement a two-pointer approach to simulate movement of cars with different lengths.

Lab Sheet 8: String Matching

Level 1: Write a program to count occurrences of a substring in a given string.

Level 2: Implement the KMP algorithm to efficiently find all occurrences of a pattern in a text.

WEEK 5

Lab Sheet 9: Priority Queue

Level 1: Write a program to track the highest bid in an auction system.

Level 2: Design a system to handle concurrent bids efficiently and update the highest bid.

Lab Sheet 10: Shortest Path

Level 1: Write a program to check if two users are connected in a social network.

Level 2: Implement Dijkstra's algorithm to find the shortest path between users.

WEEK 6

Lab Sheet 11: Tree Traversal

Level 1: Write a program to traverse a file system and display all files and directories.

Level 2: Design a file system supporting operations like create, delete, and search.

Lab Sheet 12: Tree Applications

Level 1: Write a program to calculate the total price of items in a shopping cart represented as a tree.

Level 2: Design a system to apply discounts efficiently in a hierarchical shopping cart.

WEEK 7

Lab Sheet 13: Disjoint Set

Level 1: Write a program to check whether all users in a group are connected.

Level 2: Find the minimum number of connections required to connect all users.

Lab Sheet 14: Greedy Algorithms

Level 1: Write a program to find the cheapest path to reach a destination.

Level 2: Design an algorithm to find the fastest path considering time constraints.

WEEK 8

Lab Sheet 15: BFS Applications

Level 1: Write a program to find the minimum number of moves for a rook to reach a target square.

Level 2: Implement BFS to find shortest path in a grid.

Lab Sheet 16: Game Algorithms

Level 1: Represent a game state using a tree structure.

Level 2: Implement minimax algorithm with alpha-beta pruning.

WEEK 9

Lab Sheet 17: Sorting Algorithms

Level 1: Write a program to implement insertion sort.

Level 2: Write a program to implement merge sort or quick sort.

Lab Sheet 18: Searching Algorithms

Level 1: Write a program to perform linear search.

Level 2: Write a program to perform binary search efficiently.

WEEK 10

Lab Sheet 19: Algorithm Analysis

Level 1: Write a program to compare execution time of two algorithms.

Level 2: Optimize a given algorithm to improve performance.

Lab Sheet 20: Recursion

Level 1: Write a program to solve Tower of Hanoi.

Level 2: Compare recursive and iterative solutions for a problem.

WEEK 11

Lab Sheet 21: Graph Representation

Level 1: Represent a graph using adjacency list.

Level 2: Represent a graph using adjacency matrix.

Lab Sheet 22: Graph Traversal

Level 1: Implement Depth First Search (DFS).

Level 2: Implement Breadth First Search (BFS).

WEEK 12

Lab Sheet 23: Data Structure Integration

Level 1: Combine arrays and lists to solve a problem.

Level 2: Use multiple data structures (tree + hash table) in one solution.

Lab Sheet 24: Real-world Application

Level 1: Design a simple recommendation system.

Level 2: Optimize it for large-scale data.

WEEK 13

Lab Sheet 25: String Algorithms

Level 1: Write a program for basic string matching.

Level 2: Implement efficient pattern matching algorithm.

Lab Sheet 26: Network Algorithms

Level 1: Check connectivity in a network.

Level 2: Detect anomalies using data comparison techniques.

WEEK 14

Lab Sheet 27: Algorithm Design

Level 1: Solve a problem using brute-force approach.

Level 2: Improve the solution using greedy or dynamic approach.

Lab Sheet 28: Performance Evaluation

Level 1: Analyze runtime of a program.

Level 2: Compare different algorithmic approaches.

WEEK 15

Lab Sheet 29: Case Study

Level 1: Implement a real-world problem solution.

Level 2: Optimize and document the solution.

Lab Sheet 30: Mini Project

Level 1: Combine multiple algorithms to build a system.

Level 2: Present optimized solution with performance analysis.

TargetedApplication&Toolsthatcanbeused:

1. CorC++Compiler(g++):ThestandardcompilerforCP.Familiarizestudentswithcompilationflags (e.g., -O2 for optimization).
2. IDE(IntegratedDevelopment Environment):Code::Blocks, VisualStudio, CLion, orsimilar IDEs. These provide debugging capabilities, code completion, and other helpful features.
3. OnlineJudges(CodeChef,Codeforces,LeetCode,HackerRank):Essentialforpracticingandsubmitting 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. WolframAlpha:Ausefultoolforverifyingnumber theorycalculationsandexploringconcepts.
7. StringLibraries:Familiarizestudentswiththestringmanipulationfunctionsavailable inC++.
8. GraphVisualizationTools:ToolslikeGraphvizcanbehelpfulforvisualizinggraphsand understanding graph algorithms.
9. DPDebuggingTechniques: PracticedebuggingDP solutions, astheycanbecomplex. Visualizing the DP table can be helpful

TextBooks:

1 Guide to Competitive Programming: Learning and Improving Algorithms Through Contests" (3rd Edition), *Antti Laaksonen, springer, 2024*

2 "DataStructuresandAlgorithmsinJava:AProject-BasedApproach"–*DanS.Myers, Cambridge University Press*

ReferenceBooks:

1. DataStructuresandAlgorithmicThinkingwithPython/C++/Java",NarasimhaKarumanchi,5th Edition, Career Monk, 2017.
2. IntroductiontoAlgorithms,ThomasH.Cormen(Author), CharlesE.Leiserson(Author),RonaldL.Rivest,fourtheditionApril2022 .

Web Resources

1.https://nptel.ac.in/courses/106106231

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

Assessment Type

1. Midterm exam
2. Assignment (review of digital/ e-resource from PU link given in references section - mandatory to

submit screen shot accessing digital resource.) 3. Quiz 4. End Term Exam 5. Self-Learning

Course Code: CSE2200	**Course Title: Program Solving Using C** **Type of Course: Theory Course**	**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 ableto develop logics which will help them to create programs and applications in C. ACAlso bylearningthebasicprogrammingconstructs theycaneasilyswitchover toanyotherlanguageinfuture.					
Course Objective	TheobjectiveofthecourseistofamiliarizethelearnerswiththeconceptsofProblemSolvingUsingC andattainEmployabilitythroughProblemSolvingMethodologies.					
Course Outcomes	**Onsuccessfulcompletion ofthiscoursethestudentsshallbeableto:** 1. Describe algorithms with flowcharts for solving problems [Understand] 2. Demonstrate the application of C programming constructs in developing simple programs [Apply] 3. Implement applications using arrays and strings [Apply] 4. Apply functional programming principles to construct modular and reusable software components for solving complex problem [Apply] 5. Develop applications in C using structures and unions [Apply] 6. Differentiate between Sequential and Random Access File Processing through applications. [Analyze]					
Course Content:						
Module 1	**IntroductiontoCLanguage**					**6 Sessions**
Topics: Introduction to Programming – Algorithms – Pseudo Code -Flow Chart – Compilation – Execution – PreprocessorDirectives (#define, #include, #undef) - Overview of C – Constants, Variables and Data types – Operators andExpressions – Managing Input and Output Operations – Decision Making and Branching - Decision Making andLooping.						
Module 2	**IntroductiontoArrays andStrings**					**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 TwoDimensionalArrays.ExamplePrograms– Matrixoperations.**Strings:**Introduction-DeclaringandInitializingString Variables–ReadingStringsfrom Terminal–Writing StringtoScreen– StringHandlingFunctions.						
Module 3	**FunctionsandPointers**					**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 PointerVariables–InitializationofVariables –PointerOperators–PointerArithmetic–ArraysandPointers–Parameter Passing:PassbyValue,PassbyReference.						
Module 4	**StructuresandUnion**					**6 Sessions**

Topics: **Structures:**Introduction–DefiningaStructure–DeclaringStructureVariable–AccessingStructureMembers–ArrayofStructures –Arrayswithin Structures–**Union:**Introduction–DefiningandDeclaringUnion–DifferenceBetween Unionand Structure.		
Module 5	**Filehandling**	**6 Sessions**
Topics: Files:DefiningandOpeningaFile –ClosingaFile–Input/OutputOperationsonFile –RandomAccessFiles		
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/		

Course Code: CSE2201	**Course Title: Program Solving Using C Lab** **Type of Course: Laboratory Course**	**L- T-P-C**	**0**	**0**	**4**	**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 ableto develop logics which will help them to create programs and applications in C. ACAlso bylearningthebasicprogrammingconstructs theycaneasilyswitchover toanyotherlanguageinfuture.					
Course Objective	TheobjectiveofthecourseistofamiliarizethelearnerswiththeconceptsofProblemSolvingUsing C andattainEmployabilitythroughProblemSolvingMethodologies.					
Course Outcomes	**Onsuccessfulcompletion ofthiscoursethestudentsshallbeableto:** 1. Describe algorithms with flowcharts for solving problems [Understand] 2. Demonstrate the application of C programming constructs in developing simple programs [Apply] 3. Implement applications using arrays and strings [Apply] 4. Apply functional programming principles to construct modular and reusable software components for solving complex problem [Apply] 5. Develop applications in C using structures and unions [Apply] 6. Differentiate between Sequential and Random Access File Processing through applications. [Analyze]					

Course Content:	

List of Practicals: **60 Sessions**

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

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):

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/

Course Code: CSE2251	**Course Title: Data Communications and Computer Networks** **Type of Course: Theory Course (PCC)**	**L- T-P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	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.					
Course Objectives	To develop a strong foundation in data communication and computer networks by enabling students to understand networking concepts, analyze network architectures and protocols, explore transmission media and switching techniques, and evaluate network performance and emerging networking technologies.					
Course Outcomes	On successful completion of the course, the students shall be able to: **CO1:** Explain the fundamental concepts of data communication and computer networks. [Understand] **CO2:** Analyze the functions and protocols of the Data Link Layer. [Analyze] **CO3:** Apply IP addressing schemes and routing techniques in computer networks. [Apply] **CO4:** Explain the working principles of the Transport and Application layers. [Understand] **CO5:** Analyze network protocols and evaluate their performance for efficient data communication. [Analyze]					
Course Content:						
Module 1	**Introduction and Physical Layer**					**12 Sessions**
Topics: 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.						
Module 2	**Reference Models and Data Link Layer**					**12 Sessions**
Topics: 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.		
Module 3	**Network Layer**	**12 Sessions**
Topics: 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.		
Module 4	**Transport and Application Layer**	**9 Sessions**
Topics: 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.		
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		

Course Code: CSE2252	**Course Title: Data Communications and Computer Networks Lab Type of Course: Laboratory Course (PCC)**	**L- T-P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					

Course Description	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.
Course Objective	To develop practical skills in computer networking by enabling students to gain hands-on experience with networking tools and protocols, simulate and analyze data communication processes, implement socket programming for network applications, and evaluate network performance through experimental studies.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Design and configure basic network topologies using routers, switches, and end devices. [Apply] **CO2:** Analyze and troubleshoot network connectivity and performance issues using networking tools and simulators. [Analyze] **CO3:** Demonstrate the working of key networking protocols such as TCP/IP, ARP, ICMP, and DHCP. [Apply] **CO4:** Apply IP addressing and subnetting techniques for efficient network resource management. [Apply] **CO5:** Evaluate network performance and optimize configurations for improved reliability and efficiency. [Evaluate]

Course Content:

List of Laboratory Tasks: **30 Sessions**

WEEK 1: Introduction to Networks

Lab Sheet 1: Network Basics

Level 1: Identify different types of networks (LAN, WAN, MAN) and list their characteristics.

Level 2: Analyze a real-world network setup and classify its components and topology.

Lab Sheet 2: Network Topologies

Level 1: Draw and explain Bus, Star, Ring, and Mesh topologies.

Level 2: Compare topologies and suggest the best topology for a given scenario.

WEEK 2: Transmission Media & Models

Lab Sheet 3: Transmission Media

Level 1: Identify guided and unguided transmission media with examples.

Level 2: Compare different transmission media based on bandwidth, cost, and reliability.

Lab Sheet 4: OSI & TCP/IP Models

Level 1: Explain the layers of OSI and TCP/IP models.

Level 2: Map OSI layers with TCP/IP layers and justify their functions.

WEEK 3: Physical Layer Concepts

Lab Sheet 5: Signals & Transmission

Level 1: Differentiate between analog and digital signals.

Level 2: Analyze signal transmission in real-world communication systems.

Lab Sheet 6: Multiplexing

Level 1: Explain FDM, TDM, and WDM.

Level 2: Design a multiplexing scenario for multiple users.

WEEK 4: Spread Spectrum & Encoding

Lab Sheet 7: Spread Spectrum

Level 1: Explain FHSS and DSSS.

Level 2: Compare spread spectrum techniques in wireless communication.

Lab Sheet 8: Encoding Techniques

Level 1: Demonstrate digital encoding techniques.

Level 2: Analyze encoding schemes for efficient transmission.

WEEK 5: Error Detection & Correction

Lab Sheet 9: Error Detection

Level 1: Implement Parity and LRC.

Level 2: Implement CRC algorithm.

Lab Sheet 10: Error Correction

Level 1: Implement Hamming Code.

Level 2: Analyze error correction efficiency.

WEEK 6: Flow Control

Lab Sheet 11: Stop and Wait

Level 1: Simulate Stop-and-Wait protocol.

Level 2: Analyze efficiency and delay.

Lab Sheet 12: Sliding Window

Level 1: Implement Sliding Window protocol.

Level 2: Compare with Stop-and-Wait.

WEEK 7: Multiple Access Protocols

Lab Sheet 13: CSMA/CD

Level 1: Explain working of CSMA/CD.

Level 2: Simulate collision detection.

Lab Sheet 14: CSMA/CA

Level 1: Explain CSMA/CA.

Level 2: Compare CSMA/CD and CSMA/CA.

WEEK 8: IEEE Standards

Lab Sheet 15: Ethernet (802.3)

Level 1: Study Ethernet frame structure.

Level 2: Analyze Ethernet performance.

Lab Sheet 16: Wireless LAN (802.11)

Level 1: Study Wi-Fi architecture.

Level 2: Analyze wireless communication challenges.

WEEK 9: Network Layer Basics

Lab Sheet 17: IP Addressing

Level 1: Identify IPv4 address classes.

Level 2: Convert IPv4 to binary and vice versa.

Lab Sheet 18: IPv6

Level 1: Understand IPv6 format.

Level 2: Compare IPv4 and IPv6.

WEEK 10: Subnetting

Lab Sheet 19: Subnetting Basics

Level 1: Calculate subnet mask.

Level 2: Design subnet for given requirements.

Lab Sheet 20: Advanced Subnetting

Level 1: Perform CIDR calculations.

Level 2: Solve real-world subnetting problems.

WEEK 11: Routing

Lab Sheet 21: Distance Vector Routing

Level 1: Explain RIP algorithm.

Level 2: Simulate routing table updates.

Lab Sheet 22: Link State Routing

Level 1: Explain OSPF.

Level 2: Compare RIP and OSPF.

WEEK 12: Advanced Routing

Lab Sheet 23: Multicast Routing

Level 1: Explain multicast concepts.

Level 2: Analyze MOSPF/DVMRP.

Lab Sheet 24: Modern Networking

Level 1: Study VXLAN/EVPN.

Level 2: Analyze virtual networking concepts.

WEEK 13: Transport Layer

Lab Sheet 25: TCP & UDP

Level 1: Compare TCP and UDP.

Level 2: Analyze use cases.

Lab Sheet 26: Congestion Control

Level 1: Explain congestion control methods.

Level 2: Analyze RED/DECbit techniques.

WEEK 14: Application Layer

Lab Sheet 27: DNS & HTTP

Level 1: Explain DNS working.

Level 2: Analyze HTTP request-response.

Lab Sheet 28: Email Protocols

Level 1: Explain SMTP, POP3, IMAP.

Level 2: Compare email protocols.

WEEK 15: Network Applications

Lab Sheet 29: SNMP & Web Services

Level 1: Explain SNMP architecture.

Level 2: Analyze web services.

Lab Sheet 30: Virtual Networking

Level 1: Study virtual networking basics.

Level 2: Analyze real-world virtual network implementation.

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.
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

Course Code: **CSE2253**	**Course Title**: **Data Structures** **Type of Course**: **Theory Course (PCC)**	**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 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.					
Course Objective	To develop a strong foundation in data structures by enabling students to understand fundamental data structures and their operations, analyze algorithm efficiency, implement linear and non-linear data structures and apply appropriate techniques to					

	solve real-world problems while evaluating trade-offs between different data structures.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the concepts of basic data structures such as arrays, stacks, and queues and their operations. [Understand] **CO2:** Apply linked list data structures to solve real-world problems. [Apply] **CO3:** Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios. [Apply] **CO4:** Implement and demonstrate various searching and sorting algorithms. [Apply] **CO5:** Analyze the efficiency of data structures and algorithms in terms of time and space complexity. [Analyze]

Course Content:

Module 1	**Introduction to Data Structure and Linear Data Structure – Stacks and Queues**	**9 Sessions**

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	**Linear Data Structure- Linked List**	**12 Sessions**

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	**Non-linear Data Structures - Trees**	**12 Sessions**

Topics: 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 traversal, Binary Serach Tree, AVL Trees - Red Black Tree, Expression Tree , Heaps.

Module 4	**Non-linear Data Structures - Graphs and Hashing**	**6 Sessions**

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**	**6 Sessions**

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

Targeted Application & Tools that can be used

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

Project work/Assignment:

Assignment: 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.

References

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

R2Programming 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":

Linked list and stacks

Topics relevant to development of "Environment and sustainability: Queues

Course Code: CSE2254	**Course Title: Data Structures Lab** **Type of Course:Laboratory Course (PCC)**	**L-T- P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course 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.					
Course Objective	To develop programming and problem-solving skills by enabling students to implement various data structures, analyze their performance through experimentation and apply structured approaches to solve computational problems effectively.					

Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the concepts of fundamental data structures such as arrays, stacks, and queues and their operations. [Understand] **CO2:** Implement linear data structures such as linked lists to solve real-world problems. [Apply] **CO3:** Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios. [Apply] **CO4:** Implement and demonstrate searching and sorting algorithms. [Apply] **CO5:** Analyze the efficiency of different data structures and algorithms in terms of time and space complexity. [Analyze]
Course Content:	**30 Sessions**

List of Laboratory Tasks:30 Sessions

WEEK 1: Introduction & Arrays

Lab Sheet 1: Arrays Basics

Level 1: Write a program to create an array and perform insertion and deletion operations.

Level 2: Write a program to find largest, smallest, and average elements in an array.

Lab Sheet 2: Array Applications

Level 1: Write a program to reverse an array.

Level 2: Write a program to merge two arrays and remove duplicates.

WEEK 2: Stack

Lab Sheet 3: Stack Implementation

Level 1: Implement stack using array (push and pop operations).

Level 2: Write a program to check overflow and underflow conditions.

Lab Sheet 4: Stack Applications

Level 1: Write a program to convert infix expression to postfix.

Level 2: Write a program to evaluate postfix expression.

WEEK 3: Queue

Lab Sheet 5: Queue Implementation

Level 1: Implement queue using array (enqueue and dequeue).

Level 2: Implement circular queue.

Lab Sheet 6: Queue Applications

Level 1: Write a program to simulate a queue in real-life scenario (ticket booking).

Level 2: Implement priority queue.

WEEK 4: Linked List Basics

Lab Sheet 7: Singly Linked List

Level 1: Create a singly linked list and perform insertion.

Level 2: Perform deletion and display operations.

Lab Sheet 8: Linked List Applications

Level 1: Count number of nodes in a linked list.

Level 2: Reverse a linked list.

WEEK 5: Advanced Linked List

Lab Sheet 9: Circular Linked List

Level 1: Create and traverse a circular linked list.

Level 2: Perform insertion and deletion in circular linked list.

Lab Sheet 10: Linked List Problems

Level 1: Merge two linked lists.

Level 2: Detect loop in linked list.

WEEK 6: Recursion

Lab Sheet 11: Recursion Basics

Level 1: Write a recursive program for factorial.

Level 2: Write a recursive program for Fibonacci series.

Lab Sheet 12: Recursion Applications

Level 1: Solve Tower of Hanoi using recursion.

Level 2: Compare recursive and iterative approaches.

WEEK 7: Trees Basics

Lab Sheet 13: Binary Tree

Level 1: Create a binary tree and perform traversal.

Level 2: Implement preorder, inorder, and postorder traversal.

Lab Sheet 14: Tree Properties

Level 1: Count nodes and height of tree.

Level 2: Check if tree is balanced.

WEEK 8: Binary Search Tree

Lab Sheet 15: BST Implementation

Level 1: Insert elements into BST.

Level 2: Delete elements from BST.

Lab Sheet 16: BST Applications

Level 1: Search an element in BST.

Level 2: Find minimum and maximum elements.

WEEK 9: Advanced Trees

Lab Sheet 17: AVL Trees

Level 1: Implement AVL tree insertion.

Level 2: Perform rotations (LL, RR, LR, RL).

Lab Sheet 18: Heaps

Level 1: Implement max heap.

Level 2: Perform heap sort.

WEEK 10: Expression Trees & Red-Black Trees

Lab Sheet 19: Expression Tree

Level 1: Construct expression tree from postfix.

Level 2: Evaluate expression tree.

Lab Sheet 20: Red-Black Tree

Level 1: Understand insertion rules.

Level 2: Implement basic operations.

WEEK 11: Graph Basics

Lab Sheet 21: Graph Representation

Level 1: Represent graph using adjacency matrix.

Level 2: Represent graph using adjacency list.

Lab Sheet 22: Graph Traversal

Level 1: Implement BFS.

Level 2: Implement DFS.

WEEK 12: Graph Algorithms

Lab Sheet 23: Shortest Path

Level 1: Implement Dijkstra's algorithm.

Level 2: Analyze shortest path in weighted graph.

Lab Sheet 24: Minimum Spanning Tree

Level 1: Implement Prim's algorithm.

Level 2: Implement Kruskal's algorithm.

WEEK 13: Hashing

Lab Sheet 25: Hash Tables

Level 1: Implement hashing using division method.

Level 2: Handle collisions using chaining.

Lab Sheet 26: Advanced Hashing

Level 1: Implement linear probing.

Level 2: Compare static vs dynamic hashing.

WEEK 14: Searching

Lab Sheet 27: Searching Techniques

Level 1: Implement sequential search.

Level 2: Implement binary search.

Lab Sheet 28: Searching Applications

Level 1: Compare searching techniques.

Level 2: Analyze time complexity.

WEEK 15: Sorting

Lab Sheet 29: Basic Sorting

Level 1: Implement bubble sort and selection sort.

Level 2: Implement insertion sort.

Lab Sheet 30: Advanced Sorting

Level 1: Implement quick sort.

Level 2: Implement merge sort and compare performance
Targeted Application & Tools that can be used Use of PowerPoint software for lecture slides and use of Modern IDE like VS Code and Eclipse for lab programs to execute.
Project work/Assignment:
Assignment: 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.
References **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": Linked list and stacks **Topics relevant to development of "Environment and sustainability: Queues**

Course Code: **CSE2255**	**Course Title: Object Oriented Programming Using Java** **Type of Course: Theory Course (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 Outcomes	On successful completion of the course, students shall be able to: **CO1:** Describe fundamental programming concepts. [Understand] **CO2:** Apply the concepts of classes, objects, and methods to solve problems. [Apply] **CO3:** Apply the concepts of arrays and strings. [Apply] **CO4:** Implement inheritance and polymorphism to develop secure applications. [Apply] **CO5:** Apply the concepts of interfaces and error-handling mechanisms. [Apply]	
Course Content:		
Module 1	**Basic Concepts of Programming and Java**	**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**	**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**	**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**	**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**	**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 1. Herbert Schildt, "The Complete Reference Java 2", Tata McGraw Hill Education, 11th Edition,2019.		
References		

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

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

E book link 2: 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.

This is attained through assessment component mentioned in course handout.

Course Code: CSE2256	**Course Title: Object Oriented Programming Using Java Lab** **Type of Course: Laboratory Course (PCC)**	**L-T- P- C**	**0**	**0**	**2**	**1**
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 Outcomes	On successful completion of the course, students shall be able to: **CO1:** Demonstrate fundamental programming concepts. [Apply] **CO2:** Apply the concepts of classes, objects, and methods to solve problems. [Apply] **CO3:** Apply the concepts of arrays and strings. [Apply] **CO4:** Implement inheritance and polymorphism to develop secure applications. [Apply] **CO5:** Apply the concepts of interfaces and error-handling mechanisms. [Apply]					
Course Content: 30 Sessions						

List of Laboratory Tasks:

Week 1:

Lab Sheet 1: Java Program Structure & Basic I/O

Level 1:

Write a Java program to display "Hello, Java".

Demonstrate use of variables and data types.

Level 2:

Write a Java program to take user input (name, age) and display formatted output.

Week 2:

Lab Sheet 2: Operators and Expressions

Level 1:

Write a Java program to perform arithmetic operations (+, -, *, /, %).

Level 2:

Develop a program to evaluate a complex expression using relational and logical operators.

Week 3:

Lab Sheet 3: Control Statements – Branching

Level 1:

Write a Java program to check whether a number is even or odd using if-else.

Level 2:

Implement a menu-driven calculator using switch-case.

Week 4:

Lab Sheet 4: Control Statements – Looping

Level 1:

Write programs using loops:

1. Print numbers from 1 to N
2. Find factorial of a number

Level 2:

Generate Fibonacci series and check whether a number is prime.

Week 5:

Lab Sheet 5: Classes and Objects

Level 1:

Create a class Student with data members and display details using object.

Level 2:

Create multiple objects and demonstrate accessing members using reference variables.

Week 6:

Lab Sheet 6: Methods and Access Specifiers

Level 1:

Write a class with methods to perform addition and subtraction.

Level 2:

Demonstrate use of public, private, and protected access specifiers.

Week 7:

Lab Sheet 7: Constructors and Overloading

Level 1:

Implement a class with a default constructor and parameterized constructor.

Level 2:

Demonstrate constructor overloading and use of this keyword.

Week 8:

Lab Sheet 8: Static Keyword & Nested Classes
Level 1:
Write a program demonstrating static variables and static methods.
Level 2:
Create a nested class and access its members from the outer class.

Week 9:

Lab Sheet 9: Arrays (1D and 2D)
Level 1:
Write programs for:
1. Sum of array elements
2. Find largest element

Level 2:
Perform matrix addition and multiplication using 2D arrays.

Week 10:

Lab Sheet 10: Array of Objects
Level 1:
Create an array of objects (e.g., Employee class).
Level 2:
Perform operations like sorting objects based on a field.

Week 11:

Lab Sheet 11: Strings and StringBuffer
Level 1:
Perform basic string operations: Length, concatenation, comparison
Level 2:
Demonstrate mutable operations using StringBuffer and StringBuilder.

Week 12:

Lab Sheet 12: Inheritance
Level 1:
Create a base class and derived class using single inheritance.
Level 2:
Implement multilevel inheritance and demonstrate use of super keyword.

Week 13:

Lab Sheet 13: Polymorphism & Abstract Classes
Level 1:
Demonstrate method overriding (runtime polymorphism).
Level 2:
Create an abstract class and implement its methods in a subclass.

Week 14:

Lab Sheet 14: Exception Handling
Level 1:

Handle exceptions using try-catch-finally.

Level 2:

Create custom exceptions and demonstrate multiple catch blocks.

Week 15:

Lab Sheet 15: File Handling and Serialization

Level 1:

Write a Java program to Create a file & Write and read data using File I/O

Level 2:

Implement object serialization and deserialization.

Use File streams and buffered streams for efficient I/O.

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

Text Book

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

References

1. Cay S Horstmann and Cary Gornell, "CORE JAVA volume I-Fundamentals", Tenth Edition, Pearson 2015.
2. James W. Cooper, "Java TM Design Patterns – A Tutorial", Addison-Wesley Publishers.4th Edition, 2000.
3. 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.

This is attained through assessment component mentioned in course handout.

Course Code: CSE2257	**Course Title: Computer Organization and Architecture** **Type of Course: Theory Course (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 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.
Course Objective	To develop a comprehensive understanding of computer system organization by enabling students to analyze processor architecture and instruction execution, explore memory hierarchy and cache mechanisms, examine input/output organization and interfacing and evaluate system performance and optimization techniques.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Describe the fundamental components of a computer system and their interconnections. [Remember] **CO2:** Explain the concepts of Instruction Set Architecture (ISA) and memory organization. [Understand] **CO3:** Apply techniques to perform arithmetic and logical operations in computer systems. [Apply] **CO4:** Explain the organization and functioning of processor and memory subsystems. [Understand] **CO5:** Analyze the design and performance aspects of computer architecture components for efficient system operation. [Analyze]

Course Content:

Module 1	**Basic Structure of Computer**	**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.

Module 2	**Instruction Set Architecture and Memory Unit**	**12 Sessions**

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

Module 3	**Arithmetic And Input/output Design**	**10 Sessions**

Topics:
Arithmetic: Carry lookahead Adder, Signed-Operand Multiplication, Integer Division, and Floating point operations.

Input/output Design: Accessing I/O Devices, I/O communication, Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits

Module 4	BPU and Pipelining	11 Sessions

Topics:
Basic Processing Unit: Fundamental Concepts, Single Bus organization, Control sequence, Execution of a Complete Instruction, Multiple Bus Organization.

Pipelining: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, Hazards.

Targeted 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

Project work/Assignment:

Each batch of students (self-selected batch mates – up to 4 in a batch) will be allocated case studies/assignments

Textbook(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.

References

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 Development through Participative Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: **CSE2258**	**Course Title: Web Technologies** **Type of Course : Theory Course (PCC)**	**L-T-P- C**	**3 – 0 – 0 - 3**
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course highlights the basic web design using Hypertext Markup Language and Cascading Style Sheets. Students will be trained in planning and designing effective web pages by writing code using current leading trends in the web domain, enhancing web pages with the use of page layout techniques, text formatting, graphics, images, and multimedia. The focus is on popular key technologies that will help students to build Internet- and web-based applications that interact with other applications and with databases.		
Course Objective	To develop a strong foundation in web technologies by enabling students to understand internet protocols, design and develop static and dynamic web applications, explore client-side scripting and frameworks, implement server-side programming with database integration, and evaluate modern web application architectures.		
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the core concepts of web technologies including the Internet, web servers, and browsers. [Understand] **CO2:** Develop well-structured web pages using XHTML with text formatting, images, links, lists, tables, forms, and frames following web standards. [Apply] **CO3:** Design responsive and structured web page layouts using advanced CSS techniques such as positioning, floating, and multicolumn design. [Apply] **CO4:** Apply JavaScript concepts to create dynamic and interactive web pages. [Apply] **CO5:** Implement dynamic server-side web applications using PHP, including control structures, arrays, superglobals, and file handling. [Apply] **CO6:** Develop modular web applications by integrating PHP with MySQL databases and applying object-oriented programming principles. [Create]		
Course Content:			
Module 1	**Introduction to XHTML**		**10 Sessions**
Topics:			

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.

Module 2	**Advanced CSS**	**10 Sessions**

Topics:

CSS: Introduction to CSS, Defining & Applying a style, Creating style sheets, types of style sheet, selectors, CSS font properties, border properties, Box model, opacity, CSS pseudo class and pseudo-elements

Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Responsive Design, CSS Frameworks **XML:** Basics, demonstration of applications using XML

Module 3	**Fundamentals of JavaScript**	**10 Sessions**

Topics:

JavaScript: Introduction to JavaScript, Basic JavaScript Instructions, Functions, Methods & Objects, Decisions and Loops, Document Object Model, Event handling, handling window pop-ups, JavaScript validation.

Module 4	**PHP – Application Level**	**15 Sessions**

Topics:

PHP: Introduction to server-side Development with PHP, Arrays, $GET and $ POST, $_Files Array, Reading/Writing Files, PHP Classes and Objects, Working with Databases, SQL, Database APIs, Managing a MySQL Database. Accessing MySQL in PHP.

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):

T1. Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 8th Edition,2015.
T2. CSS Notes for Professionals, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022)
T3. Deitel, Deitel, Goldberg,"Internet& World Wide Web How to Program", Fifth Edition, Pearson Education,2021.

References:

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.

Topics related to development of "FOUNDATION": Web, WWW, Web browsers, Web servers, Internet.

CSS, PHP. Designing for healthcare.

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

E-References:

1. W3schools.com
2. Developer.mozilla.org/en-US/docs/Learn
3. docs.microsoft.com
4. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking
5. https://presiuniv.knimbus.com/user#/home

Course Code: **CSE2259**	**Course Title: Web Technologies Lab** **Type of Course: Laboratory Course (PCC)**	**L- T-P- C**	**0-0-2-1**
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course highlights the basic web design using Hypertext Markup Language and Cascading Style Sheets. Students will be trained in planning and designing effective web pages by writing code using current leading trends in the web domain, enhancing web pages with the use of page layout techniques, text formatting, graphics, images, and multimedia. The focus is on popular key technologies that will help students to build Internet- and web-based applications that interact with other applications and with databases.		
Course Objective	To develop skills in designing and building interactive web applications by enabling students to create web pages using HTML, CSS, and JavaScript, implement server-side scripting techniques, integrate applications with databases, and develop complete web-based solutions.		
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the core concepts of web technologies including the Internet, web servers, and browsers. [Understand] **CO2:** Develop well-structured web pages using XHTML with text formatting, images, links, lists, tables, forms, and frames in accordance with web standards. [Apply] **CO3:** Design responsive and structured web page layouts using advanced CSS techniques such as positioning, floating, and multicolumn layouts. [Apply]		

	CO4: Apply JavaScript concepts to develop dynamic and interactive web applications. [Apply] **CO5:** Implement dynamic server-side web applications using PHP, including control structures, arrays, superglobals, and file handling. [Apply] **CO6:** Develop modular web applications by integrating PHP with MySQL databases and applying object-oriented programming principles. [Create]

COURSE CONTENT:

List of Laboratory Tasks:30 Sessions

WEEK 1: Introduction to Web & XHTML

Lab Sheet 1: Web Basics

Level 1: Create a simple XHTML page explaining Web, WWW, browsers, and servers.

Level 2: Design a webpage illustrating Internet architecture using text and images.

Lab Sheet 2: XHTML Structure

Level 1: Create a basic XHTML document with proper syntax and structure.

Level 2: Design a webpage using headings, paragraphs, and formatting tags.

WEEK 2: XHTML Elements

Lab Sheet 3: Lists and Links

Level 1: Create ordered, unordered, and definition lists in XHTML.

Level 2: Create hyperlinks for navigation between multiple pages.

Lab Sheet 4: Images and Tables

Level 1: Insert images into a webpage.

Level 2: Design a table with rows, columns, and formatting.

WEEK 3: XHTML Forms & Frames

Lab Sheet 5: Forms

Level 1: Create a form with text fields and buttons.

Level 2: Design a registration form with validation requirements.

Lab Sheet 6: Frames

Level 1: Create a webpage using frames.

Level 2: Design a multi-frame layout for a website.

WEEK 4: Introduction to CSS

Lab Sheet 7: CSS Basics

Level 1: Apply inline and internal CSS styles.

Level 2: Create an external stylesheet and apply it to multiple pages.

Lab Sheet 8: Selectors

Level 1: Use element, class, and ID selectors.

Level 2: Apply multiple selectors for styling complex pages.

WEEK 5: CSS Properties

Lab Sheet 9: Font and Text

Level 1: Apply font properties (size, style, family).

Level 2: Design a styled webpage using text formatting.

Lab Sheet 10: Box Model

Level 1: Demonstrate margin, padding, and border.

Level 2: Design layout using box model concepts.

WEEK 6: Advanced CSS

Lab Sheet 11: Positioning

Level 1: Use static and relative positioning.

Level 2: Design layout using absolute and fixed positioning.

Lab Sheet 12: Floating & Layout

Level 1: Apply float properties.

Level 2: Create a multi-column layout using float.

WEEK 7: Responsive Design

Lab Sheet 13: Responsive Web Design

Level 1: Use media queries.

Level 2: Design a responsive webpage.

Lab Sheet 14: CSS Frameworks

Level 1: Use a CSS framework (Bootstrap basics).

Level 2: Design responsive webpage using Bootstrap.

WEEK 8: XML Basics

Lab Sheet 15: XML Structure

Level 1: Create a simple XML document.

Level 2: Display XML data in webpage.

Lab Sheet 16: XML Applications

Level 1: Validate XML structure.

Level 2: Use XML for data representation in web pages.

WEEK 9: JavaScript Basics

Lab Sheet 17: JavaScript Introduction

Level 1: Write basic JavaScript program for output.

Level 2: Perform arithmetic operations using JavaScript.

Lab Sheet 18: Control Statements

Level 1: Use if-else statements.

Level 2: Implement loops in JavaScript.

WEEK 10: Functions & DOM

Lab Sheet 19: Functions

Level 1: Create and call functions.

Level 2: Use functions with parameters and return values.

Lab Sheet 20: DOM Manipulation

Level 1: Access HTML elements using DOM.

Level 2: Modify content dynamically using JavaScript.

WEEK 11: Events & Validation

Lab Sheet 21: Event Handling

Level 1: Handle click events.

Level 2: Create interactive webpage using events.

Lab Sheet 22: Form Validation

Level 1: Validate input fields using JavaScript.

Level 2: Create complete form validation system.

WEEK 12: Advanced JavaScript

Lab Sheet 23: Objects & Methods

Level 1: Create JavaScript objects.

Level 2: Use methods and object properties.

Lab Sheet 24: Pop-ups

Level 1: Implement alert and prompt.

Level 2: Create dynamic pop-up-based interaction.

WEEK 13: PHP Basics

Lab Sheet 25: PHP Introduction

Level 1: Write basic PHP script.

Level 2: Handle form input using PHP.

Lab Sheet 26: PHP Control Structures

Level 1: Use if-else in PHP.

Level 2: Implement loops in PHP.

WEEK 14: PHP Applications

Lab Sheet 27: PHP Forms

Level 1: Process form data using PHP.

Level 2: Create login system using PHP.

Lab Sheet 28: PHP File Handling

Level 1: Read/write files using PHP.

Level 2: Store and retrieve data dynamically.

WEEK 15: Integration & Mini Project

Lab Sheet 29: Web Integration

Level 1: Combine XHTML, CSS, JavaScript.

Level 2: Build dynamic webpage using all technologies.

Lab Sheet 30: Mini Project

Level 1: Design a complete website.

Level 2: Develop full web application using XHTML, CSS, JS, PHP.

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):

T1. Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 8th Edition,2015.
T2. CSS Notes for Professionals, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022)
T3. Deitel, Deitel, Goldberg,"Internet& World Wide Web How to Program", Fifth Edition, Pearson Education,2021.

References:

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.

E-References:

1. W3schools.com
2. Developer.mozilla.org/en-US/docs/Learn
3. docs.microsoft.com
4. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking
5. https://presiuniv.knimbus.com/user#/home

Topics related to development of "FOUNDATION": Web, WWW, Web browsers, Web servers, Internet.

CSS, PHP. Designing for healthcare For Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CSE2260	**Course Title: Database Management Systems** **Type of Course: Theory Course (PCC)**	**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 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	To develop a strong foundation in database management systems by enabling students to understand database concepts and architecture, design relational databases using ER modeling, apply normalization techniques to eliminate redundancy, execute SQL queries					

	and transactions, and analyze query optimization and database performance.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental concepts of database systems, data models, schemas, and their role in modern applications. [Understand] **CO2:** Apply relational algebra operations to retrieve and manipulate data in relational databases. [Apply] **CO3:** Apply SQL queries and analyze query processing and optimization techniques. [Apply] **CO4:** Design and develop database systems by applying normalization principles, and demonstrate transaction processing, recovery, and security mechanisms. [Apply] **CO5:** Explain advanced database concepts and their applications in real-world scenarios. [Understand]

Course Content:

Module 1	**Introduction to Database Modelling and Relational Algebra**	**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**	**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 of expression, choosing evaluation plans, linear and bushy plans, dynamic programming algorithms.

Module 3	**Relational Database Design & Transaction Management**	**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**	**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, Native XML 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.

Assignments:

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. vandenBroucke 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 Lab** **Type of Course: Laboratory Course (PCC)**	**L-T-P-C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	NIL					
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	To develop practical skills in database systems by enabling students to design and implement relational database schemas, perform SQL queries and operations, implement constraints, triggers, and stored procedures, and develop database-driven applications.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Create databases, tables, and constraints using SQL Data Definition Language (DDL) commands. [Understand] **CO2:** Apply SQL Data Manipulation Language (DML) commands to insert, update, delete, and retrieve data from relational databases. [Apply] **CO3:** Implement advanced SQL queries using joins, subqueries, views, indexes, and aggregate functions to solve real-world problems. [Apply] **CO4:** Design and normalize relational schemas and enforce integrity constraints using keys, triggers, and stored procedures. [Apply] **CO5:** Perform database transaction management, including concurrency control, backup, recovery, and security mechanisms using DBMS tools. [Apply]					
Course Content:						
List of Laboratory Tasks:30 Sessions **WEEK 1: Introduction to Database Concepts** **Lab Sheet 1: DBMS Basics**						

Level 1: Identify schema and instance for a given database scenario.

Level 2: Analyze differences between traditional file systems and DBMS with examples.

Lab Sheet 2: Database Architecture

Level 1: Explain 3-schema architecture with diagram.

Level 2: Illustrate physical and logical data independence using examples.

WEEK 2: ER Modeling

Lab Sheet 3: ER Diagram Basics

Level 1: Draw ER diagram for a student database.

Level 2: Design ER diagram for a real-world system (library/hospital).

Lab Sheet 4: ER to Relational Mapping

Level 1: Convert ER diagram into relational schema.

Level 2: Handle complex relationships (many-to-many, weak entities).

WEEK 3: Relational Algebra

Lab Sheet 5: Basic Operations

Level 1: Perform selection and projection operations.

Level 2: Apply rename and set operations.

Lab Sheet 6: Advanced Operations

Level 1: Perform joins (inner join).

Level 2: Solve queries using outer joins and division operator.

WEEK 4: SQL Basics

Lab Sheet 7: DDL Commands

Level 1: Create and modify tables using SQL.

Level 2: Apply constraints (primary key, foreign key).

Lab Sheet 8: DML Commands

Level 1: Insert, update, delete records.

Level 2: Perform queries with conditions and operators.

WEEK 5: SQL Queries

Lab Sheet 9: Joins

Level 1: Write SQL queries using joins.

Level 2: Solve complex queries using multiple joins.

Lab Sheet 10: Aggregate Functions

Level 1: Use COUNT, SUM, AVG.

Level 2: Solve grouping queries with HAVING clause.

WEEK 6: Advanced SQL

Lab Sheet 11: Views

Level 1: Create and use views.

Level 2: Update data using views.

Lab Sheet 12: Procedures & Functions

Level 1: Write stored procedures.

Level 2: Implement user-defined functions.

WEEK 7: Triggers & Embedded SQL

Lab Sheet 13: Triggers

Level 1: Create trigger for insert/update.

Level 2: Design trigger for auditing changes.

Lab Sheet 14: Embedded & Dynamic SQL

Level 1: Demonstrate embedded SQL.

Level 2: Implement dynamic SQL queries.

WEEK 8: Query Optimization

Lab Sheet 15: Query Optimization Basics

Level 1: Explain query transformation techniques.

Level 2: Estimate cost of queries.

Lab Sheet 16: Query Plans

Level 1: Analyze linear execution plans.

Level 2: Compare bushy vs linear plans.

WEEK 9: Normalization

Lab Sheet 17: Normal Forms

Level 1: Convert schema to 1NF and 2NF.

Level 2: Convert schema to 3NF and BCNF.

Lab Sheet 18: Advanced Normalization

Level 1: Apply 4NF and 5NF.

Level 2: Analyze lossy vs lossless decomposition.

WEEK 10: Transactions

Lab Sheet 19: ACID Properties

Level 1: Explain ACID with examples.

Level 2: Analyze transaction schedules.

Lab Sheet 20: Concurrency Control

Level 1: Implement locking mechanisms.

Level 2: Analyze serializability and recoverability.

WEEK 11: Recovery Techniques

Lab Sheet 21: Logging

Level 1: Explain write-ahead logging.

Level 2: Simulate crash recovery.

Lab Sheet 22: Checkpointing

Level 1: Explain checkpointing.

Level 2: Design recovery process after failure.

WEEK 12: Advanced DB Concepts

Lab Sheet 23: NoSQL

Level 1: Compare SQL and NoSQL.

Level 2: Design document-based database.

Lab Sheet 24: Data Warehousing

Level 1: Explain data warehouse concepts.

Level 2: Design simple data warehouse schema.

WEEK 13: Advanced Database Systems

Lab Sheet 25: Object-Oriented DBMS

Level 1: Explain OODBMS concepts.

Level 2: Compare relational vs object-oriented DBMS.

Lab Sheet 26: Spatial & Temporal DB

Level 1: Explain spatial databases.

Level 2: Design temporal database application.

WEEK 14: Specialized Databases

Lab Sheet 27: Multimedia DB

Level 1: Explain multimedia database concepts.

Level 2: Design multimedia database system.

Lab Sheet 28: XML Databases

Level 1: Explain XML databases.

Level 2: Design document-oriented database.

WEEK 15: Case Study & Integration

Lab Sheet 29: Case Study

Level 1: Design database for real-world application.

Level 2: Optimize queries and schema.

Lab Sheet 30: Mini Project

Level 1: Implement database application.

Level 2: Integrate SQL, optimization, and transactions.

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.

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. vandenBroucke 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:	Course Title: Software Design and Development	L- T-P- C	3	0		3

CSE2287	**Type of Course: Theory Course (PCC)**				**0**	
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces foundational concepts of software engineering and emphasizes a practical, structured approach to software system development. It spans software requirement analysis, process modeling, system design, Agile principles, DevOps culture, testing strategies, quality assurance, and software maintenance. Hands-on assignments and problem-solving foster understanding of contemporary software practices and industry tools such as CASE tools, GitHub, and Selenium. AI-enhanced tools are introduced for documentation, automation, and intelligent testing workflows					
Course Objectives	To develop a comprehensive understanding of software engineering by enabling students to analyze software development life cycle models, apply design principles and patterns, develop modular and maintainable software systems, perform testing and quality assurance, and manage software projects effectively.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain software engineering principles, ethics, and various software development life cycle models. [Understand] **CO2:** Identify software requirements and develop design models, including Software Requirement Specification (SRS) documents. [Apply] **CO3:** Apply Agile development practices and DevOps principles in software development. [Apply] **CO4:** Apply software testing strategies, configuration management, and maintenance techniques. [Apply] **CO5:** Evaluate software design and development practices to ensure quality, reliability, and scalability. [Evaluate]					
Course Content:						
Module 1	**Introduction to Software Engineering and Process Models**				**10 Sessions**	

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.

AI Perspective: Use of AI-based modelling assistants (e.g., ChatGPT-based flow generation)

Module 2	**Software Requirements, Analysis and Design**	**12 Sessions**

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.

AI Perspective: Taskade AI SRS Generator/ Bit.ai AI SRS Generator

Module 3	**Agile Principles &Devops**	**10 Sessions**

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.

AI Perspective: AI assistants for sprint planning and backlog prioritization

Module 4	**Software Testing and Maintenance**	**13 Sessions**

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.

AI Perspective: LambdaTest AI Test Case Generator tool/ Testsigma, testRigor tool for QA.

Targeted Application & Tools that can be used:

Git hub/Lambda/Taskade/Bit.ai/Testsigma/TestRogor

Text Books:

T1. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", VII Edition, McGraw-Hill, 2017.

T2. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", VI Edition, McGraw-Hill, 2018.

References:

R1. Rajib Mall, "Fundamentals of Software Engineering", VI Edition, PHI learning private limited, 2015.

R2. Ian Sommerville, "Software Engineering", IX Edition, Pearson Education Asia, 2011.

R3. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002

Journals / Magazines

- IEEE Software
- ACM Transactions on Software Engineering and Methodology
- Journal of Systems and Software

SWAYAM / NPTEL / MOOCs

- NPTEL – Software Engineering (Prof. Rajib Mall, IIT KGP)
- Coursera – Agile Development (University of Virginia)
- Udemy – DevOps Masterclass
- EdX – Software Testing Fundamentals (TUM)

Topics Relevant to "Skill Development": Black box Testing, White box Testing, AI tools for design, modelling, testing and document Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout

Course Code: **CSE2269**	**Course Title: Operating Systems** **Type of Course: Theory Course (PCC)**	**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 Systems operations, Operating System structure and its design and implementation. It covers the 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 Objective	To develop a comprehensive understanding of operating system principles by enabling students to analyze process management and scheduling algorithms, explore memory and virtual memory management, examine file systems and storage organization and evaluate synchronization mechanisms and deadlock handling techniques.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental concepts of operating systems and relevant case studies. [Understand] **CO2:** Apply various CPU scheduling algorithms for process management. [Apply] **CO3:** Apply synchronization techniques to handle concurrency problems. [Apply] **CO4:** Apply deadlock detection and recovery methods in operating systems. [Apply] **CO5:** Explain various memory management techniques used in operating systems. [Apply]

Course Content:

Module 1	**Introduction to Operating Systems**	**9 Sessions**

Topics:

Introduction to OS , Operating-System Operations, Operating Systems Services, , System Calls and its types, Operating Systems Structure, System Program and its types, Linkers and Loaders, Overview of OS design and implementation, Open-source Operating Systems

Module 2	**Process Management**	**12 Sessions**

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**	**12 Sessions**

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**	**12 Sessions**

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 :

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

Course Code: CSE2270	Course Title: Operating Systems Lab Type of Course: Laboratory Course (PCC)	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 Objectives	To develop practical skills in operating systems by enabling students to implement scheduling and memory management algorithms, simulate process synchronization and deadlock handling, and build system-level programming capabilities.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Demonstrate system-level programming using system calls and operating system structures. [Apply] **CO2:** Simulate process scheduling and multithreading techniques. [Apply] **CO3:** Apply synchronization techniques using semaphores and shared memory to handle concurrency problems. [Apply] **CO4:** Demonstrate memory management and file system concepts using simulation or scripting tools. [Apply] **CO5:** Analyze the performance of operating system algorithms related to scheduling, synchronization and memory management. [Analyze]					

Course Content:

List of Laboratory Tasks:30 Sessions

WEEK 1: Introduction to OS

Lab Sheet 1: OS Basics

Level 1: Write a program to demonstrate basic system calls (file operations like open, read, write).

Level 2: Analyze different OS services with real-time examples.

Lab Sheet 2: OS Structure

Level 1: Illustrate OS structures (monolithic, layered, microkernel).

Level 2: Compare open-source OS (Linux) with proprietary OS.

WEEK 2: System Programs

Lab Sheet 3: Linkers & Loaders

Level 1: Demonstrate compilation process (preprocessing, compiling, linking).

Level 2: Analyze object files and executable generation.

Lab Sheet 4: System Utilities

Level 1: Write simple shell commands using system calls.

Level 2: Implement a mini shell program.

WEEK 3: Process Concept

Lab Sheet 5: Process Creation

Level 1: Write a program using fork() system call.

Level 2: Demonstrate parent-child communication.

Lab Sheet 6: Process Operations

Level 1: Implement process states and transitions.

Level 2: Simulate process control block (PCB).

EEK 4: Inter Process Communication

Lab Sheet 7: Pipes

Level 1: Implement communication using pipes.

Level 2: Implement bidirectional communication using pipes.

Lab Sheet 8: Sockets/RPC

Level 1: Write a client-server program using sockets.

Level 2: Simulate RPC communication.

WEEK 5: Threads

Lab Sheet 9: Thread Creation

Level 1: Create threads using pthread library.

Level 2: Compare multi-threading vs multi-processing.

Lab Sheet 10: Thread Issues

Level 1: Demonstrate thread synchronization issues.

Level 2: Solve race condition problem.

WEEK 6: CPU Scheduling

Lab Sheet 11: FCFS & SJF

Level 1: Implement FCFS scheduling.

Level 2: Implement SJF and compare results.

Lab Sheet 12: Advanced Scheduling

Level 1: Implement Round Robin scheduling.

Level 2: Implement Priority scheduling and analyze performance.

WEEK 7: Synchronization

Lab Sheet 13: Critical Section

Level 1: Implement Peterson's solution.

Level 2: Analyze critical section problem.

Lab Sheet 14: Semaphores

Level 1: Implement semaphore operations.

Level 2: Solve producer-consumer problem using semaphores.

WEEK 8: Classical Problems

Lab Sheet 15: Reader-Writer

Level 1: Implement Reader-Writer problem.

Level 2: Analyze starvation issues.

Lab Sheet 16: Dining Philosophers

Level 1: Implement Dining Philosophers problem.

Level 2: Avoid deadlock using strategies.

WEEK 9: Deadlocks

Lab Sheet 17: Deadlock Detection

Level 1: Implement Resource Allocation Graph.

Level 2: Detect deadlock using algorithm.

Lab Sheet 18: Deadlock Avoidance

Level 1: Implement Banker's algorithm.

Level 2: Analyze safe and unsafe states.

WEEK 10: Memory Management Basics

Lab Sheet 19: Memory Allocation

Level 1: Implement First Fit, Best Fit, Worst Fit.

Level 2: Compare memory utilization.

Lab Sheet 20: Paging

Level 1: Simulate paging mechanism.

Level 2: Analyze page table structure.

WEEK 11: Virtual Memory

Lab Sheet 21: Page Replacement

Level 1: Implement FIFO page replacement.

Level 2: Implement LRU and Optimal algorithms.

Lab Sheet 22: Page Faults

Level 1: Calculate page faults.

Level 2: Analyze thrashing condition.

WEEK 12: Advanced Memory Management

Lab Sheet 23: Segmentation

Level 1: Implement segmentation.

Level 2: Compare segmentation vs paging.

Lab Sheet 24: Demand Paging

Level 1: Simulate demand paging.

Level 2: Analyze copy-on-write mechanism.

WEEK 13: File System

Lab Sheet 25: File Operations

Level 1: Implement file operations (create, read, write).

Level 2: Simulate directory structure.

Lab Sheet 26: File Allocation

Level 1: Implement file allocation methods.

Level 2: Compare allocation techniques.

WEEK 14: File System Implementation

Lab Sheet 27: File System Design

Level 1: Design simple file system.

Level 2: Analyze file system performance.

Lab Sheet 28: Access Methods

Level 1: Implement sequential access.

Level 2: Implement indexed access.

WEEK 15: Integration & Simulation

Lab Sheet 29: OS Simulation

Level 1: Simulate OS components (process + memory).

Level 2: Integrate scheduling and memory management.

Lab Sheet 30: Mini Project

Level 1: Implement OS concept-based project.

Level 2: Analyze performance and optimization.

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].

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:

1. 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.

2. 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.

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

Course Code: CSE2502	**Course Title: Cryptography and Network Security** **Type of Course: Theory Course (PCC)**	**L-T-P-C**	**3**	**0**	**0**	**3**
Version No.	1					
Course Pre-requisites	CSE2251 Data Communication and Computer Networks					
Anti-requisites	NIL					
Course Description	The Course covers the principles and practice of cryptography and network security, focusing in particular on the security aspects of the web and Internet. The cryptographic tools such as shared key encryption, public key encryption, key exchange, and digital signature are explored. The use and utilization of the internet protocols and applications such as SSL/ TLS, IPSEC, Kerberos, PGP, and S/ MIME, SET are reviewed. System security issues such as viruses, intrusion and firewalls are also explored.					
Course Objective	To develop a comprehensive understanding of cryptography and network security by enabling students to analyze security threats and vulnerabilities, apply encryption, authentication, and hashing techniques, study security protocols and standards, and evaluate secure system design.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamentals of cryptography and security architecture. [Understand] **CO2:** Apply classical encryption techniques for securing data. [Apply] **CO3:** Analyze symmetric key cryptography algorithms and their applications. [Apply] **CO4:** Apply number theory concepts in cryptographic systems. [Apply] **CO5:** Demonstrate public key cryptography and authentication methods. [Apply]					

	CO6: Analyze and compare network security protocols and their applications. [Analyze]

Course Content:

Module 1	**Introduction to Cryptography and types of Ciphers**	**10 Sessions**

Topics:

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 techniques: Caesar cipher, Monoalphabetic cipher, Polyalphabetic cipher, Play-fair and Hill Cipher, Transposition techniques: Rail fence, Basics of Steganography.

Module 2	**Symmetric Key Cryptography and Number Theory**	**13 Sessions**

Topics:

Introduction to Block Cipher and Stream Cipher, Feistel Structure, Symmetric Encryption Algorithms: Data Encryption Standard, Introduction to Galois Field, Advanced Encryption Standard, Modular Arithmetic, Prime numbers, Fermat's little theorem, Primality testing: Miller-Rabin algorithm, Euclidean and Extended Euclidean Algorithm, Euler Totient Function, Chinese Remainder Theorem

Module 3	**Public Key Cryptography and its Applications**	**12 Sessions**

Topics:

Asymmetric Encryption and Decryption: Overview of Public Key Cryptography, Distribution of public keys, X.509 certificates, RSA, Diffie-Helman Key exchange, Man in the middle attack, Cryptographic Hash functions, Secure Hash Algorithm, Message Authentication Codes – HMAC, Digital Signature case studies.

Module 4	**Network Security**	**10 Sessions**

Topics:

Network Security applications: Authentication: Kerberos, PKI. E-mail security: PGP, S/MIME. Web Security: Secure Socket Layer (SSL), Transport Layer Security (TLS). IP Security: IPSec Policy, Encapsulating Security Payload (ESP).

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.

PEDAGOGY PLANNED WITH TOPICS:

1. Problem Solving: Playfair and Hill cipher
2. Problem Solving & Participative Learning: DES, AES, RSA & Diffie Hellman
3. Self-learning: Man in the Middle attack.
4. Flip Class: Network security and Email security

Textbooks:

1. William Stallings, “Cryptography and Network Security - Principles and Practices”, Prentice Hall, 7th Edition, 2017.

Reference Books:

1. Behrouz A Forouzan, Debdeep Mukhopadhyay, “Cryptography and Network Security”, McGraw Hill, third edition, 2010.

2. R. Rajaram, “Network Security and Cryptography” SciTech Publication. 3rd Edition, 2014.

3. Atul Kahate, “Cryptography and Network Security”, Tata McGraw-Hill, 2nd Edition, 2019.

4. Bruce Schneier, “Applied Cryptography”, John Wiley and Sons Inc. Second Edition, 2015.

Web references:

1. https://onlinecourses.nptel.ac.in/noc22_cs90/preview

2. e-pgpathshala UGC lecture series: E-Series and Self learning Materials.

https://epgp.inflibnet.ac.in/Home/ViewSubject?catid=fBYckQKJvP3a/8Vd3L08tQ==

3. http://182.72.188.195/cgi-bin/koha/opacdetail.pl?biblionumber=10133

4. http://182.72.188.195/cgi-bin/koha/opacdetail.pl?biblionumber=5875

Topics relevant to “Skill Development”: Symmetric and Asymmetric Encryption Algorithms and its problems.

Course Code: ISE2503	**Course Title: Information Theory and Coding** **Type of Course: PCC - Theory**	**L-P-C**	**3**	**0**	**0**	**3**
Version No.	1.1					
Course Pre-requisites	NIL					
Anti-requisites	NIL					

Course Description	Information Theory is the science for measuring, preserving, transmitting, and estimating information in random data. It was initially proposed by Shannon as a mathematical theory of communication more than five decades ago. It provides the fundamental limits of performance for transmission of messages generated by a random source over a noisy communication channel. On the one hand, Information Theory has been the driving force behind the revolution in digital communication and has led to various practical data compression and error correcting codes that meet the fundamental theoretical limits of performance. On the other hand, over the years, techniques and concepts from Information Theory have found applications well beyond communication theory. In this course, we will introduce the basic notions and results of Information Theory, keeping in mind both its fundamental role in communication theory and its varied applications beyond communication theory. This course, and the follow-up advanced courses to be offered in the future, will be of interest to students from various backgrounds.	
Course Objective	The objective of the course is to familiarize the learners with the concepts of Information Theory and Coding and attain Employability through Problem Solving Methodologies.	
Course Out Comes	On successful completion of the course the students shall be able to: **CO1:** **Calculate** the entropy of zero-memory sources, **analyze** Markov sources, and **apply** the properties of entropy for given source statistics*(Analyze)* **CO2:** **Determine** codewords and **compute** coding efficiency using Shannon, Shannon-Fano, Huffman, and Arithmetic coding techniques for memoryless sources, and **apply** the Lempel-Ziv (LZ) algorithm for sources with memory *(Analyze)* **CO3:** **Analyze** channel entropy, mutual information, and channel capacity for Discrete Memoryless Channels (DMC) based on given channel models, and **interpret** Shannon-Hartley Law and Shannon's Limit*(Understand)* **CO4:** **Determine** codewords, syndromes, and error detection and correction capabilities for Linear Block Codes and Cyclic Codes, and **design** single-error-correcting linear block codes for given message lengths*(Create)* **CO5:** **Evaluate** convolutional encoder outputs, **apply** Sequential Search and Viterbi decoding algorithms, and **analyze** advanced error control codes including BCH, Reed-Solomon, Golay, Turbo, and burst error-correcting codes*(Analyze)*	
Course Content:		
Module 1	**Information Theory**	**12 Sessions**
Topics: Introduction, Measure of information, Average information content (entropy) of symbols in long independent sequences, Information rate, Properties of entropy, Extension of discrete memory less (zero-memory) sources, Average information content (entropy) of symbols in long dependent sequences, Mark		

off statistical model for information source, Entropy and information rate of Mark off sources.

Module 2	**Source Coding**	**12 Sessions**

Topics:

Properties of codes- Block codes, on-singular codes, Uniquely decodable codes. Instantaneous codes and Optimal codes, Prefix of a code, Test for instantaneous property, Construction of Instantaneous code, Decision tree, Kraft's inequality, Source coding theorem (Shannon's Noiseless coding theorem), Shannon's encoding algorithm, Shannon Fano Algorithm, Huffman minimum redundancy code (binary, ternary and quaternary), Code efficiency and redundancy, Extended Huffman Coding, Arithmetic Codes, Lempel – Ziv Algorithm.

Module 3	**Channels and Mutual Information**	**12 Sessions**

Topics:

Introduction, Discrete communication channels, Representation of a channel, Probability relations- Apriori, Posteriori entropy, Equivocation, Mutual information, Properties, Rate of information transmission over a discrete channel, Capacity of a discrete memoryless channel, Shannon‟s theorem on channel capacity (Shannon‟s second theorem), Special channels- Symmetric, Binary symmetric, Binary erasure, Noiseless, Deterministic and cascaded channels, Estimation of channel capacity by Muroga‟s method, Continuous channels, Shannon-Hartley theorem and its implications, Shannon‟s limit, Rate Distortion Theory.

Module 4	**Linear Block Codes**	**9 Sessions**

Topics:

Introduction to Fields and Vector Spaces, Types of errors, Examples, Methods of controlling errors, Types of codes, Linear Block Codes- Matrix description, Encoding circuit, Syndrome and error detection, Syndrome circuit, hamming weight, hamming distance, Minimum distance of a block code error detection and correction capabilities of a linear block code, Single error-correcting Hamming codes, Table lookup decoding using standard array, General decoder for a linear block code. Binary cyclic codes: Algebraic structures of cyclic codes, Encoding using (n-k) bit shift register, Syndrome calculation.

Text Book

1. K. Sam Shanmugham,"Digital and Analog Communication Systems", John Wiley Publications, 1996.
2. Simon Haykin, "Digital Communications", John Wiley Publications, 2003.
3. Shu Lin, Daniel J. Costello, "Error Control Coding", Pearson / Prentice Hall, 2ndEdition, 2004.

References

1. Muralidhar Kulkarni and K. S. Shivaprakasha, "Information Theory and Coding", Wiley (India), 2015.
2. Glover and Grant, "Digital Communications", Pearson 2nd Edition, 2008.
3. Abramson, "Information Theory &Coding", McGraw-Hill, 1963.
4. Weblinks: pu.informatics.global.

Topics relevant to development of "EMPLOYABILITY SKILL": Algebraic structures of cyclic codes, Encoding using (n-k) bit shift register, Syndrome calculation, for developing Employability Skills through

Problem Solving Techniques. This is attained through assessment component mentioned in course handout.

Course Code: ISE2504	**Course Title: Image Processing and Computer Vision** **Type of Course: Discipline Elective - Theory**	**L- T-P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2264					
Anti-requisites	NIL					
Course Description	This course provides an introduction to the fundamental principles and techniques of digital image processing and computer vision. It emphasizes both theoretical foundations and practical applications including image enhancement, segmentation, feature extraction, object detection, and recognition, preparing students for real-world computer vision challenges in areas like robotics, medical imaging, and autonomous systems.					
Course Objective	To understand the fundamentals of digital images and image processing techniques. To explore computer vision algorithms for object detection, tracking, and classification. To implement practical solutions using modern image processing libraries and tools. To apply image and video analytics for real-time intelligent systems.					
Course Outcomes	On successful completion of the course, students will be able to: **CO1:** Apply appropriate image enhancement and transformation techniques to improve image quality and representation. **CO2:** Analyze and segment digital images using suitable algorithms to identify meaningful regions and structures. **CO3:** Analyze, extract, and evaluate feature descriptors for effective matching and classification of image data. **CO4:** Implement and develop computer vision models for object detection, tracking, and recognition tasks. **CO5:** Evaluate the performance of computer vision systems using appropriate metrics and design optimized solutions for real-world applications.					
Course Content:						
Module 1	**Fundamentals of Digital Image Processing**	**Assignment**			**[8] Sessions**	

Topics:

Image formation and perception, ampling, quantization, and pixel relationships, Intensity transformations and histogram processing, Spatial filtering: smoothing and sharpening, Frequency domain processing using Fourier transforms.

Module 2	**Image Segmentation and Morphological Processing**	**Assignment**		**[7] Sessions**

Topics:

Thresholding, region growing, and watershed algorithms, Edge detection: Sobel, Prewitt, Canny, Morphological operations: dilation, erosion, opening, closing, Connected components labelling, Segmentation evaluation techniques.

Module 3	**Feature Extraction and Matching**	**Assignment**		**[7] Sessions**

Topics:

Texture and color features, Corner and blob detection: Harris, FAST, DoG, SIFT, SURF, Feature descriptors and matching (BRIEF, ORB, FLANN), Hough transform for lines, circles, Motion detection and optical flow.

Module 4	**Object Detection, Recognition, and Vision Applications**	**Assignment**		**[8] Sessions**

Topics:

Object detection using Haar cascades and HOG, Face detection and facial recognition, Deep learning in vision: CNNs and YOLO, Video surveillance and real-time tracking, Applications: Medical imaging, AR/VR, Robotics, Self-driving vehicles.

REFERENCE MATERIALS

TEXTBOOKS

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Edition, Pearson Education, 2018.
2. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2nd Edition, 2022. https://szeliski.org/Book/

REFERENCE BOOKS

1. Sonka, Hlavac, and Boyle, Image Processing, Analysis, and Machine Vision, Cengage Learning, 4th Edition, 2014.
2. Mark Nixon and Alberto Aguado, Feature Extraction and Image Processing for Computer Vision, Academic Press, 3rd Edition, 2012.
3. Adrian Rosebrock, Practical Python and OpenCV, PyImageSearch Press, 2016.
4. Gary Bradski and Adrian Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library,

O'Reilly Media, 2nd Edition, 2016.

JOURNALS / MAGAZINES

1. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)
   https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34
2. Computer Vision and Image Understanding (Elsevier)
   https://www.sciencedirect.com/journal/computer-vision-and-image-understanding
3. International Journal of Computer Vision (Springer)
   https://www.springer.com/journal/11263

SWAYAM / NPTEL / MOOCs

1. NPTEL – Computer Vision (IIT Hyderabad) – Prof. P.J. Narayanan
   https://nptel.ac.in/courses/106106210
2. Coursera – Introduction to Computer Vision (Georgia Tech)
   https://www.coursera.org/learn/computer-vision-basics
3. edX – Fundamentals of Digital Image and Video Processing (Northwestern University)
   https://www.edx.org/course/digital-image-and-video-processing
4. Udacity – Intro to Computer Vision with OpenCV and Python
   https://www.udacity.com/course/introduction-to-computer-vision--ud810

Course Code: ISE2505	**Course Title: Image Processing and Computer Vision Lab** **Type of Course: Discipline Elective - Laboratory**	**L- T-P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	CSE2264					
Anti-requisites	NIL					
Course Description	This lab course provides hands-on experience in the foundational techniques and applications of digital image processing and computer vision. Students will gain practical skills in reading, manipulating, and analyzing images using Python, OpenCV, and deep learning frameworks to solve real-world visual tasks.					
Course Objective	To provide practical exposure to image acquisition, processing, and enhancement To understand and implement key computer vision algorithms To apply feature extraction, segmentation, and classification techniques					

	To develop small-scale vision-based applications using Python and OpenCV
Course Outcomes	Upon successful completion, students will be able to: **CO1:** Apply image transformation and enhancement techniques to preprocess and improve image quality. *(Apply)* **CO2:** Implement image segmentation and object detection methods for analyzing visual data. *(Apply)* **CO3:**Analyze and extract distinctive visual features and perform feature matching across images. *(Analyze)* **CO4:** Develop vision-based applications using OpenCV and Convolutional Neural Network (CNN) models. *(Create)* **CO5:** Evaluate the performance of computer vision models using appropriate metrics and optimize them for real-world applications. *(Evaluate)*

List of Laboratory Tasks

Lab 1: Read and Display Images using OpenCV
Objective: Load and visualize images using OpenCV
Task: Read image files and convert between color spaces
Activity: Display images in grayscale, RGB, and HSV formats

Lab 2: Image Enhancement Techniques
Objective: Apply contrast and brightness enhancement
Task: Use histogram equalization and contrast stretching
Activity: Compare results on different image datasets

Lab 3: Perform Image Filtering and Smoothing
Objective: Reduce noise using filters
Task: Apply Gaussian, median, and bilateral filters
Activity: Observe effects of different kernels on noisy images

Lab 4: Edge Detection Techniques
Objective: Detect edges using gradient-based methods
Task: Use Sobel, Prewitt, and Canny edge detectors
Activity: Tune thresholds and visualize edge maps

Lab 5: Geometric Transformations of Images
Objective: Apply image transformation techniques
Task: Perform translation, rotation, scaling, and affine transformations
Activity: Warp an image using transformation matrices

Lab 6: Image Thresholding and Binarization

Objective: Segment images using intensity values Task: Apply global, adaptive, and Otsu's thresholding Activity: Visualize segmentation results and contours
Lab 7: Morphological Operations Objective: Use structuring elements for shape-based processing Task: Apply dilation, erosion, opening, and closing Activity: Clean up binary images using morphological filters
Lab 8: Color-based Object Detection Objective: Segment objects based on color Task: Use HSV color space for object detection Activity: Detect colored objects (e.g., red ball) in real-time
Lab 9: Feature Detection using SIFT and ORB Objective: Extract and match image features Task: Use SIFT and ORB for keypoint detection Activity: Match features between two images and draw correspondences
Lab 10: Object Tracking in Videos Objective: Track moving objects in video streams Task: Implement object tracking using meanshift or CSRT Activity: Track objects with bounding boxes in webcam input
Lab 11: Face Detection using Haar Cascades Objective: Detect faces in images and videos Task: Load pre-trained Haar classifiers Activity: Draw bounding boxes around detected faces
Lab 12: Image Classification using Pre-trained CNNs Objective: Classify images using deep learning Task: Use MobileNet, VGG, or ResNet for classification Activity: Load and predict labels for input images
Lab 13: Implement Background Subtraction Objective: Separate foreground from background in video Task: Use frame differencing or background subtraction algorithms Activity: Detect motion and isolate moving objects
Lab 14: Create an Image Stitching Pipeline Objective: Stitch overlapping images into panoramas Task: Detect and match features, compute homography Activity: Combine images to form a wide-view panorama

Lab 15: Mini Project – Build a Real-Time Computer Vision Application Objective: Apply techniques learned to solve a real-world problem Task: Choose a task (e.g., face mask detection, barcode scanner) Activity: Develop and present the complete working prototype
REFERENCE MATERIALS **TEXTBOOKS** 1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 4th Edition, Pearson Education, 2018 2. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2022 https://szeliski.org/Book/
REFERENCE BOOKS 1. Sonka, Hlavac, and Boyle, Image Processing, Analysis, and Machine Vision, Cengage Learning, 2014 2. Adrian Rosebrock, Practical Python and OpenCV, PyImageSearch Press 3. Gary Bradski & Adrian Kaehler, Learning OpenCV, O'Reilly, 2nd Edition, 2016

Course Code: **ISE2507**	**Course Title: Machine Learning Lab** **Type of Course: 1] Program Core** **2] Laboratory**	**0**	**0**	**4**	**2**
Version No.	1.0				
Course Pre-requisites	CSE1001				
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 cover both the theoretical foundations as well as the essential algorithms for the various learning methods. Lab sessions complement the lectures and enable the students to develop intelligent systems for real life problems.				
Course Objectives	This course is designed to improve the learners 'EMPLOYABILITY SKILLS' by using EXPERIENTIAL LEARNING techniques.				

Course Out Comes	On successful completion of the course the students shall be able to: **CO1:** Apply advanced supervised machine learning methods for predictive modeling [Apply] **CO2:** Employ advanced unsupervised learning algorithms for clustering, competitive learning and outlier detection [Apply] **CO3:** Build machine learning models with better predictive performance using Ensemble learning algorithms [Apply] **CO4:** Build predictive models using Perceptron learning algorithms [Apply] **CO5:** Design and evaluate intelligent machine learning solutions by integrating multiple algorithms and optimizing model performance for real-world applications [Evaluate]

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, Word Cloud

Experiment No. 3: Regression learning

Level 1: Given a data set from UCI repository, implement the simple linear regression algorithm and estimate the model's 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 based classification model.

Level 2: Given data sets from UCI repository, implement a non-linear SVM based classification model.

Experiment No. 7: Unsupervised Learning

Level 1: K–means – simple and mini-batch. Finding the optimal number of clusters using Elbow method and Silhouette 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. 8: Density Based Clustering

Level 1: Implement DBSCAN – clustering using the local density estimation.

Level 2: Perform hard and soft clustering for new instances.

Experiment No. 9: Ensemble Learning using Subset of Instances

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. 10: Ensemble Learning using Subset of Features

Level 1: Apply ensemble learning techniques such as AdaBoost and Gradient Boosting **Level 2:** Apply ensemble learning techniques such as Stacking

Experiment No. 11: Perceptron Learning

Level 1: Implement the Perceptron Classifier

Level 2: An Image Classifier Using the Sequential API of Keras

Experiment No. 12: Outlier Detection

Level 1: Outlier Detection using Isolation Forest

Level 2: Outlier Detection using 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:

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.

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

1. Giuseppe Bonaccorso, "Machine Learning Algorithms: A reference guide to popular algorithms from data science and machine learning", Packt Publishing, 2017

Course Code: **ISE2502**	**Course Title: Information Retrieval** **Type of Course: PCC - Theory**	**L- P- T-C**	**3**	**0**	**0**	**3**
Version No.	1					
Course Pre-requisites	CSE2253					
Anti-requisites	NIL					
Course Description	The course studies the theory, design and implementation of Text- based information systems. The Information Retrieval core concepts of the course include statistical characteristics of text, representation of information needs and documents. Topics Include Several important retrieval models (Basic IR Models, Boolean Model, TF-IDF (Term Frequency/Inverse Document Frequency) Weighting, Vector Model, Probabilistic Model, Latent Semantic Indexing Model, Neural Network Model). Retrieval Evaluation, Retrieval Metrics, Text Classification and Clustering algorithms, Web Retrieval and Crawling. Recommender Systems: Basics of Content-based Recommender Systems, Content-based						

	Filtering, Collaborative Filtering, Matrix factorization models and neighborhood models.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of Information Retrieval and attain SKILL DEVELOPMENT through Participative Learning techniques			
Course Out Comes	On successful completion of the course the students shall be able to: **CO1:** Define basic concepts of information Retrieval [Knowledge] **CO2:** Evaluate the effectiveness and efficiency of different information retrieval methods [Application] **CO3:** Explain different indexing methodology requirements and the concept of web retrieval and crawling [Comprehension] **CO4:** Classify different recommender system and its aspect [Comprehension] **CO5:** Design and develop an intelligent information retrieval or recommender system using appropriate algorithms and evaluation techniques **[Create]**			
Course Content:				
Module 1	**Introduction to Information Retrieval**	**Assignment**	**Data collection**	**12 Sessions**
Topics: Information Retrieval – Early Developments – The IR Problem – The Users Task – Information versus Data Retrieval – The IR System – The Software Architecture of the IR System – The Retrieval and Ranking Processes				
Module 2	**Modeling and Retrieval Evaluation**	**Assignment**	**Problem solving**	**12 Sessions**
Topics: Basic IR Models – Boolean Model – TF-IDF (Term Frequency/Inverse Document Frequency) Weighting – Vector Model – Probabilistic Model – Latent Semantic Indexing Model – Neural Network Model – Retrieval Evaluation – Retrieval Metrics – Precision and Recall – Reference Collection – User-based Evaluation – Relevance Feedback and Query Expansion – Explicit Relevance Feedback.				
Module 3	**Indexing & Web-Retrieval**	**Term paper/Assignment**	**Data analysis**	**12 Sessions**
Topics: Indexing and Searching – Inverted Indexes – Sequential Searching – Multi-dimensional Indexing. The Web – Search Engine Architectures – Cluster based Architecture - Search Engine Ranking – Link based Ranking – Simple Ranking Functions, Evaluations — Search Engine Ranking – Applications of a Web Crawler.				
Module 4	**Recommender System**	**Term paper/Assignment**	**Problem solving**	**9 Sessions**
Topics: Recommender Systems Functions – Data and Knowledge Sources – Recommendation Techniques – Basics of Content-based Recommender Systems – High Level Architecture – Advantages and Drawbacks of				

Content-based Filtering – Collaborative Filtering – Matrix factorization models.

Targeted Application & Tools that can be used:

Information Retrieval System, Collaborative Filtering System, Feedback System, Evaluation Metrics

Assignment:

Group assignment, Quiz

Text Book

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —" Modern Information Retrieval: The Concepts and Technology behind Search", Third Edition, ACM Press Books, 2018. Link: https://people.ischool.berkeley.edu/~hearst/irbook/
2. Ricci, F, Rokach, L. Shapira, B.Kantor, —"Recommender Systems Handbook", Fourth Edition, 2018.

References

1. Stefan Buettcher, Charles L. A. Clarke and Gordon V. Cormack, —"Information Retrieval: Implementing and Evaluating Search Engines", The MIT Press, 2017.
2. Jian-Yun Nie Morgan & Claypool –" Cross-Language Information Retrieval", Publisher series 2011.
3. Stefan M. Rüger Morgan & Claypool – "Multimedia Information Retrieval", Publisher series 2014.
4. B. Liu, Springer, - "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data", Second Edition, 2013.
5. C. Manning, P. Raghavan, and H. Schütze, —"Introduction to Information Retrieval", Cambridge University Press, 2015. Link: https://nlp.stanford.edu/IR-book/

Web Based Resources and E-books:

https://puniversity.informaticsglobal.com/login

Topics relevant to the development of SKILLS: Recommender Systems, Content-based Filtering, Collaborative Filtering, Matrix factorization models and neighborhood models for Skill Development through Participative Learning Techniques. This is attained through assessment component mentioned in course handout.

Course Code:CSE2266	**Course Title: Theory of Computation Type of Course: Theory Course (PCC)**	**L- T-P-C**	**3**	**0**	**0**	**3**
Version No.	2.0					
Course Pre-requisites	NIL					
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	To develop a strong theoretical foundation in formal languages and automata by enabling students to analyze finite and pushdown automata, explore Turing machines and computability, evaluate complexity and decidability of problems, and apply these concepts to language design.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental components and concepts of automata. [Understand] **CO2:** Construct finite automata for given languages. [Apply] **CO3:** Differentiate between regular grammars and context-free grammars. [Understand] **CO4:** Construct pushdown automata for given languages. [Apply] **CO5:** Design Turing machines for given languages. [Apply]

Course Content:

Module 1	**Introduction to automata theory**	**6 Sessions**

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**	**12 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**	**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**	**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**	**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: ISE2500	**Course Title: Software Testing and Quality Assurance** **Type of Course: Theory**	**L- T-P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2287					
Anti-requisites	NIL					
Course Description	This course provides a comprehensive understanding of software testing strategies and quality assurance practices essential for delivering reliable and high-quality software products. Students will learn various testing techniques, software quality metrics, standards, and tools that support automation and continuous integration in the software development lifecycle.					

Course Objective	To understand the principles and practices of software testing. To introduce various levels and types of testing strategies. To equip students with knowledge of quality assurance methodologies and standards. To provide hands-on knowledge of software testing tools and frameworks.			
Course Outcomes	**On successful completion of the course, students will be able to:** **CO1:** Design and implement effective software test strategies. **CO2:**Analyze and evaluate testing coverage and effectiveness. **CO3:** Apply QA practices, including reviews, audits, and process assessments. **CO4:** Use automated testing tools for enhancing software reliability. **CO5:**Develop and optimize intelligent software testing frameworks by integrating AI-driven tools and automation techniques to enhance test efficiency, accuracy, and scalability.			
Course Content:				
Module 1	**Introduction to Software Testing**	**Assignment**		**[11] Sessions**
Topics: Software Development Life Cycle (SDLC) and Testing Life Cycle (STLC), Testing objectives and principles, Test levels: Unit, Integration, System, Acceptance, Test types: Functional, Non-functional, Regression, Smoke, and Sanity testing, Software Defect life cycle.				
Module 2	**Test Design and Execution Techniques**	**Assignment**		**[12] Sessions**
Topics: Black-box and White-box testing, Equivalence class partitioning, Boundary value analysis, Decision table, State transition, and Use case-based testing, Code coverage, Path testing, Mutation testing, Test case design and test data preparation.				
Module 3	**Software Quality Assurance (SQA)**	**Assignment**		**[11] Sessions**
Topics: SQA fundamentals and importance, Reviews, Walkthroughs, and Audits, Quality metrics and defect prevention, Process models: CMMI, ISO 9001, Six Sigma, Software configuration management and version control.				
Module 4	**Test Automation**			**[11]**

	and Tools			**Sessions**

Topics:

Automation Testing: Introduction and benefits, Selenium, JUnit, TestNG frameworks, Continuous Integration tools: Jenkins, GitLab CI/CD, Performance Testing using JMeter, Case studies and real-world QA practices.

REFERENCE MATERIALS:

TEXTBOOKS

1. Ron Patton, Software Testing, 2nd Edition, Pearson Education, 2005.
2. Paul Ammann and Jeff Offutt, Introduction to Software Testing, Cambridge University Press, 2nd Edition, 2016.

REFERENCES

1. Cem Kaner, Jack Falk, Hung Q. Nguyen, Testing Computer Software, Wiley, 2nd Edition, 1999.
2. Glenford J. Myers, The Art of Software Testing, Wiley, 3rd Edition, 2011.
3. Naresh Chauhan, Software Testing: Principles and Practices, Oxford University Press, 2010.
4. Pressman & Maxim, Software Engineering: A Practitioner's Approach, McGraw-Hill, 8th Edition, 2014 – Selected Chapters.

JOURNALS / MAGAZINES

IEEE Transactions on Software Engineering
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=32

Software Quality Journal (Springer)
https://www.springer.com/journal/11219

ACM Transactions on Software Engineering and Methodology (TOSEM)
https://dl.acm.org/journal/tosem

SWAYAM / NPTEL / MOOCs

NPTEL – Software Testing (IIT Kharagpur)
https://nptel.ac.in/courses/106105150

Coursera – Software Testing and Automation (University of Minnesota)
https://www.coursera.org/learn/software-testing-automation

Udemy – Automated Software Testing with Python
https://www.udemy.com/course/automated-software-testing-with-python/

edX – Software Testing Fundamentals (TUM)
https://www.edx.org/course/software-testing

Course Code: ISE2501	**Course Title: Software Testing and Quality Assurance Lab** **Type of Course: Laboratory**	**L- T-P- C**	**0**	**0**	**2**	**1**
Version No.	1.0					
Course Pre-requisites	CSE2287					
Anti-requisites	NIL					
Course Description	This laboratory course offers hands-on practice in software testing techniques and tools to ensure software quality and reliability. Students will implement unit, integration, system, and regression tests, and gain experience with manual and automated testing tools. Emphasis is placed on testing strategies, test case development, defect tracking, and quality assurance practices.					
Course Objective	To provide practical knowledge of software testing life cycle and strategies To enable test design and execution using black-box and white-box techniques To introduce automated testing tools for functional and performance testing To develop defect tracking, reporting, and quality assurance processes					
Course Outcomes	Upon successful completion, students will be able to: **CO1:** Design effective test cases for different levels of testing **CO2:** Execute manual and automated test procedures **CO3:**Analyze software quality through metrics and reports **CO4:** Use software testing tools to automate and validate results **CO5:**Evaluate and optimize software testing strategies by integrating advanced testing tools and frameworks to ensure quality, reliability, and efficiency of software systems.					

List of Laboratory Tasks

Lab 1: Introduction to Software Testing Tools
Objective: Understand basic tools and environment for testing
Task: Explore tools like JUnit, Selenium, Bugzilla
Activity: Set up testing environment for future labs

Lab 2: Requirement Analysis and Test Case Design
Objective: Design test cases from software requirements
Task: Derive test scenarios and test cases from a sample SRS
Activity: Document test cases in standard format

Lab 3: Black-Box Testing Techniques
Objective: Apply functional testing methods
Task: Perform equivalence partitioning and boundary value analysis
Activity: Design and execute test cases on login module

Lab 4: White-Box Testing Techniques
Objective: Explore code-based testing methods
Task: Perform statement and branch coverage testing

Activity: Generate test data for decision structures in Java/Python

Lab 5: Unit Testing with JUnit
Objective: Write and execute unit tests
Task: Create unit test cases using JUnit framework
Activity: Test Java functions and assert expected outputs

Lab 6: Integration Testing
Objective: Test interaction between modules
Task: Use stubs and drivers to simulate integration
Activity: Execute integration test cases on a student registration system

Lab 7: System and Acceptance Testing
Objective: Perform end-to-end testing
Task: Prepare test cases for system-level testing
Activity: Validate expected output on sample system software

Lab 8: Regression Testing
Objective: Validate unchanged functionality after changes
Task: Rerun previously passed test cases after updates
Activity: Identify regression defects and document reports

Lab 9: Automation with Selenium WebDriver
Objective: Automate web application testing
Task: Create scripts to test login and navigation
Activity: Validate elements and capture screenshots

Lab 10: Performance Testing using Apache JMeter
Objective: Simulate and evaluate system load
Task: Run load tests for concurrent users
Activity: Generate reports and interpret performance metrics

Lab 11: Bug Reporting using Bugzilla
Objective: Track software defects
Task: Report and manage bugs in Bugzilla
Activity: Update bug status and assign severity

Lab 12: Test Management using TestLink
Objective: Manage and document testing activities
Task: Organize test plans, cases, and execution
Activity: Report test coverage and result summaries

Lab 13: Code Coverage Analysis
Objective: Measure code coverage using tools
Task: Use tools like JaCoCo or Coverage.py
Activity: Analyze coverage reports and improve test suite

Lab 14: Software Quality Metrics Calculation
Objective: Compute quality assurance metrics
Task: Calculate defect density, defect leakage, test effectiveness
Activity: Analyze metrics for project quality reporting

Lab 15: Capstone Project – Test a Web Application
Objective: Apply full testing lifecycle
Task: Plan, design, execute, and report tests
Activity: Submit complete test documentation for the selected app

REFERENCE MATERIALS

TEXTBOOKS

1. Ron Patton, Software Testing, 2nd Edition, Pearson Education, 2005
2. Paul Ammann and Jeff Offutt, Introduction to Software Testing, Cambridge University Press, 2nd Edition, 2016

REFERENCE BOOKS

1. Glenford J. Myers, The Art of Software Testing, Wiley, 3rd Edition, 2011
2. Naresh Chauhan, Software Testing: Principles and Practices, Oxford University Press, 2010
3. Cem Kaner et al., Testing Computer Software, Wiley, 2nd Edition, 1999

Course Code: CSE2262	**Course Title: Analysis of Algorithms** **Type of Course: Theory Course (PCC)**	**L- T-P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
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	To develop the ability to design and analyze efficient algorithms by enabling students to evaluate time and space complexity, apply various algorithmic paradigms such as divide-and-conquer, greedy, and dynamic programming, and compare different approaches for solving computational problems.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the efficiency of algorithms in terms of time and space complexity. [Understand] **CO2:** Apply divide-and-conquer and decrease-and-conquer techniques to solve searching and sorting problems. [Apply] **CO3:** Apply dynamic programming and greedy techniques to solve computational problems. [Apply] **CO4:** Apply time–space trade-off techniques and analyze the limitations of algorithms for problem solving. [Apply] **CO5:** Analyze problems using backtracking and branch-and-bound techniques to develop optimal solutions. [Analyze]					
Course Content:						
Module 1	**Introduction**					**14 Sessions**

Introduction: Notion of algorithm, Importance of algorithm, Important Problem Types, Fundamentals of analysis of algorithm efficiency: General equation, Order of growth, basic efficiency classes, asymptotic notations, mathematical analysis of recursive & non-recursive algorithms, Brute Force technique: Bubble sort, selection sort, linear search, brute force string matching, Exhaustive search: Travelling salesperson problem, knapsack problem.

Module 2	**Divide-and-conquer**	**11 Sessions**

Divide & Conquer technique: General Method, General recursive equation, Masters Theorem, Binary search, Merge sort, quick sort. Decrease & Conquer technique: General method, insertion sort, topological sorting

Module 3	**Dynamic programming**	**13 Sessions**

Dynamic Programming: General method, Warshall algorithm, Floyd algorithm, 0/1 knapsack, Greedy method: General method, prim's algorithm, Kruskal algorithm, Dijkstra's algorithm, Huffman coding

Module 4	**Space and time tradeoff:**	**11 Sessions**

Space and time tradeoff: introduction, Input enhancement in string matching: Horspool algorithm and Boyer-Moore algorithm, Limitations of algorithm power: Introduction, lower bound arguments, decision trees, P, NP and NP-complete problems

Module 5	**Coping with limitations of algorithm power:**	**11 Sessions**

Coping with limitations of algorithm power: Introduction, Backtracking: n-queens problem, subset-sum problem, Hamiltonian circuit problem, Branch & bound: Travelling salesperson problem, knapsack problem, Job assignment problem.

Text 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.

Reference Books:

R1. AV Aho, J Hopcroft, JD Ullman, *"The Design and Analysis of Algorithms"*, Addison-Wesley, 1974

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

R3. J. Kleinberg and E. Tardos, "*Algorithm Design*", Addison-Wesley, 2005

Online Resources:

1. EBook: https://presiuniv.knimbus.com/user#/home
2. https://onlinecourses.nptel.ac.in/noc26_cs67/
3. www.youtube.com/watch?v=so4zT-u09LM&list=PL8GCet93Cu1iRF611mrUu-Rhy12fMwZOf

Topics 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.

<table>
<tr><td>Course Code:

CSE2263</td><td>Course Title: Analysis of Algorithms Lab

Type of Course: Laboratory Course (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">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">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.</td></tr>
<tr><td>Course Objective</td><td colspan="6">To develop practical skills in algorithm design and analysis by enabling students to implement algorithms using programming languages, evaluate their performance through experimentation, compare the efficiency of different approaches, and solve optimization problems using appropriate algorithmic techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="6">On successful completion of this course, students will be able to:

CO1:
Compute the efficiency of algorithms in terms of time and space complexity. [Apply]

CO2:
Apply divide-and-conquer techniques to solve searching and sorting problems. [Apply]

CO3:
Apply dynamic programming techniques to solve computational problems. [Apply]

CO4:
Apply greedy techniques for solving optimization problems. [Apply]

CO5:
Apply backtracking techniques and analyze the limitations of algorithms for problem solving. [Apply]</td></tr>
<tr><td colspan="7">Course Content</td></tr>
<tr><td colspan="7">List of Laboratory Tasks:30 Sessions

WEEK 1: Algorithm Basics & Analysis

Lab Sheet 1: Algorithm Fundamentals

Level 1: Write an algorithm and program to find minimum and maximum elements in a list.

Level 2:Analyze the number of comparisons and derive time complexity.</td></tr>
</table>

Lab Sheet 2: Asymptotic Notations

Level 1: Write programs to demonstrate linear and quadratic time complexity.

Level 2: Compare growth rates of different functions using Big-O, Ω, and Θ.

WEEK 2: Efficiency Analysis

Lab Sheet 3: Recursive Algorithms

Level 1: Write a recursive program for factorial.

Level 2: Solve recurrence relations for recursive algorithms.

Lab Sheet 4: Non-Recursive Analysis

Level 1: Write programs with nested loops and count operations.

Level 2: Derive general equations and order of growth.

WEEK 3: Brute Force Techniques

Lab Sheet 5: Sorting Algorithms

Level 1: Implement Bubble Sort and Selection Sort.

Level 2: Compare their time complexity experimentally.

Lab Sheet 6: Searching & String Matching

Level 1: Implement Linear Search.

Level 2: Implement brute-force string matching algorithm.

WEEK 4: Exhaustive Search

Lab Sheet 7: Knapsack Problem

Level 1: Solve 0/1 Knapsack using brute force.

Level 2: Analyze solution space complexity.

Lab Sheet 8: Travelling Salesperson Problem

Level 1: Solve TSP using brute force approach.

Level 2: Compare performance for different inputs.

WEEK 5: Divide and Conquer

Lab Sheet 9: Binary Search

Level 1: Implement Binary Search.

Level 2: Analyze recurrence relation.

Lab Sheet 10: Merge Sort

Level 1: Implement Merge Sort.

Level 2: Analyze using Master's Theorem.

WEEK 6: Advanced Divide & Conquer

Lab Sheet 11: Quick Sort

Level 1: Implement Quick Sort.

Level 2: Compare best and worst cases.

Lab Sheet 12: Master's Theorem

Level 1: Solve recurrence equations.

Level 2: Apply Master's Theorem to different cases.

WEEK 7: Decrease and Conquer

Lab Sheet 13: Insertion Sort

Level 1: Implement Insertion Sort.

Level 2: Compare with Bubble Sort.

Lab Sheet 14: Topological Sorting

Level 1: Implement Topological Sort.

Level 2: Apply on real-world dependency graph.

WEEK 8: Dynamic Programming

Lab Sheet 15: Warshall Algorithm

Level 1: Implement Warshall algorithm.

Level 2: Compute transitive closure.

Lab Sheet 16: Floyd Algorithm

Level 1: Implement Floyd's algorithm.

Level 2: Compare with Dijkstra.

WEEK 9: DP Applications

Lab Sheet 17: Knapsack DP

Level 1: Solve 0/1 Knapsack using DP.

Level 2: Compare with brute force approach.

Lab Sheet 18: DP Optimization

Level 1: Identify overlapping subproblems.

Level 2: Convert recursive solution to DP.

WEEK 10: Greedy Algorithms

Lab Sheet 19: MST Algorithms

Level 1: Implement Prim's algorithm.

Level 2: Implement Kruskal's algorithm.

Lab Sheet 20: Shortest Path

Level 1: Implement Dijkstra's algorithm.

Level 2: Compare greedy vs DP approach.

WEEK 11: Huffman Coding

Lab Sheet 21: Huffman Tree

Level 1: Construct Huffman Tree.

Level 2: Analyze compression efficiency.

Lab Sheet 22: Greedy Analysis

Level 1: Demonstrate greedy choice property.

Level 2: Prove correctness of greedy algorithms.

WEEK 12: String Matching Optimization

Lab Sheet 23: Horspool Algorithm

Level 1: Implement Horspool algorithm.

Level 2: Compare with brute-force method.

Lab Sheet 24: Boyer-Moore Algorithm

Level 1: Implement Boyer-Moore algorithm.

Level 2: Analyze efficiency improvements.

WEEK 13: Complexity Theory

Lab Sheet 25: Lower Bounds

Level 1: Analyze decision tree for sorting.

Level 2: Derive lower bounds.

Lab Sheet 26: NP Problems

Level 1: Classify problems as P or NP.

Level 2: Analyze NP-complete problems.

WEEK 14: Backtracking

Lab Sheet 27: N-Queens

Level 1: Solve N-Queens problem.

Level 2: Analyze search tree.

Lab Sheet 28: Subset Sum

Level 1: Solve subset sum problem.

Level 2: Compare with DP solution.

WEEK 15: Branch & Bound

Lab Sheet 29: TSP

Level 1: Solve TSP using Branch & Bound.

Level 2: Compare with brute-force approach.

Lab Sheet 30: Job Assignment

Level 1: Solve assignment problem.

Level 2: Analyze pruning efficiency.

Targeted Application & Tools that can be used :

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

Course Code: CSE7000	**Course Title: Internship** **Type of Course: PRW**	**L- T-P- C**	**0**	**0**	**0**	**2**
Version No.	1.0					
Course Pre-requisites	Knowledge and Skills related to all the courses studied in previous semesters.					
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.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skillsthrough Experiential Learningtechniques.					
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:** Identify engineering problems related to local, regional, national, or global needs. [Understand] **CO2:** Apply appropriate techniques and modern tools to solve engineering problems. [Apply] **CO3:** Design experiments in accordance with standard procedures and specifications. [Analyze] **CO4:** Conduct experiments and collect relevant data for analysis. [Apply] **CO5:**					

	Interpret experimental results and draw meaningful conclusions. [Evaluate]

Course Code: CSE7100	**Course Title: Mini Project** **Type of Course: PRW**	**L- T-P- C**	**0**	**0**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	Knowledge and Skills related to all the courses studied in previous semesters.					
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 Skillsthrough Experiential Learningtechniques.					
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:** Identify engineering problems related to local, regional, national, and global needs. [Understand] **CO2:** Apply appropriate techniques and modern tools to solve the intended problems. [Apply] **CO3:** Design experiments in accordance with standards and specifications. [Analyze] **CO4:** Interpret events and results to derive meaningful conclusions. [Evaluate] **CO5:** Appraise project findings and communicate effectively through scholarly publications. [Create]					

Course Code: CSE7300	**Course Title: Capstone Project** **Type of Course: PRW**	**L- T-P- C**	**0**	**0**	**0**	**10**
Version No.	1.0					
Course Pre-requisites	Knowledge and Skills related to all the courses studied in previous semesters.					
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 Skillsthrough Experiential Learningtechniques.					
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:** Identify problems based on societal /research needs. [Understand] **CO2:** Apply Knowledge and skill to solve societal problems in a group. [Apply] **CO3:** Develop interpersonal skills to work as member of a group or leader. [Apply] **CO4:** Analyze the inferences from available results through theoretical / Experimental / Simulations. [Analyze] **CO5:** Analyze the impact of solutions in societal and environmental context for sustainable development. [Analyze] **CO6:** Improve in written and oral communication. [Create]					

Course Code: **CHE7601**	**Environmental Studies** **Type of Course: MOOC course**	**L- T- P- C**	**1**	**0**	**0**	**0**
Course Pre-requisites	NIL					
Anti-requisites	**NIL**					
Course 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**					
Course Objective	The objective of the course is **'SKILL DEVELOPMENT'** of the student by using **'PARTICIPATIVE LEARNING'** techniques					
Course 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					
Course Content:						
Module 1	**Understanding Environment, Natural Resources, and Sustainability**	**3 Sessions**				
Topics: Classification 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. Concept 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.						
Module 2	**Ecosystems, Biodiversity, and Sustainable Practices**	**4 Sessions**				
Topics: **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.

Module 3	**Environmental Pollution, Waste Management, and Sustainable Development**	**4 Sessions**

Topics:

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.

Module 4	**Social Issues, Legislation, and Practical Applications**	**4 Sessions**

Topics:

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).

Targeted Application & Tools that can be used:

Application areas are Energy, Environment and sustainability

Tools: Online Tools – NPTEL and Swayam.

Project work/Assignment:

Assessment Type

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

Online 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.

* Other source links are available in below Resources link.

Text Book

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

Reference 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

The topics related to Environment and Sustainability :

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

Course Code: **LAW7601**	**Indian Constitution** **Type of Course: MAC course**	**L-T- P- C**	**1 – 0 – 0 - 0**
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course 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.**		
Course Objective	The objective of the course is **'SKILL DEVELOPMENT'** of the student by using **'PARTICIPATIVE LEARNING'** techniques		
Course Outcomes	On successful completion of this course, students shall be able to: **CO1:** Describe the fundamental concepts of the Indian Constitution and relate them to contemporary societal issues, while recognizing its core values. [Understand] **CO2:** Explain citizen-centric rights and responsibilities in relation to the State. [Understand] **CO3:** Explain the role of State actors in nation-building. [Understand] **CO4:** Interpret Gandhian perspectives on the significance of Local Self-Governance (LSG). [Understand]		
Course Content:			
Module 1	**Understanding the Making of the Constitution: The Constituent Assembly & The Constitution of India**		**3 Sessions**
Topics: 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			
Module 2	**Citizen's Fundamental Rights and State's Responsibilities (Directive Principles)**		**4 Sessions**
Topics: 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

Module 3	Organs Of the Government	4 Sessions

Topics:

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

Module 4	Federalism & Decentralization	4 Sessions

Topics:

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)

Targeted Application & Tools that can be used:

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

Tools: Online Tools – NPTEL and Swayam.

Project work/Assignment:

Assessment Type

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

Online 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.

Text 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)

Reference 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

The topics related to Constitutional Studies and its application :

All topics in theory component are relevant to Indian Constitution.

Course Code: CIV7601	**Course Title: Universal Human Values and Ethics** **Type of Course: MAC course**	**L-T-P-C**	**1**	**0**	**0**	**0**
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using 'SELF LEARNING' techniques					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the importance of value education through the process of self-exploration. [Understand] **CO2:**					

	Explain the concept of human beings as the co-existence of the self and the body in harmony. [Understand] **CO3:** Describe the role of foundational values in building harmonious relationships. [Understand] **CO4:** Summarize the importance of a holistic perspective in developing ethical professional behavior. [Understand]

Course Content:

Module 1	**Introduction to Value Education**	**3 Sessions**

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.

Module 2	**Harmony in the Human Being**	**4 Sessions**

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

Module 3	**Harmony in the Family and Society**	**4 Sessions**

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.

Module 4	**Implications of the Holistic Understanding – A Look at Professional Ethics**	**4 Sessions**

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

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.

Project work/Assignment:

Assessment Type

Online exams (MCQs) will be conducted by the Department of Civil Engineering through Linways.

Online Link*:

1) UHV II - https://www.youtube.com/watch?v=NhFBzn5qKIM&list=PLWDeKF97v9SO8vvjC1KyqteziTbTjN1So&pp=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: PPS1025	**Course Title: Industry Readiness Program – I** **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"EmployabilityforYoungProfessionals"andattainSKILLDEVELOPMENT through PARTICIPATIVE LEARNING techniques.					
CourseOut Comes	On successful completion of this course, students shall be able to: **CO1:**Define their career goals. [Remember] **CO2:**Practice ethical habits for achieving career success. [Apply] **CO3:**Demonstrate effective email writing techniques. [Apply]					

Course Content

Module1	**Goal Setting & Grooming**	**10 Sessions**
Topics:SMART Goals, formal grooming through self-introduction activity **Activity:**Realworld scenarios		
Module2	**Habit Formation**	**10 Sessions**
Topics: Professional and Personal ethics for success and activity-based practice **Activity:**Students to present 2 min video on building professional ethics		
Module3	**Email Etiquettes**	**10 Sessions**
Topics: Types of prompts to generate effective or desired results for email etiquettes **Activity:**Individualstudent presenting various search prompts		

TargetedApplication&Toolsthatcanbe used:

1. TED Talks
2. YouTube Links

Activities

Assignmentproposedforthis course Assignment 1: SMART Goal Assignment 2: AI tools for prompt search
ContinuousIndividualAssessment Module 1: Presentation Module 2: Activity based assessment Module 3: Class assessment
Thetopicsrelatedtoskilldevelopment: 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: PPS1026	**Course Title: Industry Readiness Program – II** **Type of Course: Practical Only Course**	**L- T - P- C**	**0**	**0**	**2**	**0**
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	NIL					
Course Description	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.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of"Industry ReadinessforYoungProfessionals"andattain SKILLDEVELOPMENT through PARTICIPATIVE LEARNING techniques.					
Course Out Comes	On successful completion of this course, students shall be able to: **CO1:** Apply effective communication skills for success in the workplace. [Apply] **CO2:** Practice team-building skills to enhance career success. [Apply] **CO3:** Demonstrate ethical leadership skills in the workplace. [Apply]					

Course Content

Module 1	**Effective Communication**	**Classroomactivities**	**10 Sessions**
Topics: Practice effective communication skills (Verbal, Non-verbal, Written and Visual) **Activity:**Use social media prompts to prepare self-introduction videos			
Module 2	**Team Building**	**Group Activity**	**10 Sessions**
Topics: Skills of an effective team player **Activity:**Student group activity to build class networking			
Module 3	**Leadership**	**Case study**	**10 Sessions**
Topics: Types of leadership, using empathy in leadership **Activity:**Individualpresentation by students on corporate leaders.			
Faculty : L&D			

TargetedApplication&Toolsthatcanbe used: 1. TED Talks 2. YouTube Links Activities
Assignmentproposedforthis course Assignment 1: One minute reel Assignment 2: Team building assignment
ContinuousIndividualAssessment Module 1: L-S-R-W class assessment Module 2: Team Presentation Module 3: Individual Assessment
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.

CourseCode : APT4002	**CourseTitle:Introductionto Aptitude**			**L-T-P- C**	0	0	2	0
Version No.	1.0							
CoursePre - equisites								
Anti-requisites	Nil							
Course Description	Theobjectiveofthiscourseistopreparethetraineestotacklethequestionson 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 higherorderthinkingquestions.Thefocusofthiscourseistoteachthestudents tonotonlygettothe correct answers,butto get there fasterthanever before, which will improve their employability factor.							
Course Objective	The objective of the course is to familiarize the learners with the concepts ofAptitude and attain Skill Development through Problem Solving techniques.							
Course Outcomes	Onsuccessfulcompletionofthecoursethestudentsshallbeableto: CO1]**Recall**allthebasicmathematicalconceptstheylearntinhighschool. CO2] **Identify** the principle concept needed in a question. CO3] **Solve** the quantitative and logical ability questions with the appropriate concept. CO4]**Analyze**thedatagivenincomplex problems. CO5]**Rearrange**theinformation tosimplify thequestion							
CourseContent:								
Module1	Quantitative Ability						**12Sessions**	
Topics: IntroductiontoAptitude,workingofTables,Squares,Cubes								
Module2	Logical Reasoning						**18Sessions**	
Topics: Linear&CircularArrangementPuzzle,Coding&Decoding,BloodRelations,Directions,Orderingand Ranking, Clocks and Calendars,Number Series, Wrong number series, Visual Reasoning								

TargetedApplication&Toolsthatcanbeused: Applicationarea:PlacementactivitiesandCompetitiveexaminations. Tools: LMS
TextBook 1. QuantitativeAptitudebyRS Aggarwal 2. Verbal&Non-VerbalReasoningbyRSAggarwal
References 1. www.indiabix.com 2. www.youtube.com/c/TheAptitudeGuy/videos
Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill DevelopmentthroughProblemsolvingTechniques.Thisisattainedthroughassessment componentmentionedincoursehandout.

Course Code: **APT4004**	**CourseTitle:AptitudeTraining- Intermediate** TypeofCourse:PracticalOnlyCourse	**L- T - P- C**	0	0	2	0
VersionNo.						
CoursePre-requisites	Studentsshouldhavethebasicconceptsof Quantitativeaptitudealongwithits applicationsinreallifeproblems.					
Anti-requisites	NIL					
Course Description	Thisisaskill-basedtrainingprogramforthestudents.Thiscourseisdesigned to enable the students to enhance their skills in Quantitative Aptitude.					
Course Objective	**Theobjectiveofthecourseistofamiliarizethelearnerswiththeconceptsof Aptitude and attain Skill Development through Problem Solving techniques.**					
CourseOut Comes	**Onsuccessfulcompletionofthiscoursethestudentsshallbeableto:** CO1: Recall allthebasicmathematicalconcepts. CO2:Identifytheprincipleconceptneededinaquestion. CO3: Solve the quantitative and logical ability questions with the appropriateconcept. CO4:Analyzethedatagivenincomplex problems.					
Course Content:						
Module1	**QuantitativeAbility1**					**16Sessions**
Topics: Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss						
Module2	**QuantitativeAbility2**					**14Sessions**
Topics: TimeSpeedandDistance,BoatsandStreams,SimpleInterest,CompoundInterest,Probability, Permutation and Combination						
Targeted Application & Tools that can be used: Applicationarea:PlacementactivitiesandCompetitiveexaminations. Tools: LMS						
ContinuousEvaluation:						
CA1–OnlineTest CA2–OnlineTest CA3–OnlineTest Assignment						
Text Book:						

1. FastTrackObjectivebyRajesh Verma
2. RSAggarwal
3. RakeshYadav

References:

1. www.indiabix.com
2. www.testbook.com
3. www.youtube.com/c/TheAptitudeGuy/videos

TopicsrelevanttoSkillDevelopment:QuantitativeaptitudeforSkillDevelopmentthroughProblem solving Techniques. This is attained through components mentioned in course handout.

<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>1</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] Analyzeand structure the reasoning techniques and spatial visualizationskills</td></tr>
<tr><td colspan="8">Course Content:</td></tr>
<tr><td>Module 1</td><td colspan="5">Logical Thinking</td><td colspan="2">16 Sessions</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 colspan="5">Critical Thinking</td><td colspan="2">14 Sessions</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>
</table>

References

1. www.indiabix.com
2. www.testbook.com
3. www.youtube.com/c/TheAptitudeGuy/videos

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.

<table>
<tr><td>Course Code:
CAI3427</td><td colspan="2">Course Title: Language Models for Text Mining

Type of Course: Lab Integrated 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="7">1.0</td></tr>
<tr><td>Course Pre- requisites</td><td colspan="7">CSE2264 Essentials of AI</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">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 such as Text Mining, NLP, Tokenization, Lemmatization, Stemming, One-hot encoding, Language modelling, Bag-of-words, Term-document Matrix, Cosine similarity, Viterbi Algorithm, etc.</td></tr>
<tr><td>Course Objectives</td><td colspan="7">To develop a strong foundation in natural language processing by enabling students to understand language models and NLP techniques, apply text mining and analysis methods, utilize transformer-based models, and analyze textual data for meaningful insights.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of this course, students will be able to:

CO1:
Process and preprocess textual data to extract meaningful information. [Apply]

CO2:
Apply insights derived from textual data to solve real-world problems. [Apply]

CO3:
Develop solutions for NLP tasks using machine learning and deep learning techniques. [Apply]

CO4:
Utilize NLP tools, libraries, and language models for text mining applications. [Apply]

CO5:
Analyze and evaluate the performance of language models and text mining techniques across different applications. [Analyze]</td></tr>
<tr><td colspan="8">Course Content:</td></tr>
<tr><td>Module 1</td><td colspan="5">Text Mining</td><td colspan="2">8 Sessions</td></tr>
<tr><td colspan="8">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).</td></tr>
<tr><td>Module 2</td><td colspan="5">Text Preprocessing</td><td colspan="2">7 Sessions</td></tr>
<tr><td colspan="8">Introduction to Preprocessing. Tokenization. Stop Words Removal. Lemmatization and Stemming. PoS Tagging. Integer Encoding. Padding. One-Hot Encoding.</td></tr>
<tr><td>Module 3</td><td colspan="5">Text Representations</td><td colspan="2">8 Sessions</td></tr>
</table>

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**	**7 Sessions**

Word Embeddings vs. One-Hot Encoding. Contextual Bag of Words (CBOW). Skipgram. Deep Learning for Document Classification.

List of Laboratory Tasks: **30 Sessions**

Module: 1 (Lab Sheets 1–4)

Week 1: Lab Sheet 1 (File Handling & Text Extraction)

Level 1:
Write a Python program to read a text file and count the number of words, sentences, and characters.

Level 2:
Develop a program that reads multiple text files, extracts meaningful content (removes special characters, numbers), and stores cleaned text into a new file.

Week 2: Lab Sheet 2 (Text Cleaning & String Processing)

Level 1:
Write a program to remove punctuation, convert text to lowercase, and remove extra spaces.

Level 2:
Build a reusable text preprocessing pipeline function that performs cleaning, normalization, and token splitting.

Week 3: Lab Sheet 3 (Tokenization &Stopword Removal)

Level 1:
Use NLTK to tokenize a paragraph into words and remove stopwords.

Level 2:
Create a custom tokenizer that handles contractions (e.g., “don’t → do not”) and removes domain-specific stopwords.

Week 4: Lab Sheet 4 (Stemming & Lemmatization)

Level 1:
Implement stemming using Porter Stemmer and display results.

Level 2:
Compare stemming vs lemmatization and evaluate differences on a dataset.

Module: 2 (Lab Sheets 5–8)

Week 5: Lab Sheet 5 (POS Tagging & Encoding)

Level 1:
Perform POS tagging using SpaCy.

Level 2:
Convert tagged words into integer encoding and one-hot vectors.

Week 6: Lab Sheet 6 (Padding & Sequence Processing)

Level 1:

Implement sequence padding for sentences of different lengths.

Level 2:
Build a preprocessing pipeline that prepares sequences for deep learning models.

Week 7: Lab Sheet 7 (Bag of Words Model)

Level 1:
Implement Bag-of-Words using CountVectorizer.

Level 2:
Create a BoW-based text classifier using Naive Bayes.

Week 8: Lab Sheet 8 (TF-IDF & Similarity)

Level 1:
Compute TF-IDF vectors for a dataset.

Level 2:
Implement cosine similarity to find similar documents.

Module: 3 (Lab Sheets 9–12)

Week 9: Lab Sheet 9 (N-Gram Language Model)

Level 1:
Generate unigrams and bigrams from text.

Level 2:
Build a bigram language model to predict next word probability.

Week 10: Lab Sheet 10 (Naive Bayes Text Classification)

Level 1:
Implement Naive Bayes classifier using BoW features.

Level 2:
Evaluate model using accuracy, precision, recall, and confusion matrix.

Week 11: Lab Sheet 11 (Topic Modeling - LSA)

Level 1:
Apply Singular Value Decomposition (SVD) for dimensionality reduction.

Level 2:
Extract topics using Latent Semantic Analysis (LSA).

Week 12: Lab Sheet 12 (LDA Topic Modeling)

Level 1:
Implement LDA using Gensim.

Level 2:
Visualize topics and interpret topic distributions.

Module: 4 (Lab Sheets 13–15)

Week 13: Lab Sheet 13 (Word Embeddings)

Level 1:
Use pre-trained Word2Vec or GloVe embeddings.

Level 2:

Compute word similarity and analogy tasks.

Week 14: Lab Sheet 14 (Deep Learning for NLP)

Level 1:
Build a simple neural network for text classification using Keras.

Level 2:
Implement CBOW or Skip-gram model.

Week 15: Lab Sheet 15 (Transformer-based NLP Task)

Level 1:
Use a pre-trained transformer model for sentiment analysis.

Level 2:
Fine-tune a transformer model (e.g., BERT) on a custom dataset.

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):

1. 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).
2. Aditya Joshi, Pushpak Bhattacharyya. "Natural Language Processing", Wiley Publication, 2023 (1st Edition).

References:

1. R1. Chris Manning and Hinrich Schutze, "Foundations of Statistical Natural Language Processing", 1st Edition, MIT Press. 1999.
2. R2. Pawan Goyal. "Natural Language Processing". 1st Edition, 2016.

Weblinks

1. W1. E-Book link or R2: https://drive.google.com/file/d/10nbwAJd-dv6htOOZVBgAvLd1WscI0RqC/view
2. W2. Web Resource for T1: https://web.stanford.edu/~jurafsky/slp3/ - VERY VERY IMPORTANT!!!
3. 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** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2264 Essentials of AI					
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	To develop practical expertise in deep learning by enabling students to understand the TensorFlow framework, build and train neural network models, evaluate model performance, and deploy AI-based solutions.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamentals of neural networks, backpropagation, and gradient descent techniques. [Understand] **CO2:** Apply backpropagation and optimization techniques to train neural networks effectively. [Apply] **CO3:** Build and train deep learning models using Python libraries such as TensorFlow and Keras. [Apply] **CO4:** Apply deep learning techniques to solve real-world problems such as image classification, object detection, sentiment analysis, and language modeling. [Apply] **CO5:** Analyze and evaluate the performance of deep learning models and optimize them for improved accuracy and efficiency. [Analyze]					
Course Content:						
Module 1	**Basics of Neural Networks**				**10 Sessions**	
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.						
Module 2	**TensorFlow Basics**				**10 Sessions**	

Topics:

Introduction to TensorFlow, TensorFlow dataset, Machine Learning with TensorFlow

Module 3	Deep Learning methods with Tensor Flow and Keras	10 Sessions

Topics:

Main Features of TensorFlow, Keras basics, AI with Keras.

List of Laboratory Tasks: **30 Sessions**

Module: 1 (Lab Sheets 1–5)

Week 1: Lab Sheet 1 (Introduction to DL Frameworks)

Level 1:
Write a Python program to import TensorFlow, Keras, NumPy, and Matplotlib and display their versions. Perform basic tensor creation and visualization.

Level 2:
Develop a small program to compare TensorFlow and NumPy operations (matrix addition, multiplication) and visualize results using Matplotlib.

Week 2: Lab Sheet 2 (Perceptron Implementation)

Level 1:
Implement a single-layer perceptron for AND/OR logic gate using Python.

Level 2:
Implement a perceptron with adjustable learning rate and plot error convergence.

Week 3: Lab Sheet 3 (Multilayer Perceptron Basics)

Level 1:
Build a simple MLP using Keras for binary classification.

Level 2:
Tune hyperparameters (learning rate, epochs, hidden layers) and compare accuracy.

Week 4: Lab Sheet 4 (Activation Functions)

Level 1:
Implement and plot activation functions (ReLU, Sigmoid, Tanh).

Level 2:
Train a neural network with different activation functions and compare performance.

Week 5: Lab Sheet 5 (Backpropagation & Gradient Descent)

Level 1:
Implement gradient descent manually for a simple linear regression problem.

Level 2:
Visualize loss reduction and compare with TensorFlow optimizer.

Module: 2 (Lab Sheets 6–10)

Week 6: Lab Sheet 6 (Tensor Creation & Operations)

Level 1:

Create tensors using Python lists and NumPy arrays.

Level 2:
Perform tensor reshaping, slicing, and broadcasting operations.

Week 7: Lab Sheet 7 (TensorFlow Math Operations)

Level 1:
Perform arithmetic operations (add, subtract, multiply, divide) on tensors.

Level 2:
Apply advanced functions (log, exp, sqrt) and visualize outputs.

Week 8: Lab Sheet 8 (Tensor Concatenation & Stacking)

Level 1:
Concatenate tensors along different axes.

Level 2:
Stack tensors and build a mini dataset pipeline.

Week 9: Lab Sheet 9 (Dataset from CSV File)

Level 1:
Load a CSV dataset using tf.data and display samples.

Level 2:
Preprocess dataset (normalize, shuffle, batch).

Week 10: Lab Sheet 10 (TensorFlow Datasets Library)

Level 1:
Load dataset (MNIST/IMDB) using tfds.

Level 2:
Split dataset into training/testing and visualize samples.

Module: 3 (Lab Sheets 11–15)

Week 11: Lab Sheet 11 (Artificial Neural Network - ANN)

Level 1:
Build ANN for classification (e.g., diabetes dataset).

Level 2:
Evaluate model using confusion matrix and accuracy metrics.

Week 12: Lab Sheet 12 (CNN for Image Classification)

Level 1:
Build CNN for MNIST digit classification.

Level 2:
Extend CNN for Dog vs Cat classification.

Week 13: Lab Sheet 13 (Model Optimization Techniques)

Level 1:
Apply dropout and regularization.

Level 2:
Compare model performance before and after optimization.

Week 14: Lab Sheet 14 (RNN/LSTM for Sequential Data)

Level 1:
Build RNN model for sequence prediction.

Level 2:
Implement LSTM for sentiment analysis using IMDB dataset.

Week 15: Lab Sheet 15 (End-to-End Deep Learning Project)

Level 1:
Build a complete deep learning pipeline (data → training → evaluation).

Level 2:
Deploy model using TensorFlow (save/load model and prediction API).

Project work/Assignment:

1. Assignment 1 on (Module 1 and Module 2)
2. Assignment 2 on (Module 3)

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
2. https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385 IEEE Transactions on Pattern Analysis and Machine Intelligence
3. 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:

1. Swayam Nptel – Deep Learning – IIT Ropar https://onlinecourses.nptel.ac.in/noc21_cs35/preview
2. Coursera – Neural Networks and Deep Learning Andrew Ng
3. Coursera - Neural Networks for Machine Learning by Geoffrey Hinton in Coursera

<table>
<tr><td>Course Code: CAI3429</td><td colspan="2">Course Title: Deep Learning Techniques for Computer Vision

Type of Course: Lab Integrated 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="7">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">MAT2402 Probability and Statistics</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">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.</td></tr>
<tr><td>Course Objective</td><td colspan="7">To develop a strong foundation in computer vision by enabling students to understand core concepts, apply deep learning models for image processing, implement CNN-based solutions, and analyze vision-based applications.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of this course, students will be able to:

CO1:
Explain core concepts of neural networks and deep learning architectures for image processing. [Understand]

CO2:
Implement and optimize convolutional neural networks (CNNs) for image classification tasks. [Apply]

CO3:
Apply object detection and image segmentation techniques using models such as YOLO, Faster R-CNN, SSD, U-Net, and Mask R-CNN. [Apply]

CO4:
Utilize advanced deep learning techniques such as Vision Transformers (ViTs), attention mechanisms, and Generative Adversarial Networks (GANs). [Apply]

CO5:
Analyze, evaluate, and deploy deep learning models for real-world computer vision applications. [Analyze]</td></tr>
<tr><td colspan="8">Course Content:</td></tr>
<tr><td>Module 1</td><td colspan="4">Fundamentals of Deep Learning for Vision</td><td colspan="3">8 Sessions</td></tr>
<tr><td colspan="8">Topics:

Introduction to Deep Learning & Neural Networks, Convolutional Neural Networks (CNNs) Architecture Backpropagation & Optimization in CNNs, Transfer Learning & Pretrained Models.</td></tr>
<tr><td>Module 2</td><td colspan="4">Object Detection & Image Segmentation</td><td colspan="3">7 Sessions</td></tr>
<tr><td colspan="8">Topics:</td></tr>
</table>

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**	**8 Sessions**

Topics:

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**	**7 Sessions**

Topics:

Edge AI & Mobile Deployment (TensorFlow Lite, ONNX), Adversarial Attacks & Robustness in Vision Models, Explainability & Interpretability of Vision Models, Case Studies & Industry Applications

List of Laboratory Tasks: **30 Sessions**

Module: 1 (Lab Sheets 1–4)

Week 1: Lab Sheet 1 (Keras Sequential & Functional API)

Level 1:
Write a Python program using Keras Sequential API to build a simple neural network for image classification (MNIST dataset).

Level 2:
Implement the same model using Keras Functional API and compare training performance and flexibility.

Week 2: Lab Sheet 2 (Softmax Regression)

Level 1:
Implement softmax regression for multi-class classification using Keras.

Level 2:
Tune hyperparameters (learning rate, batch size, epochs) and analyze their impact on accuracy.

Week 3: Lab Sheet 3 (CNN for Grayscale Images)

Level 1:
Build a CNN model for grayscale image classification (MNIST/Fashion-MNIST).

Level 2:
Visualize feature maps and analyze convolution layer outputs.

Week 4: Lab Sheet 4 (CNN for Color Images)

Level 1:
Implement a CNN for CIFAR-10 dataset classification.

Level 2:
Apply data augmentation techniques and evaluate performance improvement.

Module: 2 (Lab Sheets 5–8)

Week 5: Lab Sheet 5 (Time Series & RNN Basics)

Level 1:

Implement exponential smoothing for time-series prediction.

Level 2:
Build an RNN model to predict future values of time-series data.

Week 6: Lab Sheet 6 (LSTM for Text Classification)

Level 1:
Implement LSTM model for document classification (multi-class).

Level 2:
Evaluate model performance using accuracy, precision, recall.

Week 7: Lab Sheet 7 (Binary Classification with LSTM)

Level 1:
Build LSTM model for binary sentiment classification.

Level 2:
Optimize model using dropout and compare results.

Week 8: Lab Sheet 8 (CNN + LSTM Hybrid Model)

Level 1:
Implement CNN + LSTM hybrid model for text classification.

Level 2:
Compare performance with standalone CNN and LSTM models.

Module: 3 (Lab Sheets 9–12)

Week 9: Lab Sheet 9 (Digit Recognition Models)

Level 1:
Implement softmax regression for handwritten digit recognition.

Level 2:
Build a deep neural network (MLP) and compare results with softmax model.

Week 10: Lab Sheet 10 (Object Detection using YOLO)

Level 1:
Use pre-trained YOLOv5 model to detect objects in images.

Level 2:
Fine-tune YOLO model on a custom dataset.

Week 11: Lab Sheet 11 (Image Segmentation using U-Net)

Level 1:
Implement U-Net model for basic image segmentation.

Level 2:
Train U-Net on custom dataset and evaluate segmentation metrics.

Week 12: Lab Sheet 12 (Faster R-CNN & Detection Models)

Level 1:
Implement object detection using Faster R-CNN.

Level 2:
Compare YOLO vs Faster R-CNN performance (speed vs accuracy).

Module: 4 (Lab Sheets 13–15)

Week 13: Lab Sheet 13 (Vision Transformers - ViT)

Level 1:
Implement Vision Transformer for image classification using pre-trained model.

Level 2:
Compare ViT with CNN in terms of accuracy and training time.

Week 14: Lab Sheet 14 (GANs for Image Generation)

Level 1:
Implement DCGAN to generate images (e.g., MNIST).

Level 2:
Enhance GAN performance and visualize generated samples.

Week 15: Lab Sheet 15 (Model Deployment & Explainability)

Level 1:
Deploy trained model and perform predictions on new images.

Level 2:
Implement Grad-CAM or similar technique for model explainability.

Tools/Software Required :

1. OpenCV 4
2. Python 3.7
3. MATLAB

Text Books:

1. "Deep Learning for Computer Vision Image Classification, Object Detection and Face Recognition in Python" Jason Brownlee (2019)
2. "Deep Learning for Computer Vision with python" Adrian Rosebrock (2017)

References:

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
2. A foundational book covering deep learning principles, including CNNs, optimization, and generative models.
3. Raschka, S., &Mirjalili, V. (2022). Machine Learning with PyTorch and Scikit-Learn. Packt Publishing.
4. Covers practical deep learning techniques using PyTorch, including CNNs and transfer learning.
5. Geron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd Edition). O'Reilly Media.
6. Provides hands-on implementations of deep learning for computer vision using TensorFlow and Keras.
7. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into Deep Learning. Available online (https://d2l.ai).
8. Open-access book covering CNNs, object detection, and advanced vision techniques with PyTorch and TensorFlow.
9. Chollet, F. (2021). Deep Learning with Python (2nd Edition). Manning Publications.
10. Explains deep learning fundamentals and applications with Keras, including image classification and segmentation.
11. Ballé, J., Laparra, V., & Simoncelli, E. P. (2017). Deep Learning for Computer Vision: A Brief Introduction.
12. A concise introduction to CNNs, object detection, and generative models.

Course Code: **CSE3426**	**Course Title: Front-end Full Stack Development** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2258 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	To develop proficiency in modern front-end development by enabling students to build responsive web applications, use contemporary frameworks and tools, integrate APIs and backend services, and design complete, user-friendly web interfaces.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamentals of DevOps principles, version control workflows, and front-end/full-stack development concepts. [Understand] **CO2:** Construct functional web pages using HTML5, CSS3, and JavaScript. [Apply] **CO3:** Implement responsive web designs using modern CSS frameworks such as Bootstrap or Tailwind and media queries. [Apply] **CO4:** Develop dynamic single-page applications (SPAs) using frameworks such as Angular by applying concepts like components, data binding, and routing. [Apply] **CO5:** Analyze and optimize front-end performance and deployment using techniques such as					

	code linting, unit testing, and CI/CD pipelines. [Analyze]

Course Content:

Module 1	**Fundamentals of DevOps**	**8 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**	**7 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**	**7 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**	**8 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).

List of Laboratory Tasks: **30 Sessions**

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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).

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Labsheet 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.

Targeted Application & Tools that can be used:

1. **Application Area:** Design and develop efficient and user-friendly front-end applications.
2. **Professionally Used Software:** GCC compiler (for understanding underlying compilation processes, though not directly used in front-end development).

Text Book:

1. **T1.** Fender, Young, "Front-end Fundamentals", Leanpub, 2015.
2. **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:

1. **R1.** Flanagan D S, "Javascript : The Definitive Guide" 7th Edition. 7th ed. O'Reilly Media; 2020.
2. **R2.** Alex Libby, Gaurav Gupta, and AsojTalesra. "Responsive Web Design with HTML5 and CSS3 Essentials", Packt Publishing, 2016.
3. **R3.** Duckett J Ruppert G Moore J. "Javascript&Jquery : Interactive Front-End Web Development."; Wiley; 2014.

Web Reference:

1. https://www.youtube.com/watch?v=JGNTYXkVCVY&list=PLd3UqWTnYXOkTSBCBNyyhxo_jxlY_uTWA&index=2
2. https://www.freecodecamp.org/news/frontend-web-developer-bootcamp/
3. https://puniversity.informaticsglobal.com:2229/login.aspx?direct=true&db=nlebk&AN=2233842&site=ehost-live (EBSCO database link, access may be restricted)
4. 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.

Course Code: CSE3427	**Course Title: Java Full Stack Development Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2255 Object Oriented Programming Using Java					
Anti-requisites	NIL					

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	To develop proficiency in full-stack application development using Java by enabling students to design and build integrated frontend, backend, and database systems, implement RESTful services, and deploy scalable web applications.	
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Apply Java programming for full-stack application development. [Apply] **CO2:** Implement web applications using Java EE technologies. [Apply] **CO3:** Apply Java Persistence API (JPA) and Hibernate to develop database-driven applications. [Apply] **CO4:** Apply Spring framework concepts to develop full-stack applications. [Apply] **CO5:** Demonstrate the use of automation tools such as Maven and Selenium in full-stack development. [Apply]	
Course Content:		
Module 1	**Introduction**	**5 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**	**7 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**	**6 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**	**6 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 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**	**6 Sessions**

Topics:

Introduction to Automation Tools; Apache Maven: Maven Fundamentals, Software Setup – Command line 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.

List of Laboratory Tasks: **30 Sessions**

Module 1: Fundamentals of DevOps (Lab Sheets 1–2)

Week 1: Lab Sheet 1 (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.

Week 2: Lab Sheet 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.

Module 2: Web Design & Development (Lab Sheets 3–5)

Week 3: Lab Sheet 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.

Week 4: Lab Sheet 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.

Week 5: Lab Sheet 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.

Module 3: Responsive Web Design (Lab Sheets 6–8)

Week 6: Lab Sheet 6 (Asynchronous JavaScript & 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.

Week 7: Lab Sheet 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).

Week 8: Lab Sheet 8 (Responsive 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.

Module 4: Angular Development (Lab Sheets 9–15)

Week 9: Lab Sheet 9 (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.

Week 10: Lab Sheet 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.

Week 11: Lab Sheet 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.

Week 12: Lab Sheet 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.

Week 13: Lab Sheet 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.

Week 14: Lab Sheet 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.

Week 15: Lab Sheet 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.

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:

1. **Problem Solving: Design of Algorithms and implementation of programs.**
2. **Programming: Implementation of given scenario using Java.**

Text Book:

T1 : Mayur Ramgir, "Full Stack Java Development with Spring MVC, Hibernate, jQuery , and Bootstrap", 1st Edition, Wiley Publication, 2020.

References

1. **R1:** Chris Northwood, "The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Developer", 1st edition, APress,2018.
2. **R2**: Herbert Schildt, "Java The complete reference" , 11th Edition, ORACLE, 2020

Course Code: **CSE3428**	**Course Title: .NET Full Stack Development** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2258 Web Technologies					
Anti-requisites	NIL					
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	To develop proficiency in full-stack application development using the .NET framework by enabling students to design and build integrated user interfaces, backend systems, and databases, implement web services and APIs, and deploy enterprise-level applications.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Apply C# programming concepts to develop basic applications. [Apply] **CO2:** Develop web applications using ASP.NET and Entity Framework for database interaction. [Apply] **CO3:** Design and implement web applications integrating SQL databases with ASP.NET technologies. [Apply] **CO4:** Apply full-stack development concepts to build end-to-end web applications using .NET technologies. [Apply] **CO5:** Analyze, test, and optimize full-stack applications for performance, usability, and scalability. [Analyze]					
Course Content:						
Module 1	**C# Programming for Full Stack Development**					**8 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**	**7 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**	**8 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	**Advanced ASP.NET**	**7 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).

List of Laboratory Tasks: 30 Sessions

Module 1 (Lab Sheets 1–4)

Week 1: Lab Sheet–1 (C# Basics & Console Applications)

Level 1:
Write a C# console application to input length and breadth of a rectangle and compute area and perimeter.

Level 2:
Develop a menu-driven C# program for a simple ATM system supporting deposit, withdrawal, and balance

inquiry with validation.

Week 2: Lab Sheet–2 (OOP Concepts in C#)

Level 1:
Create a class Student with properties (Name, RollNo, Marks) and display details using methods.

Level 2:
Design an abstract class Shape with derived classes Circle and Rectangle. Implement polymorphism to compute area and perimeter.

Week 3: Lab Sheet–3 (Collections and LINQ)

Level 1:
Use List<string> to store names and perform add, remove, and display operations.

Level 2:
Create a List<Employee> and use LINQ to filter employees by salary, sort by name, and compute average salary.

Week 4: Lab Sheet–4 (Exception Handling & Async Programming)

Level 1:
Write a program to handle divide-by-zero exception using try-catch-finally.

Level 2:
Implement an asynchronous method using async/await to simulate data fetching and handle exceptions gracefully.

Module 2 (Lab Sheets 5–7)

Week 5: Lab Sheet–5 (EF Core – Basic CRUD)

Level 1:
Create a console application using EF Core Code First approach to insert and display Product records.

Level 2:
Extend the application to include update and delete operations with a menu-driven interface.

Week 6: Lab Sheet–6 (EF Core Relationships)

Level 1:
Implement one-to-many relationship between Department and Employee entities and retrieve data.

Level 2:
Demonstrate Lazy Loading and Eager Loading and compare results.

Week 7: Lab Sheet–7 (EF Core Querying & Transactions)

Level 1:
Write LINQ queries to filter and sort products based on price.

Level 2:
Implement transaction handling for order processing involving multiple tables (Order, Product).

Module 3 (Lab Sheets 8–11)

Week 8: Lab Sheet–8 (ASP.NET Core MVC Basics)

Level 1:
Create a basic ASP.NET Core MVC application with a controller returning a static message.

Level 2:

Create a form to accept user input and display submitted data using MVC pattern.

Week 9: Lab Sheet–9 (State Management)

Level 1:
Implement session-based page visit counter.

Level 2:
Develop a multi-step registration form using Session and TempData.

Week 10: Lab Sheet–10 (EF Core with MVC)

Level 1:
Display list of products from database in MVC application.

Level 2:
Implement full CRUD operations using EF Core and MVC scaffolding.

Week 11: Lab Sheet–11 (Razor Views & Layouts)

Level 1:
Create Razor Views with layouts and partial views.

Level 2:
Develop a dynamic dashboard using layouts and reusable components.

Module 4 (Lab Sheets 12–15)

Week 12: Lab Sheet–12 (Validation & View Models)

Level 1:
Apply data annotation validations in a form (Required, Range, Email).

Level 2:
Create a ViewModel for complex form and implement custom validation logic.

Week 13: Lab Sheet–13 (Authentication & Authorization)

Level 1:
Implement login/logout using ASP.NET Core Identity.

Level 2:
Implement role-based access (Admin/User) and restrict pages.

Week 14: Lab Sheet–14 (AJAX & API Integration)

Level 1:
Use AJAX to fetch and display data without page reload.

Level 2:
Implement live search feature using AJAX and partial views.

Week 15: Lab Sheet–15 (Testing & Deployment)

Level 1:
Write unit tests for controller using MSTest/NUnit.

Level 2:
Deploy a full-stack application (local IIS/Azure) with testing integration.

Targeted Application & Tools that can be used:

1. **Targeted Application Area:** Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers.
2. **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:

1. **Problem Solving:** Design of Algorithms and implementation of programs.
2. **Programming:** Implementation of given scenarios using .NET.

Text Book:

1. **T1**. Fender, Young, "Front-end Fundamentals", Leanpub, 2015
2. **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:

1. **R1.** Benjamin Perkins, Jon D. Reid, "Beginning C# and .NET", Wiley, 2021.
2. **R2.** Piotr Gankiewicz, "Full Stack .NET Web Development", Packt Publishing, 2017.
3. **R3.** Tamir Dresher, Amir Zuker, Shay Friedman, "Hands-On Full-Stack Web Development with ASP.NET Core", Packt Publishing, 2018.
4. **R4.** Dustin Metzgar, "Exploring .NET core with microservices, ASP.NET core, and Entity Framework Core", Manning, 2017.

Web References

1. Microsoft .NET Documentation: https://docs.microsoft.com/en-us/dotnet/
2. ASP.NET Core Documentation: https://docs.microsoft.com/en-us/aspnet/core/
3. Entity Framework Core Documentation: https://docs.microsoft.com/en-us/ef/core/
4. C# Documentation: https://docs.microsoft.com/en-us/dotnet/csharp/
5. Tutorials for C#, ASP.NET Core, EF Core: https://www.tutorialspoint.com/
6. Microsoft Learn (Interactive learning paths): https://learn.microsoft.com/en-us/

Course Code: **CSE3450**	**Course Title: Introduction to NVIDIA Accelerated Computing and AI** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces the fundamentals of Artificial Intelligence and NVIDIA accelerated computing, focusing on GPU architecture, parallel processing, and the NVIDIA software ecosystem. It provides hands-on experience in GPU-accelerated data science, deep learning, and Generative AI using Python and NVIDIA tools, with					

	attention to ethical and real-world applications.
Course Objective	**The primary objective of this course is:** To introduce students to the fundamentals of Artificial Intelligence and accelerated computing To familiarize students with GPU architecture and NVIDIA ecosystem To provide hands-on exposure to GPU-accelerated data science and deep learning To develop problem-solving skills using NVIDIA tools and ethical AI practices
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1.** Explain the fundamentals of Artificial Intelligence and NVIDIA accelerated computing concepts. [Understand] **CO2.** Apply Python programming and GPU-accelerated libraries for data processing and analysis tasks. [Apply] **CO3.** Implement basic deep learning models using GPU-based frameworks for improved computational performance. [Apply] **CO4.** Analyze the performance benefits of GPU acceleration in AI and data science workflows. [Analyze] **CO5.** Design simple AI and Generative AI solutions while considering ethical, societal, and responsible AI aspects. [Create]

Course Content:

Module 1	**Foundations of AI and NVIDIA Ecosystem**	**8 Sessions**
Introduction to Artificial Intelligence: AI, Machine Learning, and Deep Learning Applications of AI in healthcare, finance, cybersecurity, smart cities, Introduction to GPUs and accelerated computing, CPU vs GPU architecture and parallel processing, Overview of NVIDIA ecosystem: GPUs, CUDA, Tensor Cores, Introduction to NVIDIA Deep Learning Institute (DLI) Hands-on: Exploring GPU environment and executing basic GPU programs.		
Module 2	**Python and Data Science with GPU Acceleration**	**7 Sessions**
Python programming fundamentals for AI, Python libraries: NumPy, Pandas, Matplotlib Introduction to Data Science workflow, Data preprocessing and exploratory data analysis, Introduction to RAPIDS for GPU-accelerated data processing, Performance comparison: CPU vs GPU computing, Hands-on: Data analysis using GPU-accelerated libraries.		
Module 3	**Fundamentals of Deep Learning using NVIDIA Tools**	**8 Sessions**
Basics of neural networks and deep learning, Activation functions, loss functions, and optimization techniques, Introduction to deep learning frameworks: PyTorch and TensorFlow, Role of CUDA and cuDNN in deep learning, training deep learning models on GPUs, Model evaluation and performance metrics, Hands-on: Training a basic neural network on GPU		
Module 4	**Generative AI and Real-World Applications**	**7 Sessions**

Introduction to Generative AI and foundation models, Basics of Large Language Models (LLMs), Image and text generation concepts, Introduction to NVIDIA NeMo, Retrieval Augmented Generation (RAG) concepts, Ethical, legal, and social issues in AI, Mini Project / Case Study: Design and demonstrate a basic AI or Generative AI application

List of Laboratory Tasks: **30 Sessions**

Module 1 (Lab Sheets 1–3)

Week 1: Lab Sheet–1 (GPU Setup & NVIDIA Environment)

Level 1:
Install NVIDIA drivers, CUDA Toolkit, and verify GPU setup using nvidia-smi. Execute a simple Python "Hello GPU" program.

Level 2:
Write and execute a basic CUDA program (vector addition) and compare execution with CPU implementation.

Week 2: Lab Sheet–2 (CPU vs GPU Parallel Computing)

Level 1:
Implement matrix multiplication using NumPy (CPU) and measure execution time.

Level 2:
Implement the same matrix multiplication using CuPy (GPU) and compare speedup and performance metrics.

Week 3: Lab Sheet–3 (NVIDIA DLI & GPU Notebook Exploration)

Level 1:
Access NVIDIA DLI or Google Colab GPU environment and execute sample notebooks.

Level 2:
Modify an existing notebook to run a parallel computation task and analyze GPU utilization.

Module 2 (Lab Sheets 4–7)

Week 4: Lab Sheet–4 (Data Preprocessing using RAPIDS cuDF)

Level 1:
Load a dataset using Pandas and perform cleaning (missing values, duplicates).

Level 2:
Perform the same operations using RAPIDS cuDF and compare execution time.

Week 5: Lab Sheet–5 (Exploratory Data Analysis - GPU)

Level 1:
Perform EDA using Matplotlib/Seaborn on a dataset.

Level 2:
Use GPU-based libraries (cuDF/cuXfilter) to visualize large datasets interactively.

Week 6: Lab Sheet–6 (Regression using GPU – cuML)

Level 1:
Implement Linear Regression using scikit-learn on CPU.

Level 2:

Implement Linear and Logistic Regression using RAPIDS cuML and compare accuracy and speed.

Week 7: Lab Sheet–7 (Clustering using GPU – cuML)

Level 1:
Perform K-Means clustering using scikit-learn.

Level 2:
Perform K-Means clustering using cuML on GPU and analyze clustering efficiency.

Module 3 (Lab Sheets 8–11)

Week 8: Lab Sheet–8 (Neural Network Basics – MLP)

Level 1:
Build a simple MLP using TensorFlow/Keras for MNIST classification (CPU).

Level 2:
Train the same model on GPU and compare training time and accuracy.

Week 9: Lab Sheet–9 (Activation Functions & Optimizers)

Level 1:
Train a neural network using different activation functions (ReLU, Sigmoid).

Level 2:
Compare optimizers (SGD vs Adam) and analyze convergence and loss graphs.

Week 10: Lab Sheet–10 (CNN for Image Classification)

Level 1:
Build a CNN model using TensorFlow/PyTorch for image classification.

Level 2:
Train the CNN on GPU and evaluate performance improvements.

Week 11: Lab Sheet–11 (Transfer Learning)

Level 1:
Use a pre-trained model (e.g., VGG16/ResNet) for feature extraction.

Level 2:
Fine-tune the model on a custom dataset and compare results.

Module 4 (Lab Sheets 12–15)

Week 12: Lab Sheet–12 (Object Detection – YOLO)

Level 1:
Run a pre-trained YOLO model for object detection.

Level 2:
Customize the model for a specific dataset and evaluate detection accuracy.

Week 13: Lab Sheet–13 (RAG Pipeline for NLP)

Level 1:
Implement a simple document retrieval system.

Level 2:
Build a Retrieval Augmented Generation (RAG) pipeline for Q&A using local documents.

Week 14: Lab Sheet–14 (Generative AI with NVIDIA NeMo)

Level 1:
Generate text using a pre-trained language model.

Level 2:
Fine-tune a generative model for domain-specific text generation.

Week 15: Lab Sheet–15 (AI Ethics & Model Evaluation)

Level 1:
Evaluate a trained model using accuracy, precision, recall.

Level 2:
Perform an AI ethics audit identifying bias, fairness, and security concerns in the model.

Targeted Application & Tools that can be used:

Tools: Python, TensorFlow, PyTorch, CUDA Toolkit, cuDNN, NVIDIA RAPIDS, TensorRT, Jupyter Notebook / Google Colab, NVIDIA NGC (GPU-optimized containers)

Applications such as

1. Image Classification and Object Detection
2. Natural Language Processing (Text Classification, Chatbots)
3. Generative AI (Text/Image Generation)
4. Recommendation Systems
5. Healthcare Analytics (medical imaging, diagnosis support)

Text Book(s):

1. T1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press
2. T2. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras& TensorFlow*, O'Reilly

Reference(s):

1. R1. NVIDIA Deep Learning Institute (DLI)
2. R2. NVIDIA Developer Documentation
3. R3. PyTorch and TensorFlow official documentation

Course Code: **CSE3451**	**Course Title: NVIDIA Applied Artificial Intelligence and Generative AI** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course focuses on applied Artificial Intelligence and Generative AI using NVIDIA GPU platforms, enabling students to design, optimize, and deploy AI systems with accelerated computing. It emphasizes hands-on development of GPU-accelerated machine learning, deep learning, and LLM-based applications, while addressing responsible, scalable, and sustainable AI practices.					

Course Objective	**The primary objective of this course is:** To strengthen applied AI skills using NVIDIA platforms To build efficient AI pipelines using GPU acceleration To understand Generative AI, LLMs, and deployment workflows To introduce responsible AI and real-world AI system design	
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1.** Analyze different AI workloads and select appropriate NVIDIA GPU-based solutions. [Analyze] **CO2.** Build and optimize machine learning pipelines using GPU acceleration techniques. [Apply] **CO3.** Train and fine-tune deep learning and Generative AI models for various applications. [Apply] **CO4.** Evaluate the performance and scalability of AI models deployed on GPU-based systems. [Evaluate] **CO5.** Design and deploy AI applications while considering ethical, scalability, and sustainability aspects. [Create]	
Course Content:		
Module 1	**NVIDIA AI Platforms and Workflows**	**8 Sessions**
Review of AI system architecture, AI workloads: training vs inference, NVIDIA AI software **stack overview:** CUDA (conceptual), cuDNN, TensorRT (overview), NVIDIA AI Enterprise ecosystem, Cloud vs On-prem GPU AI workflows, **Hands-on:** Exploring NVIDIA GPU environment & AI workflow		
Module 2	**GPU-Accelerated Machine Learning & Data Pipelines**	**7 Sessions**
End-to-end ML pipeline design, Data preprocessing and feature engineering on GPU, RAPIDS ecosystem: cuDF for dataframes, cuML for ML algorithms, Model evaluation and optimization, Performance comparison: CPU vs GPU ML pipelines, **Hands-on:** Build and optimize an ML pipeline using RAPIDS		
Module 3	**Deep Learning and Model Optimization**	**8 Sessions**
Review of deep learning architectures (CNNs, RNNs, Transformers), Training deep learning models on GPU, Transfer learning and fine-tuning, Mixed precision training and performance tuning, Model evaluation, overfitting, and generalization, **Hands-on:** Fine-tune a deep learning model on GPU		
Module 4	**Generative AI, Deployment, and Responsible AI**	**7 Sessions**
Fundamentals of Generative AI, Large Language Models (LLMs): concepts and use cases, NVIDIA NeMo framework overview, Retrieval Augmented Generation (RAG) pipeline, Model deployment basics: Inference optimization, NVIDIA Triton (conceptual), Responsible AI: Bias, fairness, privacy, Sustainability and green AI, **Mini Project:** Design and deploy a simple AI / GenAI application		

List of Laboratory Tasks: **30 Sessions**

Module 1 (Lab Sheets 1–3)

Week 1: Lab Sheet–1 (GPU Profiling & NVIDIA Environment)

Level 1:
Use nvidia-smi to identify GPU specifications such as memory, utilization, and temperature.

Level 2:
Use NVML (NVIDIA Management Library) to programmatically extract GPU metrics and analyze compute capability vs performance.

Week 2: Lab Sheet–2 (Training vs Inference Performance Analysis)

Level 1:
Train a simple neural network and measure training time on GPU.

Level 2:
Compare training vs inference latency and throughput using profiling tools.

Week 3: Lab Sheet–3 (AI Workflow & GPU Pipeline Exploration)

Level 1:
Execute a basic AI pipeline (data → model → prediction) on GPU.

Level 2:
Optimize workflow by reducing data transfer overhead between CPU and GPU.

Module 2 (Lab Sheets 4–7)

Week 4: Lab Sheet–4 (Data Processing with RAPIDS cuDF)

Level 1:
Load and preprocess dataset using Pandas (CPU).

Level 2:
Perform the same preprocessing using cuDF and benchmark execution time.

Week 5: Lab Sheet–5 (GPU-Accelerated ML Model – cuML)

Level 1:
Build a classification model using scikit-learn.

Level 2:
Implement the same model using cuML and compare performance metrics.

Week 6: Lab Sheet–6 (Hyperparameter Tuning on GPU)

Level 1:
Train a model with basic parameters.

Level 2:
Use GPU-based hyperparameter tuning to optimize model performance and measure speedup.

Week 7: Lab Sheet–7 (GPU-based ETL Pipeline Design)

Level 1:
Design a basic ETL pipeline using Python (Extract, Transform, Load).

Level 2:
Implement a GPU-resident ETL pipeline minimizing host-device memory transfer.

Module 3 (Lab Sheets 8–11)

Week 8: Lab Sheet–8 (Transfer Learning using Pretrained Models)

Level 1:
Use a pre-trained CNN (ResNet/VGG) for feature extraction.

Level 2:
Fine-tune the model on a domain-specific dataset.

Week 9: Lab Sheet–9 (Mixed Precision Training - AMP)

Level 1:
Train a model using FP32 precision.

Level 2:
Apply Automatic Mixed Precision (AMP) and compare training speed and accuracy.

Week 10: Lab Sheet–10 (Model Optimization Techniques)

Level 1:
Apply regularization (Dropout/L2) to improve model performance.

Level 2:
Compare optimization strategies using TensorBoard visualization.

Week 11: Lab Sheet–11 (Model Compression – Pruning & Quantization)

Level 1:
Apply model pruning to reduce network size.

Level 2:
Apply quantization and evaluate impact on inference speed and accuracy.

Module 4 (Lab Sheets 12–15)

Week 12: Lab Sheet–12 (LLM Deployment using NVIDIA NeMo)

Level 1:
Run a pre-trained language model for text generation.

Level 2:
Evaluate zero-shot and few-shot learning using prompt engineering.

Week 13: Lab Sheet–13 (RAG Pipeline Implementation)

Level 1:
Implement a document retrieval system using embeddings.

Level 2:
Integrate with LLM to build a Retrieval Augmented Generation (RAG) pipeline.

Week 14: Lab Sheet–14 (Model Deployment & Optimization – TensorRT& Triton)

Level 1:
Deploy a trained model for inference.

Level 2:
Optimize inference using TensorRT and simulate multi-model deployment using Triton.

Week 15: Lab Sheet–15 (Responsible AI & Final Application)

Level 1:

Evaluate model fairness using performance metrics.

Level 2:
Design and deploy a mini AI/GenAI application incorporating ethical and sustainability considerations.

Targeted Application & Tools that can be used:

Tools: Python, TensorFlow, PyTorch, CUDA Toolkit, cuDNN, NVIDIA RAPIDS, TensorRT, Jupyter Notebook / Google Colab, NVIDIA NGC (GPU-optimized containers)

Applications such as

1. Image Classification and Object Detection
2. Natural Language Processing (Text Classification, Chatbots)
3. Generative AI (Text/Image Generation)
4. Recommendation Systems
5. Healthcare Analytics (medical imaging, diagnosis support)

Text Book(s):

1. **T1. Ian Goodfellow, Yoshua Bengio, Aaron Courville**, *Deep Learning*, MIT Press – Core reference for deep learning concepts and architectures
2. **T2. Aurélien Géron**, *Hands-On Machine Learning with Scikit-Learn, Keras& TensorFlow*, O'Reilly – Practical ML & DL workflows with model tuning and evaluation
3. **T3. Kirk & Hwu**, *Programming Massively Parallel Processors*, Morgan Kaufmann – Conceptual understanding of GPU and parallel computing
4. **T4. Francois Chollet**, *Deep Learning with Python*, Manning Publications – Model design, transfer learning, and optimization
5. **T5. Chip Huyen**, *Designing Machine Learning Systems*, O'Reilly – Industry-oriented ML system design and deployment concepts

Reference(s):

R1. NVIDIA Deep Learning Institute (DLI)

1. GPU-Accelerated ML & DL
2. Generative AI Explained
3. Building RAG Applications
4. Model Optimization & Deployment

R2. NVIDIA Developer Documentation

1. NVIDIA AI software stack
2. TensorRT, Triton Inference Server (overview)
3. RAPIDS documentation

Course Code: **CSE3452**	**Course Title: Advanced AI Systems and Generative Intelligence using NVIDIA Platforms** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This advanced course explores the design, optimization, and deployment of large-scale AI and Generative Intelligence systems using NVIDIA platforms and software stacks. It combines theory, hands-on labs, and a capstone project to develop high-performance, ethical, and sustainable AI solutions for real-world, production-grade environments.					
Course Objective	**The primary objective of this course is:** To design high-performance AI systems using NVIDIA platforms To optimize deep learning and Generative AI models for scale and deployment To build end-to-end AI pipelines including training, inference, and monitoring To evaluate AI systems with respect to ethics, sustainability, and real-world constraints					
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1.** Architect and evaluate large-scale AI systems using the NVIDIA AI software stack. [Analyze] **CO2.** Optimize deep learning and large language models (LLMs) for performance, memory efficiency, and inference speed. [Apply] **CO3.** Design and deploy Generative AI systems using advanced NVIDIA frameworks and tools. [Create] **CO4.** Analyze system-level challenges in scaling AI applications, including resource management and distributed computing. (Analyze) **CO5.** Evaluate ethical, sustainable, and responsible AI practices in real-world deployments. [Evaluate]					
Course Content:						
Module 1	**Advanced NVIDIA AI Architecture & Systems**					**8 Sessions**
NVIDIA AI system architecture (training, inference, orchestration), Multi-GPU and distributed AI concepts, NVIDIA AI software stack deep dive: CUDA (advanced concepts – streams, concurrency), cuDNN, NCCL, TensorRT, AI workloads on Cloud vs On-Prem vs Edge, **Hands-on:** Profiling AI workloads using NVIDIA tools.						
Module 2	**Model Optimization and High-Performance AI**					**7 Sessions**
Performance bottlenecks in deep learning systems, Mixed precision & quantization techniques, Model pruning, distillation, and compression, Efficient data pipelines for large-scale training, Benchmarking and optimization strategies, **Hands-on:** Optimizing DL models using TensorRT / AMP						
Module 3	**Advanced Generative AI & LLM Systems**					**8 Sessions**
Foundation models and LLM architectures, Fine-tuning strategies (LoRA, PEFT, adapters), NVIDIA NeMo deep dive, Retrieval Augmented Generation (RAG) at scale, Multimodal Generative AI (text, image, video),						

Hands-on: Fine-tuning and serving LLMs on GPU.

Module 4	**AI Deployment, MLOps, and Responsible AI**	**7 Sessions**

End-to-end AI deployment pipeline, NVIDIA Triton Inference Server, Model serving, monitoring, and versioning, AI security, privacy, and adversarial threats, Green AI and sustainable computing, Ethical and legal considerations in AI systems, **Capstone Project:** Design and deploy an advanced AI / GenAI system.

List of Laboratory Tasks: **30 Sessions**

Module 1 (Lab Sheets 1–4)

Week 1: Lab Sheet–1 (GPU Profiling using NVIDIA Nsight Systems)

Level 1:
Run a deep learning training script (e.g., ResNet-50) and monitor GPU utilization using nvidia-smi.

Level 2:
Profile the same script using NVIDIA Nsight Systems to identify bottlenecks such as memory stalls and compute inefficiencies.

Week 2: Lab Sheet–2 (CUDA Streams & Parallel Execution)

Level 1:
Implement a sequential program performing data transfer and computation.

Level 2:
Use CUDA Streams to overlap computation with data transfer and compare execution time with the sequential version.

Week 3: Lab Sheet–3 (Multi-GPU Training with NCCL)

Level 1:
Train a simple neural network using a single GPU.

Level 2:
Implement distributed training using NCCL backend and measure scaling efficiency across multiple GPUs.

Week 4: Lab Sheet–4 (Cloud vs Edge Deployment)

Level 1:
Deploy a trained model on a cloud GPU environment.

Level 2:
Deploy the same model on an edge device (e.g., Jetson) and compare latency, memory usage, and power consumption.

Module 2 (Lab Sheets 5–8)

Week 5: Lab Sheet–5 (Mixed Precision Training – AMP)

Level 1:
Train a model using FP32 precision.

Level 2:
Apply Automatic Mixed Precision (AMP) and compare VRAM usage and training speed.

Week 6: Lab Sheet–6 (TensorRT Optimization)

Level 1:
Convert a trained model into an inference-ready format.

Level 2:
Optimize the model using TensorRT INT8 quantization and compare inference throughput.

Week 7: Lab Sheet–7 (Knowledge Distillation)

Level 1:
Train a standard deep learning model.

Level 2:
Implement knowledge distillation where a teacher model trains a smaller student model and evaluate performance trade-offs.

Week 8: Lab Sheet–8 (High-Performance Data Pipeline – DALI)

Level 1:
Load data using standard CPU-based data loaders.

Level 2:
Implement NVIDIA DALI pipeline and compare CPU utilization and GPU efficiency.

Module 3 (Lab Sheets 9–12)

Week 9: Lab Sheet–9 (Fine-tuning LLMs using LoRA)

Level 1:
Use a pre-trained LLM for inference tasks.

Level 2:
Apply LoRA-based fine-tuning using NVIDIA NeMo for a domain-specific dataset.

Week 10: Lab Sheet–10 (RAG Pipeline at Scale)

Level 1:
Implement a basic document retrieval system using embeddings.

Level 2:
Develop a scalable Retrieval Augmented Generation (RAG) pipeline using a vector database.

Week 11: Lab Sheet–11 (Multimodal Generative AI System)

Level 1:
Implement a text-to-image or image-to-text model.

Level 2:
Integrate a Vision-Language model with a generative model to create a multimodal AI system.

Week 12: Lab Sheet–12 (TensorRT-LLM Optimization)

Level 1:
Run inference using a standard LLM.

Level 2:
Optimize LLM using TensorRT-LLM with batching and KV-caching for high throughput.

Module 4 (Lab Sheets 13–15)

Week 13: Lab Sheet–13 (Triton Inference Server Deployment)

Level 1:
Deploy a single model using Triton Inference Server.

Level 2:
Deploy multiple models with dynamic batching and evaluate performance under concurrent requests.

Week 14: Lab Sheet–14 (AI Security & Guardrails)

Level 1:
Test a model using standard input queries.

Level 2:
Perform adversarial prompt testing and implement safety guardrails using NVIDIA NeMo Guardrails.

Week 15: Lab Sheet–15 (Green AI & Sustainability Dashboard)

Level 1:
Monitor GPU usage and energy consumption during model training.

Level 2:
Design a sustainability dashboard to estimate carbon footprint and propose optimization strategies.

Targeted Application & Tools that can be used:

Tools: Python, TensorFlow, PyTorch, CUDA Toolkit, cuDNN, NVIDIA RAPIDS, TensorRT, Jupyter Notebook / Google Colab, NVIDIA NGC (GPU-optimized containers)

Applications such as

Image Classification and Object Detection

Natural Language Processing (Text Classification, Chatbots)

Generative AI (Text/Image Generation)

Recommendation Systems

Healthcare Analytics (medical imaging, diagnosis support)

Text Book(s):

1. T1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press – Advanced deep learning theory and architectures
2. T2. Chip Huyen, *Designing Machine Learning Systems*, O'Reilly – End-to-end ML/AI system design, deployment, and monitoring
3. T3. Kirk & Hwu, *Programming Massively Parallel Processors*, Morgan Kaufmann – GPU computing and parallel processing fundamentals
4. T4. Francois Chollet, *Deep Learning with Python*, Manning – Advanced model design, optimization, and fine-tuning
5. T5. Boris Ginsburg et al., *Deep Learning Performance Optimization*, NVIDIA Whitepapers – Performance tuning and optimization strategies

Reference(s):

R1. NVIDIA Deep Learning Institute (DLI) – Advanced

1. Advanced GPU-Accelerated Deep Learning
2. Building and Optimizing LLMs with NVIDIA NeMo
3. Efficient AI Inference with TensorRT

4. Deploying AI Models with NVIDIA Triton Inference Server

R2. NVIDIA Developer Resources

1. NVIDIA AI Software Stack Documentation
2. CUDA Advanced Programming Guide
3. TensorRT and Triton official documentation
4. NVIDIA NeMo framework documentation

Course Code: **CSE3500**	**Course Title: Intelligent Systems with Machine Learning** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2267 - Machine Learning Techniques					
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	To develop a strong foundation in intelligent systems by enabling students to understand machine learning concepts, apply supervised and unsupervised learning techniques, design solutions for real-world problems, and evaluate model performance and optimization strategies.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Apply advanced supervised machine learning techniques for predictive modeling. [Apply] **CO2:** Develop machine learning models with improved predictive performance using meta-learning (ensemble) algorithms. [Apply] **CO3:** Build predictive models using perceptron learning algorithms. [Apply] **CO4:** Apply advanced unsupervised learning algorithms for clustering, competitive learning, and outlier detection. [Apply] **CO5:** Implement intelligent machine learning models using Python libraries. [Apply]					

Course Content:

Module 1	**Supervised Learning**	**8 Sessions**

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**	**7 Sessions**

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**	**8 Sessions**

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**	**7 Sessions**

Topics:

Unsupervised Learning – simple k Means clustering- simple and mini-batch**;** updating centroids incrementally; finding the optimal number of clusters using Elbow method ; Silhouttecoefficient,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: **30 Sessions**

Module: 1 (Labsheets 1 to 6)

Week 1:

Labsheet-1 (Data Preprocessing & Missing Values)

Level 1: Implement different missing value handling techniques (mean, median, mode, forward fill) using Scikit-learn.

Level 2: Develop a custom Python function to handle missing values and compare results with library methods.

Week 2:

Labsheet-2 (Data Visualization & EDA)

Level 1: Perform exploratory data analysis using scatter plot, pair plot, and count plot.

Level 2: Generate heatmaps and word clouds and interpret feature relationships.

Week 3:

Labsheet-3 (Linear & Polynomial Regression)

Level 1: Implement simple linear regression and evaluate performance metrics (MSE, R^2).

Level 2: Implement polynomial regression and compare learning curves with linear regression.

Week 4:

Labsheet-4 (Logistic Regression)

Level 1: Plot sigmoid function using custom implementation.

Level 2: Implement logistic regression, estimate probabilities, and visualize decision boundaries.

Week 5:

Labsheet-5 (Bayesian Learning - Naïve Bayes)

Level 1: Implement Naïve Bayes classifier on a dataset.

Level 2: Compare performance for categorical vs continuous features and analyze accuracy.

Week 6:

Labsheet-6 (Support Vector Machines)

Level 1: Implement linear SVM for classification.

Level 2: Implement non-linear SVM using kernel trick and compare results.

Module: 2 (Labsheets 7 to 8)

Week 7:

Labsheet-7 (Bagging & Random Forest)

Level 1: Implement Bagging and evaluate using Out-of-Bag score.

Level 2: Implement Random Forest and analyze feature importance.

Week 8:

Labsheet-8 (Boosting Techniques)

Level 1: Implement AdaBoost and Gradient Boosting.

Level 2: Implement Stacking and compare ensemble models performance.

Module: 3 (Labsheets 9 to 10)

Week 9:

Labsheet-9 (Perceptron Classifier)

Level 1: Implement Perceptron classifier for binary classification.

Level 2: Analyze convergence and decision boundaries.

Week 10:

Labsheet-10 (Multilayer Perceptron & Image Classification)

Level 1: Implement MLP using Keras for classification.

Level 2: Develop an image classifier using neural networks and evaluate accuracy.

Module: 4 (Labsheets 11 to 15)

Week 11:

Labsheet-11 (K-Means Clustering)

Level 1: Implement K-Means clustering and compute inertia.

Level 2: Determine optimal clusters using Elbow and Silhouette methods.

Week 12:

Labsheet-12 (K-Means Applications)

Level 1: Apply K-Means for clustering dataset.

Level 2: Perform image segmentation using K-Means++.

Week 13:

Labsheet-13 (DBSCAN Clustering)

Level 1: Implement DBSCAN clustering algorithm.

Level 2: Compare DBSCAN with K-Means for irregular clusters.

Week 14:

Labsheet-14 (Gaussian Mixture Models)

Level 1: Implement GMM using EM algorithm.

Level 2: Compare clustering results with K-Means.

Week 15:

Labsheet-15 (Outlier Detection)

Level 1: Implement Isolation Forest and Local Outlier Factor (LOF).

Level 2: Analyze detected anomalies and compare both techniques.

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 availableat :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 :

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: Theory Course**	**L- T-P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2267 - Machine Learning Techniques					
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. Topics include Fundamental concepts of deep neural networks, Convolutional Neural Networks, Recurrent Network structures, Deep Unsupervised Learning, Generative Adversarial Networks and applications in various problemdomains.					
Course Objective	To develop advanced knowledge in deep learning by enabling students to understand modern architectures and models, apply techniques such as CNNs, RNNs, and transformer-based models, optimize model performance and analyze real-world applications of deep learning.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain advanced deep learning concepts, architectures, optimization techniques, and model evaluation methods. [Understand] **CO2:** Apply suitable deep learning algorithms to solve complex problems across various domains. [Apply] **CO3:** Implement deep generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). [Apply] **CO4:** Apply deep learning architectures for processing image, audio, and sequential data. [Apply] **CO5:** Analyze and evaluate the performance of deep learning models and optimize them for real-world applications. [Analyze]					

Course Content:

Module 1	**Introduction to Deep Learning and Neural Networks**	**13 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**	**12 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**	**10 Sessions**

Topics:

Generative Adversarial Networks, Kohonen Networks, Autoencoders, Boltzmann Machine, Restricted Boltzmann Machine, Deep Belief Network

Module-4	**Advanced Deep Learning Architectures**	**10 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)

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:

1. Swayam Nptel – Deep Learning – IIT Ropar https://onlinecourses.nptel.ac.in/noc21_cs35/preview
2. Coursera – Neural Networks and Deep Learning Andrew Ng
3. Coursera - Neural Networks for Machine Learning by Geoffrey Hinton in Coursera

Course Code: **CSE3502**	**Course Title: Computational Optimization for Intelligent Systems** **Type of Course: Theory Course**	**L-T- P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2267 - Machine Learning Techniques					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a strong foundation in optimization techniques for intelligent systems by enabling students to understand various optimization methods, apply linear, nonlinear, and heuristic approaches, solve complex computational problems, and evaluate the performance of optimization algorithms.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain fundamental concepts of machine learning and their applications in real-world scenarios. [Understand] **CO2:** Implement basic machine learning models such as decision trees, linear regression, and neural networks using appropriate tools or programming languages. [Apply] **CO3:** Explain the principles of convex optimization and assess their suitability for solving problems in domains such as portfolio optimization, machine learning, and network design. [Understand] **CO4:** Formulate and solve convex optimization problems using real or simulated data for applications such as cost minimization and resource allocation. [Apply]					

	CO5: Analyze and evaluate optimization results to enhance model performance and support decision-making in intelligent systems. [Analyze]	
Course Content:		
Module 1:	**Fundamentals of Machine learning**	**10 Sessions**
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).**		
Module 2:	**Machine learning models**	**12 Sessions**
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.		
Module 3:	**Convex optimization models**	**13 Sessions**
Topics: Linear Optimization (Linear Programming), Convex Quadratic Optimization, Second Order Cone Optimization (SOCP), Semi-definite Optimization (SDP) - basic concepts, Convex Composite Optimization.		
Module 4:	**Methods for convex optimization**	**10 Sessions**
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).		
Targeted Application & Tools that can be used: Use of Google Colab (Python with libraries like NumPy, SciPy, scikit-learn, TensorFlow/PyTorch, CVXPY/Pyomo).		
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: 1. T1. Charu C. Aggarwal, "*Linear Algebra and Optimization for Machine Learning*", Springer, 2020. 2. T2. Sra Suvrit, Nowozin Sebastian, and Wright Stephen J, "*Optimization for Machine Learning*",The MIT Press,2012.		
References: 1. R1.Guanghui Lan, "*First-order and Stochastic Optimization Methods for Machine Learning*", Springer Cham, 2020. **Web References:** 1. W1. https://sm-nitk.vlabs.ac.in/ 2. 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: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2267 - Machine Learning Techniques					
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	To develop a strong foundation in reinforcement learning by enabling students to understand core principles and frameworks, apply value-based and policy-based learning methods, design intelligent agents for decision-making tasks, and evaluate learning performance in dynamic environments.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain fundamental and advanced concepts of reinforcement learning, including key algorithms and their mathematical foundations. [Understand] **CO2:** Formulate real-world problems as reinforcement learning tasks by selecting appropriate models and techniques. [Apply] **CO3:** Apply reinforcement learning algorithms to solve decision-making and control problems across various domains. [Apply] **CO4:** Analyze the strengths and limitations of different reinforcement learning approaches in diverse applications. [Analyze] **CO5:** Design and implement reinforcement learning experiments, evaluate performance, and interpret results to derive meaningful conclusions. [Create]					

Course Content:

Module 1	**Introduction and Foundations**	**8 Sessions**

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)**	**7 Sessions**

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**	**8 Sessions**

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**	**7 Sessions**

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.

List of Laboratory Tasks: **30 Sessions**

Module: 1 (Labsheets 1 to 3)

Week 1:

Labsheet-1 (RL Environment Setup & Exploration)

Level 1: Install required libraries (Gym, NumPy, Matplotlib) and explore environments like FrozenLake and CartPole.

Level 2: Implement a random agent and compute average reward over multiple episodes.

Week 2:

Labsheet-2 (Probability & Random Variables in RL)

Level 1: Simulate discrete and continuous random variables and compute mean and variance.

Level 2: Implement joint and conditional probability distributions and analyze independence.

Week 3:

Labsheet-3 (Markov Chains & MRP)

Level 1: Implement a simple Markov Chain and compute state transitions.

Level 2: Extend to Markov Reward Process and compute value function iteratively.

Module: 2 (Labsheets 4 to 6)

Week 4:

Labsheet-4 (MDP Modeling& Value Functions)

Level 1: Define states, actions, and rewards for a gridworld problem.

Level 2: Compute state-value and action-value functions.

Week 5:

Labsheet-5 (Policy Evaluation)

Level 1: Implement iterative policy evaluation for a fixed policy.

Level 2: Analyze convergence speed with different thresholds.

Week 6:

Labsheet-6 (Policy Iteration & Value Iteration)

Level 1: Implement policy iteration algorithm.

Level 2: Implement value iteration and compare optimal policies.

Module: 3 (Labsheets 7 to 10)

Week 7:

Labsheet-7 (Monte Carlo Prediction)

Level 1: Implement First-Visit Monte Carlo prediction.

Level 2: Implement Every-Visit MC and compare convergence.

Week 8:

Labsheet-8 (Monte Carlo Control)

Level 1: Implement MC control with exploring starts.

Level 2: Analyze learning curve and policy improvement.

Week 9:

Labsheet-9 (On-policy vs Off-policy Learning)

Level 1: Implement on-policy learning method.

Level 2: Implement importance sampling for off-policy learning.

Week 10:

Labsheet-10 (Monte Carlo Applications)

Level 1: Apply MC methods to environments like Blackjack.

Level 2: Evaluate performance and compare policies.

Module: 4 (Labsheets 11 to 15)

Week 11:

Labsheet-11 (TD(0) Learning)

Level 1: Implement TD(0) prediction.

Level 2: Compare TD vs Monte Carlo convergence.

Week 12:

Labsheet-12 (SARSA Algorithm)

Level 1: Implement SARSA for on-policy control.

Level 2: Analyze performance under different epsilon values.

Week 13:

Labsheet-13 (Q-Learning Algorithm)

Level 1: Implement Q-Learning algorithm.

Level 2: Compare SARSA vs Q-Learning performance.

Week 14:

Labsheet-14 (Policy Gradient Methods)

Level 1: Implement REINFORCE algorithm.

Level 2: Analyze variance and introduce baseline.

Week 15:

Labsheet-15 (Actor-Critic & Real-world RL)

Level 1: Implement a basic Actor-Critic model.

Level 2: Apply RL algorithm on real-world environment (Taxi/CliffWalking) and evaluate performance.

Targeted Application & Tools that can be used:

1. Reinforcement Learning as a framework for computational neuroscience to model decision-making processes.

2. Applications of RL in various industries: Resource Management, Traffic Light Control, Robotics, Web System Configuration, Finance, Healthcare, Gaming, Autonomous Systems, etc.

3. **Tools:** Python (Jupyter Notebook, Google Colaboratory, Spyder), Libraries: NumPy, OpenAI Gym, Stable Baselines3, Ray RLib, PyTorch / TensorFlow (for Deep RL).

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" 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.

Course Code: **CSE3504**	**Course Title: Computational Linguistics Natural Language Processing** **Type of Course: Theory Course**	**L- T-P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2267 - Machine Learning Techniques					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a strong foundation in Natural Language Processing by enabling students to understand linguistic concepts and NLP fundamentals, apply text processing and language modeling techniques, implement NLP algorithms for real-world applications, and analyze syntactic and semantic structures in language.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental concepts, theories, and challenges in Natural Language Processing. [Understand]					

	CO2: Apply text preprocessing, representation, and modeling techniques using programming languages and NLP libraries. [Apply] **CO3:** Utilize word embeddings and deep learning architectures to develop NLP applications. [Apply] **CO4:** Analyze NLP models and techniques for tasks such as machine translation, sentiment analysis, and question answering. [Analyze] **CO5:** Evaluate the performance of NLP models using appropriate metrics and enhance their effectiveness for real-world applications. [Evaluate]

Course Content:

Module 1	**Introduction to NLP and Linguistic Fundamentals**	**12 Sessions**

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).

Module 2	**Word and Text Representations & Traditional ML for NLP**	**12 Sessions**

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**	**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	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.

---

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

1. T1. Daniel Jurafsky, and James Martin. "Speech and Language Processing" (3rd edition draft, available online at https://web.stanford.edu/~jurafsky/slp3/)
2. T2. Christopher D. Manning and Hinrich Schütze, "Foundations of Statistical Natural Language Processing", 1st Edition, MIT Press, 1999.

References:

1. R1. Yoav Goldberg, "Neural Network Methods in Natural Language Processing", Morgan & Claypool Publishers, 2017. (Excellent for neural NLP focus)
2. 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: **CSE3505**	**Course Title: Synergistic Neural Fuzzy Computing** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2267 - Machine Learning Techniques					
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	To develop a comprehensive understanding of hybrid intelligent systems by enabling students to understand neural networks and fuzzy logic integration, apply hybrid approaches for problem solving, design adaptive systems using neuro-fuzzy techniques, and evaluate the performance of intelligent hybrid models.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental concepts, architectures, and learning principles of neural networks. [Understand] **CO2:** Explain common learning algorithms such as Perceptron, Least Mean Squares, and Backpropagation. [Understand] **CO3:** Explain the concepts of fuzzy sets, fuzzy operations, and fuzzy relations, and distinguish them from crisp set theory. [Understand] **CO4:** Apply fuzzy logic concepts, including fuzzification, inference, and defuzzification, to design fuzzy logic controllers. [Apply] **CO5:** Design and implement basic neural network models and fuzzy logic systems using appropriate tools and libraries. [Apply]					
Course Content:						
Module 1	**Introduction to Neural Networks**					**8 Sessions**
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**	**7 Sessions**

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**	**8 Sessions**

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**	**7 Sessions**

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).

List of Laboratory Tasks: **30 Sessions**

Module : 1 (Labsheets 1 to 4)

week1:

Labsheet-1 (McCulloch-Pitts Neuron Model)
Level 1: Implement a McCulloch-Pitts neuron for basic logical operations (AND, OR, NOT) and verify outputs for different input combinations.
Level 2: Extend the neuron model to simulate a threshold-based decision system and analyze how varying weights and threshold values affect output.

week2:

Labsheet-2 (Perceptron Learning Algorithm)
Level 1: Implement the Perceptron algorithm for linearly separable data and classify simple datasets.
Level 2: Visualize the decision boundary and analyze convergence behavior for different learning rates.

week3:

Labsheet-3 (LMS Algorithm)
Level 1: Implement the Least Mean Square (LMS) algorithm for a simple regression problem.
Level 2: Analyze the impact of learning rate and initial weights on convergence and error minimization.

week4:

Labsheet-4 (Neural Network Architectures)
Level 1: Compare feedforward and recurrent neural networks through simple examples.
Level 2: Implement a basic feedforward neural network and evaluate its performance on sample data.

Module : 2 (Labsheets 5 to 8)

week5:

Labsheet-5 (MLP for XOR Problem)
Level 1: Implement a multilayer perceptron (MLP) to solve the XOR problem.
Level 2: Analyze the role of hidden layers and activation functions in solving non-linear problems.

week6:

Labsheet-6 (Backpropagation Algorithm)
Level 1: Implement backpropagation for training a simple MLP.
Level 2: Evaluate performance using different activation functions (ReLU, Sigmoid, Tanh).

week7:

Labsheet-7 (RBF Networks)
Level 1: Implement a Radial Basis Function network for function approximation.
Level 2: Compare the performance of RBF network with MLP for the same dataset.

week8:

Labsheet-8 (Self Organizing Maps - SOM)
Level 1: Implement SOM for clustering a small dataset.
Level 2: Visualize clusters and interpret feature mapping for higher dimensional data.

Module : 3 (Labsheets 9 to 11)

week9:

Labsheet-9 (Fuzzy Sets & Membership Functions)
Level 1: Define fuzzy sets using triangular, trapezoidal, and Gaussian membership functions.
Level 2: Compare fuzzy and crisp representations for real-world variables.

week10:

Labsheet-10 (Fuzzy Operations)
Level 1: Implement fuzzy operations such as complement, union, and intersection.
Level 2: Analyze the effect of different operators (min, max, probabilistic sum).

week11:

Labsheet-11 (Fuzzy Relations)
Level 1: Define binary fuzzy relations and compute their compositions.
Level 2: Compare max-min and max-product composition methods and interpretresults.

Module : 4 (Labsheets 12 to 15)

week12:

Labsheet-12 (Fuzzification Process)
Level 1: Implement fuzzification for crisp inputs using predefined membership functions.
Level 2: Evaluate how different membership functions affect fuzzified outputs.

week13:

Labsheet-13 (Fuzzy Inference System – Mamdani)
Level 1: Design a rule base for a simple fuzzy system (e.g., temperature control).
Level 2: Implement Mamdani inference and visualize aggregated output.

week14:

Labsheet-14 (Defuzzification Methods)
Level 1: Implement centroid and mean of maxima methods.
Level 2: Compare outputs of different defuzzification methods for the same fuzzy set.

week15:

Labsheet-15 (Complete Fuzzy Logic Controller & Neuro-Fuzzy Concept)
Level 1: Build a complete fuzzy logic controller integrating all components.
Level 2: Analyze system performance and discuss how neuro-fuzzy models (ANFIS) improve adaptability.

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.

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: Theory Course**	**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	To develop a foundational understanding of bioinformatics by enabling students to explore biological data and concepts, apply sequence alignment and genome analysis					

	techniques, utilize computational tools in biological research, and analyze biological datasets.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental concepts of DNA and protein sequences and their structural organization. [Understand] **CO2:** Explain bioinformatics file formats and the principles of sequence alignment techniques. [Understand] **CO3:** Apply computational techniques for sequence similarity searching and motif discovery in DNA and protein sequences. [Apply] **CO4:** Analyze biological sequence data using alignment results, scoring matrices, and pattern identification methods. [Analyze] **CO5:** Evaluate and interpret bioinformatics results to derive biological insights and support decision-making in genomics and proteomics. [Evaluate]

Course Content:

Module 1	**Fundamentals of Bioinformatics**	**15Sessions**

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**	**15Sessions**

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**	**15Sessions**

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

1. https://puniversity.informaticsglobal.com:2229/login.aspx
2. NCBI Handbooks and Resources: https://www.ncbi.nlm.nih.gov/books/
3. ExPASy Bioinformatics Resource Portal: https://www.expasy.org/
4. Biopython Documentation: https://biopython.org/
5. 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:

    a. String Reconstruction Problem
    b. Sequence Similarity Searching (e.g., BLAST variants)
    c. Alignment Scores and Gap Penalties
    d. Protein Sequencing and its implications
    e. Gene Prediction Models, especially Hidden Markov Models (HMM)
    f. Finding similarities by performing Pairwise and Multiple Sequence Alignments
    g. Evaluating Phylogenetic Trees (implied as an application of sequence alignment results).

Course Code: CSE3507	**Course Title: Algorithms in Computational Biology** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE3506 – Introductionto Bioinformatics					
Anti-requisites	NIL					
Course 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.					
Course Objective	To develop a strong foundation in computational biology by enabling students to understand algorithmic techniques used in biological data analysis, apply sequence alignment and pattern matching methods, solve biological problems using computational approaches, and evaluate the efficiency of biological algorithms.					
Course OutComes	On successful completion of this course, students will be able to: **CO1:** Define key concepts in computational biology and fundamental algorithmic principles. [Remember] **CO2:** Explain the significance and challenges of applying algorithmic approaches to analyze biological data. [Understand] **CO3:** Apply algorithmic strategies, including dynamic programming, to solve biological problems such as sequence alignment and gene prediction. [Apply] **CO4:** Implement algorithms for processing various types of biological data while addressing computational requirements. [Apply] **CO5:** Analyze and evaluate the performance of computational biology algorithms in terms of efficiency, accuracy, and scalability. [Analyze]					
Course Content:						
Module 1	**Introduction**	**8 Sessions**				
Topics: 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.

Module 2	**Algorithms Issues and Problems**	**7 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

Module 3	**Algorithmic Approaches**	**8 Sessions**

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.

Module 4	**Dynamic Programming & Methods**	**7 Sessions**

Topics:

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)

List of Laboratory Tasks: **30 Sessions**

Module : 1 (Labsheets 1 to 4)

week1:

Labsheet-1(Sorting and Searching Algorithms in Biological Data)

Level 1: Implement Bubble Sort, Merge Sort, Linear Search, and Binary Search on a dataset representing gene lengths.

Level 2: Analyze and compare time complexity of sorting/searching algorithms using real biological datasets and visualize performance.

week2:

Labsheet-2 (Fibonacci and Recursion Analysis)

Level 1: Implement recursive and iterative Fibonacci algorithms.

Level 2: Compare recursive, iterative, and dynamic programming approaches in terms of time and space complexity using profiling tools.

week3:

Labsheet-3 (Algorithm Complexity & Big-O Analysis)

Level 1: Analyze time complexity of given algorithms using Big-O notation.

Level 2: Evaluate algorithm efficiency for biological datasets and justify selection of optimal

approaches.

week4:

Labsheet-4 (Basic Biological Data Processing)

Level 1: Parse and process biological sequence data (DNA/Protein) using Python.

Level 2: Implement basic preprocessing such as sequence cleaning and statistical summary.

Module : 2 (Labsheets 5 to 7)

week5:

Labsheet-5 (Travelling Salesman Problem in Genome Assembly)

Level 1: Implement brute force or greedy solution for TSP.

Level 2: Model sequence reassembly as a TSP and analyze path optimization.

week6:

Labsheet-6 (Knapsack Problem in Bioinformatics)

Level 1: Implement 0/1 Knapsack problem using dynamic programming.

Level 2: Apply knapsack logic to biological scenarios such as protein interaction or resource optimization.

week7:

Labsheet-7 (NP-Complete Problems & Reduction Concepts)

Level 1: Demonstrate polynomial-time reduction between simple problems.

Level 2: Analyze NP-complete problems in bioinformatics and discuss feasible approximations.

Module : 3 (Labsheets 8 to 11)

week8:

Labsheet-8 (Naive String Matching for DNA Sequences)

Level 1: Implement naive pattern matching algorithm for DNA sequences.

Level 2: Detect biological motifs such as start and stop codons.

week9:

Labsheet-9 (KMP Algorithm for Sequence Matching)

Level 1: Implement KMP algorithm for efficient string matching.

Level 2: Compare its performance with naive approach on large genomic datasets.

week10:

Labsheet-10 (Boyer-Moore Algorithm)

Level 1: Implement Boyer-Moore string matching algorithm.

Level 2: Analyze performance improvement over other algorithms for large sequences.

week11:

Labsheet-11 (Graph Algorithms in Bioinformatics)

Level 1: Represent biological data using graphs and implement shortest path algorithms.

Level 2: Apply graph algorithms for genome assembly using de Bruijn graph concept.

Module : 4 (Labsheets 12 to 15)

week12:

Labsheet-12 (Dynamic Programming for Sequence Alignment)

Level 1: Implement Needleman-Wunsch algorithm for global alignment.

Level 2: Compare with Smith-Waterman algorithm for local alignment.

week13:

Labsheet-13 (BLAST and FASTA Tools)

Level 1: Perform BLAST search using Biopython and analyze results.

Level 2: Interpret alignment scores and evaluate biological significance.

week14:

Labsheet-14 (Motif Discovery using Probabilistic Methods)

Level 1: Implement simple motif finding approach using pattern matching.

Level 2: Apply Gibbs sampling for motif discovery and analyze results.

week15:

Labsheet-15 (Genome Assembly & Protein Prediction)

Level 1: Simulate DNA sequence assembly using overlap-layout approach.

Level 2: Explore protein structure prediction techniques and compare methods.

Targeted Application & Tools

Targeted 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.
Software/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

1. Assignment 1: Covers concepts and problems from Module 1 and Module 2.
2. 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

1. **T1**. Phillip Compeau & Pavel Pevzner, Bioinformatics Algorithms: An Active Learning Approach, Vol. 1, 2nd Ed., Active Learning Publishers, 2019.
2. **T2**. Michael T. Goodrich & Roberto Tamassia, Algorithm Design and Applications, Wiley, 2015.
3. **T3**. Jason Kinser, Computational Biology: A Hypertextbook, 2nd Ed., Jones & Bartlett Learning, 2021.
4. **T4**. Gautam B. Singh, Fundamentals of Bioinformatics and Computational Biology, Springer, 2015.

References

1. **R1**. Zhumur Ghosh &Bibekanand Mallick, Bioinformatics: Principles and Applications, Oxford University Press, 2014.
2. **R2**. Steven Skiena, The Algorithm Design Manual, Springer, 2nd Ed., 2008.
3. **R3**. Pavel Pevzner, Computational Molecular Biology: An Algorithmic Approach, MIT Press, 2000.
4. **R4**. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd Ed., MIT Press, 2009.
5. **R5**. Arthur Lesk, Introduction to Bioinformatics, 5th Ed., Oxford University Press, 2019.

Web Resources

1. NPTEL Course on Bioinformatics Algorithms: https://onlinecourses.nptel.ac.in/noc25_cs06/preview
2. MIT OpenCourseWare - Algorithms for Computational Biology: https://ocw.mit.edu/courses/6-096-algorithms-for-computational-biology-spring-2005/pages/lecture-notes/
3. Rosalind (Online Platform for Bioinformatics Problems): http://rosalind.info/

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.

Course Code: CSE3508	**Course Title: Statistical Methods for BioInformatics** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE3506 – Introductionto Bioinformatics					
Anti-requisites	NIL					
Course 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.					
Course Objective	To develop a strong foundation in statistical methods for bioinformatics by enabling students to understand statistical concepts for biological data analysis, apply probabilistic models to bioinformatics problems, analyze genomic and proteomic datasets, and interpret statistical results in a biological context.					
Course OutComes	On successful completion of this course, students will be able to: **CO1:** Explain the basic concepts of bioinformatics, including major biological databases. [Understand] **CO2:** Apply 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. [Analyze] **CO5:** Interpret statistical results to derive meaningful biological insights and support decision-making. [Evaluate]					
Course Content:						
Module 1	**Bioinformatics Fundamentals and Database Systems**				**8 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) .						
Module 2	**Sequence Alignment**				**7 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						
Module 3	**Basic Concepts In Biostatistics**				**8 Sessions**	

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.

Module 4	**Biostatistics-Applications**	**7 Sessions**

Topics:

Introduction 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 .

List of Laboratory Tasks: **30 Sessions**

Module : 1 (Labsheets 1 to 4)

week1:

Labsheet-1 (Introduction to Bioinformatics Databases – NCBI)

Level 1: Retrieve DNA and protein sequences from NCBI using accession numbers.

Level 2: Analyze retrieved sequences and interpret biological significance such as gene function and organism details.

week2:

Labsheet-2 (Exploring Bioinformatics Portals – EBI &ExPASy)

Level 1: Explore EBI and ExPASy portals to retrieve biological data.

Level 2: Compare functionalities of different portals and extract relevant biological insights.

week3:

Labsheet-3 (Sequence Databases – GenBank & ENA)

Level 1: Retrieve gene and sequence information from GenBank and ENA databases.

Level 2: Compare sequence annotations and identify differences in database representations.

week4:

Labsheet-4 (Protein Databases – SwissProt& PROSITE)

Level 1: Identify protein sequences and domains using SwissProt and PROSITE.

Level 2: Analyze protein structure/function relationships based on domain information.

Module : 2 (Labsheets 5 to 8)

week5:

Labsheet-5 (Pairwise Sequence Alignment)

Level 1: Perform pairwise sequence alignment using EMBOSS Needle.

Level 2: Interpret alignment scores, gap penalties, and biological relevance.

week6:

Labsheet-6 (Local Alignment using BLAST)

Level 1: Perform BLAST search for given sequences.

Level 2: Analyze similarity scores, E-values, and identify homologous sequences.

week7:

Labsheet-7 (Multiple Sequence Alignment)

Level 1: Perform multiple sequence alignment using Clustal Omega.

Level 2: Analyze conserved regions and evolutionary relationships.

week8:

Labsheet-8 (Phylogenetic Tree Construction)

Level 1: Construct phylogenetic trees using MEGA or online tools.

Level 2: Interpret evolutionary relationships among species based on tree structure.

Module : 3 (Labsheets 9 to 11)

week9:

Labsheet-9 (Data Collection and Classification)
Level 1: Classify biological data into different types of variables.
Level 2: Analyze sampling methods and errors in biological datasets.
week10:
Labsheet-10 (Data Visualization Techniques)
Level 1: Create bar charts, histograms, and pie charts using Python or Excel.
Level 2: Interpret visualized biological data and identify patterns.
week11:
Labsheet-11 (Descriptive Statistics)
Level 1: Calculate mean, median, mode, variance, and standard deviation.
Level 2: Compare datasets using measures of central tendency and dispersion.
Module : 4 (Labsheets 12 to 15)
week12:
Labsheet-12 (Correlation and Regression Analysis)
Level 1: Perform correlation analysis on biological datasets.
Level 2: Build regression models and interpret relationships between variables.
week13:
Labsheet-13 (Hypothesis Testing – t-test & Z-test)
Level 1: Perform t-test and Z-test for given datasets.
Level 2: Interpret results and validate biological hypotheses.
week14:
Labsheet-14 (Chi-Square Test)
Level 1: Conduct chi-square test for categorical biological data.
Level 2: Analyze independence and goodness of fit.
week15:
Labsheet-15 (ANOVA for Biological Data Analysis)
Level 1: Perform one-way ANOVA on biological datasets.
Level 2: Compare multiple group means and interpret statistical significance.

Project work/Assignment:

1. Assignment 1 on (Module 1 and Module 2)
2. Assignment 2 on (Module 3 and Module 4)

Text Book

1. **T1** Arthur Lesk, Introduction to Bioinformatics, Oxford University Press, 5th Ed., 2019
2. **T2** Wayne W. Daniel & Chad L. Cross, Biostatistics: A Foundation for Analysis in the Health Sciences, Wiley, 11th Ed., 2019
3. **T3** Gautam B. Singh, Fundamentals of Bioinformatics and Computational Biology, Springer, 2015

References

1. **R1** Zhumur Ghosh &Bibekanand Mallick, Bioinformatics: Principles and Applications, OUP, 2014
2. **R2** David W. Mount, Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor, 2nd Ed., 2004
3. **R3** Marcello Pagano & Kimberlee Gauvreau, Principles of Biostatistics, CRC Press, 2nd Ed., 2018
4. **R4** S. C. Gupta & V. K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand

1. **Web Resources**
2. https://www.ncbi.nlm.nih.gov
3. https://archive.nptel.ac.in/courses/102/101/102101056/
4. https://onlinecourses.nptel.ac.in/noc25_bt06/preview
5. https://www.ebi.ac.uk
6. https://www.expasy.org

.

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

Course Code: CSE3509	**Course Title: Emerging Technologies in Big Data Type of Course: Lab Integrated Course**	**L-T- P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2500 Data Analytics					
Anti-requisites	NIL					
Course Description	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.					
Course Objectives	To develop a strong foundation in big data analytics by enabling students to understand big data technologies and frameworks, explore distributed computing and storage systems, apply big data tools for data processing, and analyze real-world big data applications.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Apply MapReduce programming techniques on large datasets to extract meaningful insights. [Apply] **CO2:** Employ Hadoop ecosystem tools such as Sqoop, HBase, and Hive for data ingestion, storage, and analytics. [Apply] **CO3:** Use Apache Spark for efficient processing and analysis of large-scale datasets. [Apply] **CO4:** Implement end-to-end big data workflows by integrating multiple tools and technologies to solve real-world problems. [Apply] **CO5:** Analyze and evaluate the performance of big data processing frameworks and optimize solutions for scalability and efficiency. [Analyze]					
Course Content:						
Module 1	**Introduction to Hadoop**					**10Sessions**
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).
Programming Assignment: Data Collection and Analysis

Module 2	**Hadoop Ecosystem Tools**	**10Sessions**

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**	**10Sessions**

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: **30 Sessions**
Module 1: Introduction to Hadoop (Labsheets 1–5)
Week 1:
LabSheet-1 (Hadoop Installation & HDFS Basics)
Level 1: Install Hadoop in pseudo-distributed mode and execute basic HDFS shell commands such as creating directories, uploading files, listing contents, and reading files.
Level 2: Perform advanced HDFS operations such as changing replication factor, checking disk usage, managing file permissions, and analyzing storage distribution.
Week 2:
LabSheet-2 (HDFS File Read/Write Operations)
Level 1: Write and execute programs to demonstrate file write and read operations in HDFS and observe block-level storage behavior.
Level 2: Analyze how replication and block size affect performance and fault tolerance during read/write operations.
Week 3:
LabSheet-3 (Basic MapReduce – Word Count)
Level 1: Implement a MapReduce program to count word frequencies in a dataset.
Level 2: Modify the program to count only selected keywords and analyze execution stages (map, shuffle, reduce).
Week 4:
LabSheet-4 (MapReduce Execution Pipeline Analysis)
Level 1: Execute a MapReduce job and trace each phase including mapping, shuffling, sorting, and reducing.
Level 2: Experiment with combiners and partitioners to optimize performance and reduce intermediate data.
Week 5:
LabSheet-5 (Weather Data Analysis using MapReduce)
Level 1: Implement MapReduce to compute minimum, maximum, and average temperatures from a dataset.
Level 2: Extend the analysis to identify trends and anomalies across years and interpret insights.
Module 2: Hadoop Ecosystem Tools (Labsheets 6–10)
Week 6:

LabSheet-6 (Hive Basic Operations)
Level 1: Create, modify, and delete tables in Hive and execute simple queries.
Level 2: Perform data insertion, filtering, and aggregation queries on structured datasets.
Week 7:
LabSheet-7 (Hive Partitioning Techniques)
Level 1: Implement static and dynamic partitioning in Hive tables.
Level 2: Compare query performance with and without partitioning and justify the design choice.
Week 8:
LabSheet-8 (Hive Bucketing)
Level 1: Create bucketed tables and distribute data into buckets.
Level 2: Compare partitioning vs. bucketing for query optimization and join operations.
Week 9:
LabSheet-9 (Sqoop Data Transfer)
Level 1: Import data from a relational database into HDFS using Sqoop.
Level 2: Export processed data back to a relational database and analyze data consistency.
Week 10:
LabSheet-10 (HBase Operations)
Level 1: Perform basic HBase operations such as creating tables and inserting data.
Level 2: Implement full CRUD operations and design schema for real-world datasets like employee or insurance systems.
Module 3: Apache Spark (Labsheets 11–15)
Week 11:
LabSheet-11 (Spark Installation & RDD Basics)
Level 1: Install Apache Spark and create basic RDDs using different data sources.
Level 2: Perform transformations and actions to manipulate datasets and observe lazy evaluation.
Week 12:
LabSheet-12 (Spark Word Count & Filtering)
Level 1: Implement word count using Spark and display results.
Level 2: Filter words based on conditions and sort results to analyze patterns.
Week 13:
LabSheet-13 (Spark Log File Analysis)
Level 1: Process log files using Spark and extract key information such as request counts.
Level 2: Identify patterns such as error rates, peak usage times, and system performance trends.
Week 14:
LabSheet-14 (Spark SQL and DataFrames)
Level 1: Load datasets into Spark SQL and perform queries using DataFrames.
Level 2: Optimize queries using caching, indexing, and query planning techniques.
Week 15:
LabSheet-15 (End-to-End Big Data Pipeline)
Level 1: Build a pipeline integrating data ingestion, storage, and processing using Hadoop and Spark.
Level 2: Evaluate system performance, scalability, and optimize the workflow for large datasets.

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.						
Course Code: CSE3510	**Course Title: Statistical Techniques for Data Science** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	MAT1003 Applied Statistics					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a strong foundation in statistical methods for data science by enabling students to understand statistical principles, apply statistical techniques for data analysis, perform hypothesis testing and regression analysis, and interpret data-driven insights effectively.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the significance, challenges, and applications of Big Data, including high-dimensional data across various domains. [Understand] **CO2:** Explain multiple linear regression models and the Gauss–Markov theorem. [Understand] **CO3:** Apply regularization techniques and perform inference in linear regression, including random design and partial linear regression models. [Apply] **CO4:** Apply numerical methods such as the power method and explore factor models and structured covariance learning. [Apply] **CO5:** Analyze and implement clustering techniques and variable selection methods for high-dimensional data analysis. [Analyze]					
Course Content:						
Module 1	**Introduction**					**8Sessions**
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.

Module 2	Multiple Linear Regression	7Sessions

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	8Sessions

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	7Sessions

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.

List of Laboratory Tasks: **30 Sessions**

Module 1: Introduction (Labsheets 1–3)
Week 1:
LabSheet-1 (Data Loading & Descriptive Statistics)
Level 1: Load a structured dataset using Python (Pandas) and compute descriptive statistics such as mean, median, mode, variance, and standard deviation.
Level 2: Handle missing values and outliers using imputation and transformation techniques, and visualize their impact on data distribution.
Week 2:
LabSheet-2 (Data Visualization & Noise Analysis)
Level 1: Visualize datasets using histograms, box plots, and scatter plots to understand distributions and variability.
Level 2: Analyze noise accumulation and identify spurious correlations in high-dimensional data using visualization and correlation matrices.
Week 3:
LabSheet-3 (Exploratory Data Analysis for High-Dimensional Data)
Level 1: Perform exploratory data analysis (EDA) on a high-dimensional dataset using summary statistics and visualization.
Level 2: Identify redundant or highly correlated features and discuss their impact on model performance.
Module 2: Multiple Linear Regression (Labsheets 4–8)
Week 4:
LabSheet-4 (Simple Linear Regression)
Level 1: Implement simple linear regression to model the relationship between a dependent and independent variable.
Level 2: Evaluate model performance using metrics such as R^2, MSE, RMSE, and interpret the results.
Week 5:
LabSheet-5 (Multiple Linear Regression)
Level 1: Build a multiple linear regression model using multiple predictors.

Level 2: Interpret regression coefficients and identify significant predictors using statistical measures.
Week 6:
LabSheet-6 (Model Diagnostics & Assumptions)
Level 1: Generate diagnostic plots such as residual plots and Q-Q plots.
Level 2: Detect violations of assumptions like multicollinearity and apply corrective measures.
Week 7:
LabSheet-7 (Ridge Regression & Regularization)
Level 1: Implement Ridge Regression and observe coefficient shrinkage.
Level 2: Perform cross-validation to determine optimal regularization parameter and compare with linear regression.
Week 8:
LabSheet-8 (Generalized Linear Models – Logistic Regression)
Level 1: Implement Logistic Regression for binary classification.
Level 2: Evaluate model using classification metrics and interpret odds ratios.
Module 3: Inference in Linear Regression (Labsheets 9–11)
Week 9:
LabSheet-9 (Statistical Inference for Regression Coefficients)
Level 1: Perform t-tests to evaluate significance of regression coefficients.
Level 2: Construct confidence intervals and interpret statistical significance.
Week 10:
LabSheet-10 (Hypothesis Testing in Regression)
Level 1: Conduct F-test for overall model significance.
Level 2: Use ANOVA to compare models and test multiple hypotheses.
Week 11:
LabSheet-11 (Advanced Regression Inference & Random Design)
Level 1: Explore inference techniques for regression under random design settings.
Level 2: Analyze statistical efficiency and discuss Fisher information in model estimation.
Module 4: PCA and Clustering (Labsheets 12–15)
Week 12:
LabSheet-12 (Principal Component Analysis – Basics)
Level 1: Apply PCA to reduce dimensionality of a dataset.
Level 2: Visualize transformed data and interpret principal components.
Week 13:
LabSheet-13 (PCA Applications & Optimization)
Level 1: Determine optimal number of components using scree plot.
Level 2: Use PCA as preprocessing for another model and compare performance.
Week 14:
LabSheet-14 (Clustering – K-Means & Hierarchical)
Level 1: Implement K-means clustering and visualize clusters.
Level 2: Apply hierarchical clustering and compare clustering structures.
Week 15:
LabSheet-15 (Advanced Clustering & Variable Selection)
Level 1: Implement Gaussian Mixture Models for clustering.
Level 2: Perform feature selection using correlation screening and evaluate its effect on clustering quality.

Targeted Application & Tools that can be used:

1. Predictive Modeling in various scientific and business domains.
2. Data Exploration and Insight Generation.
3. Feature Engineering and Selection.
4. Clustering and Segmentation.
5. 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.

Text Book

1. **T1**: Fan, J., Li, R., Zhang, C.-H., and Zou, H. (2020). Statistical Foundations of Data Science. CRC Press.
2. **T2**: Wainwright, M. J. (2019). High-dimensional statistics: A non-asymptotic viewpoint. Cambridge University Press.

References

1. **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.
2. **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.
3. **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:

1. https://www.youtube.com/playlist?list=PLOU2XLYxmsIK9qQfztXeybpHvru-TrqAP (General YouTube resources for data science)
2. 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.

Course Code: CSE3511	**Course Title: Predictive Analytics and Applications** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1					
Course Pre-requisites	MAT1003 Applied Statistics					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a strong foundation in predictive analytics by enabling students to understand predictive modeling techniques, apply machine learning models for forecasting, evaluate model performance, and solve real-world predictive analytics problems.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Define the nature of analytics and its key applications across various domains. [Remember] **CO2:** Explain the fundamental concepts of predictive analytics and data mining techniques.					

	[Understand] **CO3:** Apply analytical tools and techniques to solve business problems and achieve competitive advantage. [Apply] **CO4:** Build predictive models using methods such as decision trees and time series analysis for real-world scenarios. [Apply] **CO5:** Analyze and interpret predictive analytics results to support data-driven decision-making in dynamic environments. [Analyze]

Course Content:

Module 1	**Introduction to Predictive Analytics**	**8Sessions**

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.

Module 2	**Principles and Techniques**	**7Sessions**

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**	**8Sessions**

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**	**7Sessions**

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.

List of Laboratory Tasks: **30 Sessions**

Module : 1 (Labsheets 1 to 3)

Week 1:

Labsheet-1 (Introduction to Predictive Analytics & Applications)
Level 1: Identify and list three real-world applications of predictive analytics in healthcare, finance, and retail. Explain the prediction goal in each case.
Level 2: Analyze a detailed case study (e.g., customer churn prediction). Identify implementation challenges and propose suitable solutions.

Week 2:

Labsheet-2 (Analytics in Decision Making)
Level 1: Explain how predictive analytics supports decision-making in a business scenario such as inventory management or fraud detection.
Level 2: Design a predictive analytics-based decision system for a retail business including input data, model type, and expected output.

Week 3:

Labsheet-3 (Challenges in Predictive Analytics & Case Analysis)
Level 1: List and explain common challenges in predictive analytics such as data quality, bias, and scalability.
Level 2: Suggest strategies to overcome these challenges in a real-world predictive analytics application.

Module : 2 (Labsheets 4 to 9)

Week 4:

Labsheet-4 (Data Loading and Inspection)
Level 1: Load a dataset using Python (Pandas) or R and display structure, dimensions, and summary statistics.
Level 2: Perform data cleaning by handling missing values and outliers. Compare dataset characteristics before and after cleaning.

Week 5:

Labsheet-5 (Univariate Statistical Analysis)
Level 1: Calculate mean, median, mode, variance, and standard deviation for selected variables.
Level 2: Create histograms and boxplots, and interpret distribution characteristics such as skewness.

Week 6:

Labsheet-6 (Bivariate Analysis and Correlation)
Level 1: Compute correlation between two numerical variables and interpret the results.
Level 2: Visualize relationships using scatter plots with regression lines and analyze their impact on prediction.

Week 7:

Labsheet-7 (Logistic Regression – Propensity Models)
Level 1: Build a logistic regression model to predict a binary outcome (e.g., churn: Yes/No).
Level 2: Extend the model with multiple predictors and evaluate using accuracy and confusion matrix.

Week 8:

Labsheet-8 (Clustering using K-Means)
Level 1: Apply K-Means clustering to segment customers based on selected features.
Level 2: Determine optimal number of clusters using Elbow or Silhouette method and interpret cluster profiles.

Week 9:

Labsheet-9 (Collaborative Filtering – Recommendation Systems)
Level 1: Explain user-based and item-based collaborative filtering with examples.
Level 2: For a small dataset, manually compute similarity and generate recommendations.

Module : 3 (Labsheets 10 to 12)

Week 10:

Labsheet-10 (Train-Test Split and Cross Validation)

Level 1: Split dataset into training and testing sets and evaluate model performance.
Level 2: Implement K-Fold Cross Validation and compare results with simple split.

Week 11:

Labsheet-11 (Linear and Polynomial Regression)
Level 1: Build a multiple linear regression model and interpret coefficients.
Level 2: Apply polynomial regression and compare performance with linear regression.

Week 12:

Labsheet-12 (Decision Trees – Classification and Regression)
Level 1: Build and visualize a decision tree model.
Level 2: Evaluate using performance metrics and apply pruning techniques.

Module : 4 (Labsheets 13 to 15)

Week 13:

Labsheet-13 (Time Series Visualization and Decomposition)
Level 1: Plot time series data and identify trend and seasonality.
Level 2: Decompose time series into components and perform differencing.

Week 14:

Labsheet-14 (ARIMA Model Building)
Level 1: Identify ARIMA parameters using ACF and PACF plots.
Level 2: Fit ARIMA model and analyze residuals.

Week 15:

Labsheet-15 (Forecasting and Model Evaluation)
Level 1: Forecast future values using ARIMA or Holt-Winters model.
Level 2: Evaluate forecast accuracy using MAE, RMSE, and MAPE and suggest improvements.

Targeted Application & Tools that can be used:

1. Business Forecasting (Sales, Demand, Stock Prices)
2. Customer Behavior Prediction (Churn, Propensity to Buy)
3. Risk Assessment (Credit Risk, Fraud Detection)
4. Recommendation Systems
5. Market Basket Analysis
6. **Tools:** Python (Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Statsmodels), R (tidyverse, caret, forecast, ggplot2), Jupyter Notebook, RStudio.

Text Book

1. **T1** Jeffrey Strickland, Predictive Analytics Using R, Simulation Educators, Colorado Springs, 2015.
2. **T2** Max Kuhn and Kjell Johnson, Applied Predictive Modeling, 1st edition, Springer, 2013.

References

1. **R1:** Dinesh Kumar, U. (2021). *Business Analytics: The Science of Data-Driven Decision Making*.
2. **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*.

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: **CSE3512**	**Course Title: Data Mining** **Type of Course: Theory 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	To develop a strong foundation in data mining by enabling students to understand core concepts and techniques, apply classification, clustering, and association rule mining methods, extract meaningful patterns from large datasets, and evaluate mining results to support effective decision-making.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Apply data preprocessing techniques required for effective data mining tasks. [Apply] **CO2:** Explain the working principles of various data mining algorithms. [Understand] **CO3:** Apply appropriate data mining models for classification, clustering, and association rule mining problems. [Apply] **CO4:** Evaluate the strengths and limitations of different data mining techniques and models. [Evaluate] **CO5:** Analyze recent advances in data mining and assess their applicability in real-world scenarios. [Analyze]					
Course Content:						
Module 1	**Introduction to Data Mining**					**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**					**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**	**10 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**	**12 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**	**9 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

CO1. From the dataset given, find the Entropy, Gain value of the attributes and also draw the decision tree using entropy for the given dataset.

CO2. 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

1. 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.") 2. 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 3. 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: CSE3513	**Course Title: No SQL Data Management** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2500 Data Analytics					
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	To develop a comprehensive understanding of NoSQL databases by enabling students to learn their principles, models, and practical applications, analyze the advantages over traditional relational databases, understand various data models such as key-value, document, column-family, and graph, explore distributed architectures for scalability and availability, and apply MapReduce and related techniques for efficient data processing.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the limitations of relational databases and the motivation for the emergence of NoSQL databases. [Understand] **CO2:** Analyze and differentiate various NoSQL data models such as key-value, document, column-family, and graph databases based on structure, use cases, and performance. [Analyze] **CO3:** Apply 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. [Create] **CO5:** Apply MapReduce and other data processing techniques to efficiently handle large-					

	scale data in NoSQL environments. [Apply]

Course Content:

Module 1	**Introduction to NoSQL and Aggregate-Oriented Data Models**	**8Sessions**

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**	**6Sessions**

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**	**6Sessions**

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**	**6Sessions**

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**	**4 Sessions**

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: **30 Sessions**

Module : 1 (Labsheets 1 to 3)
Week 1:
Labsheet-1 (Introduction to NoSQL & Motivation)
Level 1: Compare Relational (SQL) and NoSQL databases. Highlight key limitations of relational databases.
Level 2: Analyze a real-world scenario (e.g., social media feed) and justify why NoSQL is more suitable than SQL.
Week 2:
Labsheet-2 (Aggregate-Oriented Data Models)
Level 1: Explain aggregate-oriented data models with examples (key-value, document, column-family).
Level 2: Design an aggregate-oriented model for an e-commerce product catalog and compare it with a relational schema.

Week 3:
Labsheet-3 (Key-Value and Document Data Models)
Level 1: Implement basic CRUD operations using a Key-Value store (e.g., Redis).
Level 2: Explore advanced Redis features like expiration, transactions, and implement a caching system.
Module : 2 (Labsheets 4 to 6)
Week 4:
Labsheet-4 (Document Databases – MongoDB Basics)
Level 1: Install MongoDB and perform CRUD operations with nested documents.
Level 2: Perform complex queries using aggregation pipeline ($match, $group) and design a schema for a blog system.
Week 5:
Labsheet-5 (Sharding and Replication)
Level 1: Explain sharding and replication concepts with examples.
Level 2: Design a sharding strategy (range-based or hash-based) and simulate data distribution.
Week 6:
Labsheet-6 (CAP Theorem and Consistency Models)
Level 1: Explain CAP theorem and trade-offs between consistency, availability, and partition tolerance.
Level 2: Compare NoSQL databases (MongoDB, Cassandra, Redis) based on CAP theorem and justify use cases.
Module : 3 (Labsheets 7 to 9)
Week 7:
Labsheet-7 (Consistency Models and Quorums)
Level 1: Explain strong consistency and eventual consistency with examples.
Level 2: Implement/read-write quorum concept and analyze impact on availability and consistency.
Week 8:
Labsheet-8 (MapReduce Basics)
Level 1: Explain MapReduce model with working steps (Map, Shuffle, Reduce).
Level 2: Implement a simple word count program using Hadoop/Spark.
Week 9:
Labsheet-9 (Advanced MapReduce & Key-Value Use Cases)
Level 1: Explain partitioning and combining in MapReduce.
Level 2: Implement multi-stage MapReduce (e.g., log analysis) and design a key-value schema for session storage or shopping cart.
Module : 4 (Labsheets 10 to 12)
Week 10:
Labsheet-10 (Document Database Queries)
Level 1: Perform CRUD operations and simple queries in MongoDB.
Level 2: Implement aggregation pipelines for data analysis (e.g., sales summary).
Week 11:
Labsheet-11 (Document Schema Design & CMS)
Level 1: Design a schema for a content management system using MongoDB.
Level 2: Implement a real-time analytics system storing event logs.
Week 12:
Labsheet-12 (E-Commerce Application using Document DB)
Level 1: Model product catalog using document database.
Level 2: Implement search and filtering features using query operators.
Module : 5 (Labsheets 13 to 15)
Week 13:
Labsheet-13 (Graph Database Basics – Neo4j)
Level 1: Explain nodes, relationships, and properties in graph databases.
Level 2: Install Neo4j and create a sample graph.
Week 14:
Labsheet-14 (Graph Queries using Cypher)
Level 1: Perform basic graph traversal queries using Cypher.
Level 2: Implement queries for shortest path and connected components.
Week 15:
Labsheet-15 (Graph Database Applications – Recommendation System)
Level 1: Design a graph schema for a social network.
Level 2: Implement a recommendation system using graph traversal.

Textbook

1. **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

1. **R1** Dan Sullivan, "NoSQL For Mere Mortals", 1st Edition, Pearson Education India, 2015. (ISBN-13: 978-9332557338)
2. **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)
3. **R3** Kristina Chodorow, "Mongodb: The Definitive Guide- Powerful and Scalable Data Storage", 2nd Edition, O'Reilly Publications, 2013. (ISBN-13: 978-9351102694)
4. 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: CSE3514	**Course Title: Applied Data Intelligence** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2264 Essentials of AI					
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	o develop practical expertise in data intelligence by enabling students to apply intelligent techniques for data-driven solutions, integrate machine learning with data analytics, solve real-world problems using data intelligence approaches, and evaluate the performance of intelligent systems.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain NumPy fundamentals and perform matrix and array operations. [Understand] **CO2:** Analyze the need for data preprocessing and apply data cleaning and visualization techniques. [Analyze] **CO3:** Implement supervised learning algorithms such as Decision Trees, Random Forest, Linear Regression, and Logistic Regression, and evaluate their performance. [Evaluate]					

	CO4: Apply unsupervised learning algorithms such as K-Means and K-Medoids for clustering and pattern discovery. [Apply] **CO5:** Interpret and analyze model results to support data-driven decision-making in practical applications. [Analyze]

Course Content:

Module 1	**Introduction to Data Science, Python Data Structures, Python Numpy Package**	**8 Sessions**

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**	**7 Sessions**

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**	**8 Sessions**

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**	**7 Sessions**

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: **30 Sessions**

Module : 1 (Labsheets 1 to 3)
Week 1:

Labsheet-1 (Python Fundamentals & Data Structures)

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).

Level 2: Implement advanced NumPy operations such as matrix multiplication, transpose, and slicing. Explain the concept of broadcasting in NumPy with an example.

Week 2:
Labsheet-2 (Pandas Data Handling & Initial Exploration)

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.

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.

Week 3:

Labsheet-3 (Data Preprocessing: Normalization & Standardization)

Level 1: Apply Min-Max scaling (Normalization) to a numerical feature in your dataset. Visualize the distribution before and after scaling.

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.

Module : 2 (Labsheets 4 to 6)

Week 4:

Labsheet-4 (Exploratory Data Analysis (EDA) & Basic Visualization)

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.

Level 2: Generate scatter plots to investigate relationships between two numerical variables. Create box plots to compare distributions across different categories, identifying potential outliers.

Week 5:

Labsheet-5 (Advanced Data Visualization)

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.

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 (Labsheets 6 to 10)

Week 6:

Labsheet-6 (Linear Regression)

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.

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.

Week 7:

Labsheet-7 (Logistic Regression)

Level 1: Implement a Logistic Regression model using scikit-learn for a binary classification problem. Make predictions and obtain class probabilities.

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.

Labsheet-8 (Decision Trees)

Level 1: Build a Decision Tree Classifier using scikit-learn on a classification dataset. Train the model and make predictions.

Level 2: Evaluate the Decision Tree's performance and visualize the tree structure. Experiment with pruning parameters (e.g., max_depth) to avoid overfitting.

Week 9:

Labsheet-9 (Random Forest Classifier)

Level 1: Implement a Random Forest Classifier using scikit-learn for a classification task. Compare its basic accuracy with the Decision Tree.

Level 2: Analyze the feature importances derived from the Random Forest model and discuss how ensemble methods can improve predictive performance.

Week 10:
Labsheet-10 (Naïve Bayesian Classifier)

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.

Level 2: Evaluate the Naïve Bayes model's performance and discuss its assumptions and suitability for different types of data.

Module 4: (Labsheets 11 to 15)
Week 11:
Labsheet-11(K-Means Clustering)

Level 1: Apply the K-Means clustering algorithm using scikit-learn to an unlabeled dataset. Visualize the clustered data points.

Level 2: Use the Elbow Method or Silhouette Score to determine the optimal number of clusters (K) for your dataset and justify your choice.

Week 12:
Labsheet-12(Dimensionality Reduction: Principal Component Analysis (PCA))

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.

Level 2: Visualize the data in the reduced PCA space and explain how PCA captures the most variance. Discuss the cumulative explained variance ratio.

Week 13:
Labsheet-13 (Dimensionality Reduction: Singular Value Decomposition (SVD))

Level 1: Apply Singular Value Decomposition (SVD) to a given matrix (e.g., from a dataset) using NumPy's linalg.svd function.

Level 2: Explain the components of SVD (U, S, V) and discuss its applications, particularly in dimensionality reduction and recommender systems.

Week 14:
Labsheet-14 (Association Rule Mining (Conceptual & Basic Implementation))

Level 1: Understand the concepts of Support, Confidence, and Lift in association rule mining. Discuss a real-world scenario (e.g., market basket analysis).

Level 2: Use a library like mlxtend to find frequent itemsets using the Apriori algorithm and generate association rules from a small transactional dataset.

Week 15:
Labsheet-15 (Applying Unsupervised Learning for Insights)
Level 1: Select a real-world dataset (e.g., customer transaction data, image dataset) and apply a clustering algorithm (e.g., K-Means).
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:
1. 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. 2. 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. 3. 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: CSE3535	**Course Title: Could Data Engineering** **Type of Course: Lab Integrated Course**	**L-T- P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2285 Cloud Computing					
Anti-requisites	Nil					
Course Description	This course introduces Cloud Computing and Data Engineering principles as a foundation for designing scalable, secure, and cost-effective cloud-based systems. It covers cloud service models (IaaS, PaaS, SaaS), virtualization, distributed computing paradigms, and high-throughput data processing. Hands-on labs using platforms like AWS, GCP, and Microsoft Azure are integrated with concepts such as federated and hybrid clouds, media and mobile clouds, and cloud security standards. The curriculum emphasizes designing cloud-enabled solutions with a focus on performance, cost-efficiency, and reliability.					
Course Objective	To develop a comprehensive understanding of cloud computing by enabling students to learn fundamental concepts such as virtualization and cloud service models, design and implement scalable and data-intensive solutions using modern cloud platforms, understand cloud security challenges and compliance standards along with the role of AI in cloud governance, and gain hands-on experience in configuring, managing, and					

	optimizing cloud infrastructure for performance, cost-efficiency, and reliability.
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamentals of cloud computing, virtualization models, and cloud service types. [Understand] **CO2:** Apply high-throughput and data-intensive computing techniques using cloud platforms. [Apply] **CO3:** Explain key security challenges, privacy issues, and compliance standards in cloud environments. [Understand] **CO4:** Demonstrate the setup, configuration, and orchestration of cloud platforms and virtual machines. [Apply] **CO5:** Analyze and optimize cloud-based data engineering solutions for performance, scalability, and reliability. [Analyze]

Course Content:

Module 1 :	**Introduction to Cloud and Virtualization**	**8 Sessions**
Historical evolution of cloud computing, Environments: Cloud setup, Virtualization, IaaS, PaaS, SaaS, Virtualization techniques: full, para, OS-level, Types of clouds: Public, Private, Hybrid, Federated, Economics and architectural overview **AI Perspective:** AI-driven auto-scaling policies based on workload prediction, Virtual machine placement optimization using reinforcement learning		
Module 2 :	**High Throughput and Data Intensive Computing**	**7 Sessions**
Task computing, MPI, Aneka, and MapReduce basics, Distributed programming models and execution environments, Fault-tolerant processing of large-scale datasets **AI Perspective:** Use of AI to automate resource allocation for MapReduce jobs, Predictive scaling in big data clusters using LSTM models		
Module 3 :	**Cloud Security and Standards**	**8 Sessions**
Security challenges and mitigation strategies in cloud environments, SaaS security risks, Identity & Access Management (IAM), Compliance and standardization: NIST, ISO/IEC 27001, GDPR **AI Perspective:** AI-enabled anomaly detection and policy enforcement in cloud infrastructure, Using AI/ML models for dynamic risk scoring and audit readiness		
Module 4 :	**Cloud Platforms, Advances in Cloud**	**7 Sessions**
Overview of AWS, GCP, Microsoft Azure, Media, mobile, and hybrid cloud systems, Real-time application hosting, serverless architecture, federated clouds **AI Perspective:** Generative AI workloads hosted on serverless frameworks,AI-based workload migration decision systems		

between hybrid clouds

List of Laboratory Tasks: **30 Sessions**

Module : 1 (Labsheets 1 to 3)
Week 1:
Labsheet-1 (Introduction to Cloud Computing & Service Models)
Level 1: Explain cloud computing concepts and differentiate IaaS, PaaS, and SaaS with examples.
Level 2: Design a cloud-based solution for a web application using appropriate service models and justify your choices.
Week 2:
Labsheet-2 (Virtualization and Virtual Machine Setup)
Level 1: Install and configure a virtual machine using VirtualBox/VMware and install an OS.
Level 2: Compare full virtualization, para-virtualization, and OS-level virtualization. Analyze their performance and use cases.
Week 3:
Labsheet-3 (Types of Cloud & Architecture Overview)
Level 1: Explain public, private, and hybrid cloud models with examples.
Level 2: Design a hybrid cloud architecture for an organization and explain benefits in terms of cost and scalability.
Module : 2 (Labsheets 4 to 7)
Week 4:
Labsheet-4 (IaaS Cloud Deployment – VM Launch)
Level 1: Launch and configure a virtual machine (AWS EC2 / GCP Compute Engine) and install a web server.
Level 2: Automate VM provisioning and analyze performance under different configurations.
Week 5:
Labsheet-5 (Cloud Storage Integration)
Level 1: Upload and download files using cloud storage (AWS S3 / GCS).
Level 2: Implement versioning and lifecycle policies and analyze storage cost optimization.
Week 6:
Labsheet-6 (High Throughput Computing & MapReduce Basics)
Level 1: Explain MapReduce programming model with example.
Level 2: Implement word count program using Hadoop/Spark.
Week 7:
Labsheet-7 (Distributed Computing & Fault Tolerance)
Level 1: Explain distributed computing concepts and execution models.
Level 2: Simulate fault tolerance in distributed systems and analyze system reliability.
Module : 3 (Labsheets 8 to 10)
Week 8:
Labsheet-8 (Auto Scaling and Load Management)
Level 1: Configure auto-scaling for cloud instances using threshold-based rules.
Level 2: Analyze system performance under varying loads and optimize scaling policies.
Week 9:
Labsheet-9 (IAM and Access Control)
Level 1: Create users, roles, and policies using IAM.
Level 2: Design a secure access control system using role-based access control (RBAC).
Week 10:
Labsheet-10 (Cloud Security & Compliance)
Level 1: Identify cloud security risks and implement basic security configurations.
Level 2: Conduct a security audit and analyze compliance with standards (e.g., ISO/NIST).
Module : 4 (Labsheets 11 to 15)
Week 11:
Labsheet-11 (Serverless Computing – Functions Deployment)
Level 1: Deploy a serverless function (AWS Lambda/Azure Functions).
Level 2: Integrate serverless functions with cloud storage for data processing.
Week 12:
Labsheet-12 (Cloud Monitoring & Logging)
Level 1: Set up monitoring dashboard using CloudWatch or similar tool.
Level 2: Analyze logs and detect anomalies in system performance.

Week 13:
Labsheet-13 (Hybrid Cloud Simulation)
Level 1: Connect local VM to cloud environment.
Level 2: Deploy a hybrid application and analyze latency and performance.
Week 14:
Labsheet-14 (Real-Time Application Hosting)
Level 1: Deploy a simple web application on cloud platform.
Level 2: Optimize application performance using load balancing and caching.
Week 15:
Labsheet-15 (Cloud Optimization & Cost Analysis)
Level 1: Analyze cloud usage and estimate cost for deployed services.
Level 2: Optimize infrastructure for cost, scalability, and reliability using best practices.

Text Book(s):

T1. Behrouz A Forouzan, Data Communications and Networking, 5th Ed., McGraw Hill
T2. James F Kurose & Keith W Ross, Computer Networking: A Top-Down Approach, 6th Ed., Pearson
T3. David E.Y. Sarna, Implementing and Developing Cloud Applications, CRC Press
T4. Anthony T Velte et al., Cloud Computing: A Practical Approach, Tata McGraw-Hill

Reference(s):

R1. Journals / Magazines

- IEEE Transactions on Cloud Computing
- ACM SIGCOMM Computer Communication Review
- Journal of Cloud Computing: Advances, Systems, and Applications

- **R2. SWAYAM / NPTEL / MOOCs**
- NPTEL: Cloud Computing
- Coursera: Google Cloud Data Engineering
- AWS Skill Builder / Azure Learning Paths

Course Code: CSE3516	**Course Title: Federated Learning** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2285Cloud Computing					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a strong foundation in federated learning by enabling students to understand federated learning frameworks, apply distributed machine learning techniques, ensure data privacy and security in model training, and evaluate the performance of federated learning systems.					

Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the key concepts, architecture, and terminology of federated learning. [Understand] **CO2:** Apply methods and algorithms to develop horizontal and vertical federated learning systems. [Apply] **CO3:** Apply optimization techniques in federated learning, including handling non-IID data. [Apply] **CO4:** Design and implement a scalable federated learning system using relevant tools and platforms. [Apply] **CO5:** Evaluate privacy and security concerns in federated learning and apply privacy-preserving techniques. [Evaluate]

Course Content:

Module 1	**Introduction to Federated Learning:**	**10 Sessions**

Topics:

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).**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**	**9 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**	**10 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.

List of Laboratory Tasks: **30 Sessions**

Module 1: Introduction to Federated Learning (Labsheets 1–5)

Week 1

Labsheet–1 (Introduction to Federated Learning Concepts)
Level 1:
Explain the architecture of federated learning with a neat diagram showing server, clients, and communication rounds.

Level 2:
Compare federated learning with centralized learning by implementing a simple centralized ML model and discussing its limitations in privacy and scalability.

Week 2

Labsheet–2 (Setting up TensorFlow Federated Environment)
Level 1:
Install and configure TensorFlow Federated (TFF) and run a basic "Hello World" federated example.

Level 2:
Modify the example to simulate multiple clients and observe how model updates are aggregated.

Week 3

Labsheet–3 (Basic Federated Averaging Implementation)
Level 1:
Implement a simple linear regression model using Federated Averaging (FedAvg).

Level 2:
Track model loss across communication rounds and visualize convergence using plots.

Week 4

Labsheet–4 (Client Data Simulation)
Level 1:
Create synthetic client datasets using TFF simulation APIs.

Level 2:
Simulate varying client sizes and analyze their impact on global model performance.

Week 5

Labsheet–5 (Communication Rounds and Convergence Analysis)
Level 1:
Run federated learning for different numbers of communication rounds.

Level 2:
Analyze trade-offs between communication cost and accuracy using graphs.

Module 2: Horizontal & Vertical Federated Learning (Labsheets 6–10)

Week 6

Labsheet–6 (Non-IID Data Simulation)
Level 1:
Create a non-IID dataset where each client has different class distributions.

Level 2:
Compare performance of IID vs non-IID training and interpret results.

Week 7

Labsheet–7 (Handling Non-IID Data)
Level 1:
Increase local epochs in non-IID training and observe performance changes.

Level 2:
Implement a conceptual FedProx-like modification and compare results.

Week 8

Labsheet–8 (CNN Model using Federated Learning)
Level 1:
Train a CNN on MNIST/Fashion MNIST using FedAvg.

Level 2:
Experiment with different batch sizes and evaluate accuracy.

Week 9

Labsheet–9 (Client Fraction and Participation Analysis)
Level 1:
Train the model with different client participation ratios.

Level 2:
Analyze impact of partial participation on convergence and performance.

Week 10

Labsheet–10 (Vertical Federated Learning – Data Alignment)
Level 1:
Create two datasets representing different features of same users and align them using a common key.

Level 2:
Design a conceptual workflow for secure data sharing in VFL.

Module 3: Federated Transfer Learning & Security (Labsheets 11–15)

Week 11

Labsheet–11 (Secure Multi-Party Computation using PySyft)
Level 1:
Simulate secure addition of private values using PySyft.

Level 2:
Extend to secure averaging among multiple participants.

Week 12

Labsheet–12 (Federated Transfer Learning Workflow)
Level 1:
Describe and implement a basic transfer learning model locally.

Level 2:
Extend it conceptually into a federated transfer learning workflow.

Week 13

Labsheet–13 (Differential Privacy Concepts)
Level 1:
Explain epsilon-differential privacy with an example.

Level 2:
Simulate noise addition to model updates and analyze its effect.

Week 14

Labsheet–14 (Security Attacks in Federated Learning)
Level 1:
Explain model inversion and membership inference attacks.

Level 2:
Propose mitigation strategies and justify their effectiveness.

Week 15

Labsheet–15 (Federated Learning Deployment on Cloud)
Level 1:
Design a cloud architecture for federated learning using AWS/Azure/GCP.

Level 2:
Explain client-server communication and security mechanisms in the architecture.

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

1. Heiko Ludwig (Editor), Nathalie Baracaldo, "Federated learning comprehensive overview of methods and applications", Springer Nature Switzerland AG; 1st ed. 2022 edition.
2. 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

1. Kiyoshi Nakayama PhD, George Jeno, "Federated Learning with Python", O'Reilly Media, Inc.
2. Emily Glanz, Nova Fallen, "What-is-federated learning?", O'Reilly Media, Inc.
3. **E-book Link R1:**http://www.iso.org/iso/home/standards/management-standards/iso27001.html (Relevant for security standards)
4. **E-book Link R2:**https://www.oreilly.com/library/view/what-is-federated/9781098107253/ch03.html
5. Official documentation for TensorFlow Federated (TFF)

6. Official documentation for PySyft 7. 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: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2285Cloud 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	To develop a comprehensive understanding of network security by enabling students to learn security principles, configure and manage firewall systems, analyze network threats and vulnerabilities, and implement effective security policies.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental principles, architectures, and real-world applications of edge computing. [Understand] **CO2:** Explain IoT architectures, core modules, and their integration with edge computing environments. [Understand] **CO3:** Apply edge-to-cloud communication protocols for efficient data exchange in distributed systems. [Apply] **CO4:** Demonstrate the implementation of edge computing solutions using platforms such as Raspberry Pi. [Apply] **CO5:** Analyze and evaluate the performance of edge computing systems in terms of latency, scalability, and resource utilization. [Analyze]					

Course Content:		
Module 1	**IoT and Edge Computing Definition and Use Cases**	**5 Sessions**
Topics: 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**	**6 Sessions**
Topocs: 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: Rquirements, Implementation, Use case retrospective. .		
Module 3	**RaspberryPi**	**7 Sessions**
Topics: 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**	**6 Sessions**
Topics: 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**	**6 Sessions**

Topics:

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.

List of Laboratory Tasks: **30 Sessions**

Module 1: IoT and Edge Computing Fundamentals (Labsheets 1–3)

Week 1

Module 1: Edge Basics & Communication (Lab Sheets 1–3)

Week 1: Lab Sheet 1 (Basic Edge Simulation using Python)

Level 1:
Write a Python program to simulate sensor data generation (temperature, humidity) using random values.

Level 2:
Extend the program to store generated data into a CSV file and display average, min, and max values.

Week 2: Lab Sheet 2 (Client-Server Communication)

Level 1:
Write a Python socket program to send a message from client to server.

Level 2:
Extend the program to simulate sensor data transfer from client (sensor node) to server (edge node).

Week 3: Lab Sheet 3 (Multi-device Communication Simulation)

Level 1:
Write a Python program to simulate multiple clients sending data to a single server.

Level 2:
Modify the program to handle concurrent clients using threading.

Module 2:IoT Data Processing (Lab Sheets 4–6)

Week 4: Lab Sheet 4 (IoT Data Processing using Python)

Level 1:
Write a Python program to read sensor data from a CSV file and display it.

Level 2:
Process the data to detect anomalies (values beyond threshold).

Week 5: Lab Sheet 5 (Real-time Data Visualization)

Level 1:
Plot sensor data using Matplotlib.

Level 2:
Create a real-time updating graph for streaming sensor data.

Week 6: Lab Sheet 6 (Edge Data Filtering)

Level 1:

Write a Python program to filter noise from sensor data (remove outliers).

Level 2:
Implement moving average smoothing for real-time data streams.

Module 3: Raspberry Pi Programming (Lab Sheets 7–11)

Week 7: Lab Sheet 7 (GPIO Programming)

Level 1:
Write a Python program to blink an LED using Raspberry Pi GPIO.

Level 2:
Control LED brightness using PWM.

Week 8: Lab Sheet 8 (Multiple GPIO Control)

Level 1:
Write a Python program to control multiple LEDs.

Level 2:
Control LEDs based on user input or button press.

Week 9: Lab Sheet 9 (Sensor Data Acquisition)

Level 1:
Write a Python program to read temperature and humidity from a DHT sensor.

Level 2:
Store sensor data in a file and display periodic updates.

Week 10: Lab Sheet 10 (Web Server on Raspberry Pi)

Level 1:
Create a Flask-based web server to display static text.

Level 2:
Display real-time sensor data on a web page.

Week 11: Lab Sheet 11 (Image Processing using OpenCV)

Level 1:
Capture an image using Pi Camera and display it.

Level 2:
Apply image processing techniques (grayscale, edge detection).

Module 4: MQTT Communication (Lab Sheets 12–13)

Week 12: Lab Sheet 12 (MQTT Publish-Subscribe)

Level 1:
Write a Python MQTT publisher to send messages.

Level 2:
Write a subscriber program to receive and display messages.

Week 13: Lab Sheet 13 (MQTT Sensor Integration)

Level 1:
Send sensor data using MQTT publisher.

Level 2:

Create multiple subscribers to receive and process the data.

Module 5: Edge Applications (Lab Sheets 14–15)

Week 14: Lab Sheet 14 (Edge Data Processing Application)

Level 1:
Write a Python program to process sensor data locally and display alerts.

Level 2:
Trigger an action (LED/Buzzer) when threshold exceeds.

Week 15: Lab Sheet 15 (Edge-to-Cloud Integration)

Level 1:
Send processed sensor data to a cloud API using HTTP requests.

Level 2:
Compare processing time for local (edge) vs cloud computation and print results.

Targeted Application & Tools that can be used:

Application : Smart Surveillance Video Stream Processing at the Edge for Real-Time Human Objects Tracking.

Tools :EclipseioFog : 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.

Text 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:

1. Official documentation for Raspberry Pi.
2. Official documentation for MQTT.
3. 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: Lab Integrated Course**	**L-T- P- C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2502 Cryptography and Network Security					
Anti-requisites	NIL					
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	To develop a comprehensive understanding of network security by enabling students to learn fundamental security principles, configure and manage firewall systems, analyze network threats and vulnerabilities, and implement effective security policies.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Identify firewall design elements, classify security threats, and formulate appropriate responses to network attacks. [Analyze] **CO2:** Analyze security incidents through post-mortem reporting and evaluate ongoing network security practices. [Evaluate] **CO3:** Implement fundamental authentication algorithms for secure systems. [Apply] **CO4:** Develop digital signature schemes using the Digital Signature Standard (DSS). [Apply] **CO5:** Demonstrate the implementation of network security systems using open-source tools. [Apply]					
Course Content:						
Module 1	**Introduction to Firewall**					**8 Sessions**
Topics: 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**	**7 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**	**8 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**	**7 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: **30 Sessions**

Module 1: Introduction to Firewall (Labsheets 1–4)

Week 1

Labsheet–1 (Introduction to Firewalls & Types)
Level 1:
Explain the concept, purpose, and types of firewalls (packet filtering, stateful, proxy).

Level 2:
Compare hardware vs software firewalls and design a firewall deployment for a small enterprise network.

Week 2

Labsheet–2 (Firewall Rules and Packet Filtering)
Level 1:
Configure basic firewall rules using tools like UFW/iptables to allow/block specific ports.

Level 2:
Design rule sets to secure a web server (HTTP/HTTPS/SSH) and justify each rule.

Week 3

Labsheet–3 (Firewall Architectures and Network Segmentation)
Level 1:
Explain DMZ, dual-homed host, and screened subnet architecture.

Level 2:
Design a segmented network architecture with firewall placement and explain security benefits.

Week 4

Labsheet–4 (Stateful vs Stateless Firewalls)
Level 1:
Demonstrate stateless packet filtering using simple firewall rules.

Level 2:
Analyze advantages of stateful inspection over stateless filtering using traffic scenarios.

Module 2: Computer Security & Cryptographic Techniques (Labsheets 5–8)

Week 5

Labsheet–5 (Substitution Ciphers)
Level 1:
Implement Caesar and Playfair cipher for encryption/decryption.

Level 2:
Implement Vigenère and Hill cipher and analyze strengths and weaknesses.

Week 6

Labsheet–6 (Transposition Techniques)
Level 1:
Implement Rail Fence cipher.

Level 2:
Implement Row-Column cipher and compare with substitution techniques.

Week 7

Labsheet–7 (DES Implementation)
Level 1:
Implement simplified DES using Feistel structure.

Level 2:
Encrypt/decrypt files using DES library and compare ECB vs CBC modes.

Week 8

Labsheet–8 (AES Implementation)
Level 1:
Encrypt/decrypt text using AES (128-bit key).

Level 2:
Implement AES with CBC/GCM mode and analyze role of IV in security.

Module 3: Network Security & Cryptography (Labsheets 9–12)

Week 9

Labsheet–9 (RSA Algorithm)
Level 1:
Implement RSA algorithm with small primes.

Level 2:
Develop a simple web-based RSA encryption tool using HTML & JavaScript.

Week 10

Labsheet–10 (Diffie-Hellman Key Exchange)
Level 1:
Implement Diffie-Hellman key exchange in Python.

Level 2:
Use the shared key to encrypt communication and discuss MITM attack.

Week 11

Labsheet–11 (Hashing using SHA)
Level 1:
Generate SHA-1 hash for given input.

Level 2:
Compare SHA-1 and SHA-256 and demonstrate avalanche effect.

Week 12

Labsheet–12 (Digital Signatures)
Level 1:
Explain digital signature concept and its applications.

Level 2:
Implement a digital signature scheme and verify integrity/authentication.

Module 4: Cyber Security Tools & Firewall Implementation (Labsheets 13–15)

Week 13

Labsheet–13 (Intrusion Detection using Snort)
Level 1:
Install and configure Snort IDS and detect ICMP packets.

Level 2:
Simulate attacks (port scan using Nmap) and analyze Snort alerts.

Week 14

Labsheet–14 (Vulnerability Assessment & Malware Concepts)
Level 1:
Perform port scanning using Nmap and identify services.

Level 2:
Simulate safe Trojan-like behavior and discuss prevention techniques.

Week 15

Labsheet–15 (Firewall Configuration & Network Traffic Analysis)
Level 1:
Configure firewall rules using UFW/Windows Firewall.

Level 2:

Analyze network traffic using Wireshark and identify security threats.

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:

1. **Snort:** For intrusion detection and packet analysis.
2. **N-Stalker:** A vulnerability assessment tool (for exploring concepts).
3. **Wireshark:** For network protocol analysis.
4. **OpenSSL:** For cryptographic operations and certificate management.
5. **Netcat/Nmap:** For basic network reconnaissance and port scanning.
6. **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

1. **T1 :** Behrouz A Forouzan, Data and Communications and Networking, Fifth Edition, McGraw Hill, Indian Edition
2. **T2:** James F Kurose and Keith W Ross, Computer Networking, A Top-Down Approach, Sixth edition, Pearson,2017

References

1. **R1:** Andrew S Tanenbaum, Computer Networks, fifth edition, Pearson Edition
2. **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:

1. **Network Security Implementation:** Practical skills in configuring and managing network security devices and software (e.g., firewalls, IDS/IPS).
2. **Cryptographic Algorithm Implementation:** Ability to implement fundamental encryption, decryption, hashing, and digital signature algorithms.
3. **Threat Identification and Analysis:** Skills in recognizing various network attack techniques and analyzing their impact.
4. **Vulnerability Assessment:** Practical experience with tools and methodologies for identifying security weaknesses.
5. **Incident Response:** Understanding the process of responding to and reporting on security incidents.
6. **Problem Solving:** Applying knowledge to complex network security challenges, including design, defense, and analysis.
7. **Cyber Law Awareness:** Understanding legal and ethical implications of cybersecurity activities.

Course Code: CSE3519	**Course Title: Information Security and Management** **Type of Course: Theory Course**	**L- T-P- C**	**3**	**0**	**0**	**3**
Course Pre-requisites	CSE2502 Cryptography and Network Security					
Anti-requisites	NIL					
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	To develop a comprehensive understanding of information security by enabling students to learn security frameworks, apply risk management techniques, develop security policies and procedures, and ensure compliance with governance standards.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamental concepts of information security, including core principles and common threats. [Understand] **CO2:** Explain the core concepts of cryptography and its role in securing information. [Understand] **CO3:** Apply risk management principles in information security contexts. [Apply] **CO4:**					

	Analyze data leakage threats and propose strategies to mitigate the risk of data loss. [Analyze] **CO5:** Apply information security policies and management frameworks such as ISO 27001. [Apply]

Course Content:

Module 1	**Information Security Management:**	**14 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). 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.

Module 2	**Fundamentals of Information Security and Data Leakage**	**15 Sessions**

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.

Module 3	**Information Security Policies and Management**	**16 Sessions**

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.

Targeted Application & Tools that can be used:

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:

1. ISO 27000 family of standards: Conceptual framework for ISMS.
2. ISO/IEC 27001: The best-known standard in the family, providing requirements for an ISMS.
3. Security Information and Event Management (SIEM) tools (conceptual): For centralized logging

and security event correlation.
4. Vulnerability Scanners (conceptual): For identifying system weaknesses.
5. 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: Theory Course**	**L- T-P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2502 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	To develop a comprehensive understanding of intrusion detection and prevention by enabling students to understand intrusion detection systems, analyze network traffic for potential threats, implement prevention mechanisms, and evaluate the effectiveness of security systems.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain various types of intruders, attack vectors, and their motivations in network environments. [Understand] **CO2:** Explain and formulate intrusion detection and prevention policies for organizational security. [Understand] **CO3:** Explain the fundamental concepts of network protocol analysis and techniques for monitoring network traffic. [Understand] **CO4:** Apply protocol analyzers and Network Intrusion Detection Systems (NIDS) to detect attacks and troubleshoot network issues. [Apply] **CO5:** Analyze and evaluate network security threats and the effectiveness of intrusion detection and prevention mechanisms. [Analyze]					
Course Content:						

Module 1	Introduction to Intrusion Detection	12 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.

Module 2	Intrusion Prevention System	12 Sessions

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	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	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:

1. **Network Protocol Analysis:** Ability to capture, filter, and deeply analyze network packets to understand communication flows and identify anomalies.
2. **Intrusion Detection System (IDS) Operation:** Practical skills in installing, configuring, and managing open-source IDS tools like Snort, including rule writing and alert interpretation.
3. **Intrusion Prevention System (IPS) Concepts:** Understanding of IPS functionalities and deployment, with hands-on exposure to simulating prevention capabilities.
4. **Security Policy Definition:** Ability to define and implement security policies that govern IDS/IPS behavior and incident response.
5. **Threat Identification:** Skills in identifying various types of network attacks and attack signatures.
6. **Security Log Analysis:** Competence in analyzing IDS alerts and system logs to identify malicious activity and distinguish it from false positives.
7. **Incident Response Fundamentals:** Conceptual understanding of the incident response lifecycle, with a focus on initial detection and analysis.
8. **Security Tool Proficiency:** Hands-on experience with industry-relevant network security tools.
9. **Legal & Ethical Awareness:** Understanding the legal implications of network security monitoring and data handling.
10. **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: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1					
Course Pre-requisites	CSE2502Cryptography 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	To develop a comprehensive understanding of web application security by enabling students to understand common vulnerabilities, apply secure coding practices, implement authentication and authorization mechanisms, and analyze web application security to mitigate potential threats.					

Course Code: CSE3521	**Course Title: Principles and Practices of Web Security** **Type of Course: Lab Integrated Course**	**L- T-P- C**	**2**	**0**	**2**	**3**
Version No.	1					
Course Pre-requisites	CSE2502Cryptography 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	To develop a comprehensive understanding of web application security by enabling students to understand common vulnerabilities, apply secure coding practices, implement authentication and authorization mechanisms, and analyze web application security to mitigate potential threats.					

Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the fundamentals of web applications and input validation mechanisms. [Understand] **CO2:** Explain the significance of password management and authentication in web applications. [Understand] **CO3:** Explain the importance of session management in web applications. [Understand]					

	CO4: Apply web attack techniques to identify vulnerabilities in web applications. [Apply] **CO5:** Apply defensive strategies and secure coding practices to mitigate common web vulnerabilities. [Apply]

Course Content:

Module 1	**Introduction to Web Security Fundamentals**	**7 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**	**8 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**	**8 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**	**7 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: **30 Sessions**

Module 1: Web Security Fundamentals (Labsheets 1–4)

Week 1

Labsheet–1 (HTTP/HTTPS & Web Architecture Basics)
Level 1:
Inspect HTTP requests and responses using browser developer tools and identify headers, methods, and parameters.

Level 2:
Analyze differences between HTTP and HTTPS by capturing traffic and explaining encryption benefits.

Week 2

Labsheet–2 (Encoding Techniques in Web Security)
Level 1:
Demonstrate URL encoding and HTML encoding using sample inputs.

Level 2:
Intercept requests using a proxy tool (e.g., Burp Suite) and manipulate encoded parameters to observe application behavior.

Week 3

Labsheet–3 (Client-Side Validation & Bypass Techniques)
Level 1:
Test a web form with client-side validation and identify validation rules.

Level 2:
Bypass validation using developer tools or proxy interception and submit manipulated data.

Week 4

Labsheet–4 (Application Mapping & Threat Identification)
Level 1:
Enumerate application pages and functionalities using manual browsing.

Level 2:
Perform basic threat classification and prioritize vulnerabilities using risk analysis.

Module 2: Web Authentication Mechanisms (Labsheets 5–8)

Week 5

Labsheet–5 (Password-Based Authentication & Attacks)
Level 1:
Perform a dictionary attack on a vulnerable login system.

Level 2:
Use automated tools (e.g., Burp Intruder) to simulate brute-force attacks and analyze responses.

Week 6

Labsheet–6 (Authentication Bypass Techniques)
Level 1:
Test login with default credentials and weak password recovery methods.

Level 2:
Exploit logic flaws in authentication (parameter manipulation, header changes).

Week 7

Labsheet–7 (Secure Password Storage & Hashing)
Level 1:
Implement password hashing using SHA-256 or bcrypt.

Level 2:
Demonstrate salting and compare hashed password security against rainbow table attacks.

Week 8

Labsheet–8 (Multi-Factor Authentication)
Level 1:
Implement a basic two-factor authentication system (OTP-based).

Level 2:
Analyze strengths and limitations of multi-factor authentication in web applications.

Module 3: Session Management & Web Security Principles (Labsheets 9–12)

Week 9

Labsheet–9 (Session Management Basics)
Level 1:
Identify session tokens in cookies and analyze their properties.

Level 2:
Test session expiration and reuse of session tokens.

Week 10

Labsheet–10 (Session Attacks: Fixation & Hijacking)
Level 1:
Demonstrate session fixation using predefined session IDs.

Level 2:
Simulate session hijacking and analyze its impact.

Week 11

Labsheet–11 (Access Control Vulnerabilities)
Level 1:
Exploit IDOR by modifying request parameters.

Level 2:
Demonstrate broken function-level authorization by accessing restricted endpoints.

Week 12

Labsheet–12 (XSS & CSRF Attacks)
Level 1:
Perform reflected XSS attack using simple script injection.

Level 2:
Implement stored XSS and CSRF attack scenarios and analyze their effects.

Module 4: Web Vulnerabilities & Secure Coding (Labsheets 13–15)

Week 13

Labsheet–13 (SQL Injection & Backend Attacks)
Level 1:
Perform SQL injection to bypass authentication.

Level 2:
Execute blind SQL injection and extract data conceptually.

Week 14

Labsheet–14 (Command Injection & File Path Traversal)
Level 1:
Demonstrate command injection using simple payloads.

Level 2:
Perform directory traversal attack to access restricted files.

Week 15

Labsheet–15 (Secure Coding Practices)
Level 1:
Implement server-side input validation using whitelist approach.

Level 2:
Apply output encoding techniques to prevent XSS and test effectiveness.

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:
A**ssignment:** 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:

1. **Web Application Fundamentals & Security:** Ability to define and understand the core components and security challenges of web applications.
2. **Session Management & Web Security Principles:** Practical understanding and implementation of secure session management and other fundamental web security principles.
3. **Web Application Vulnerability Identification:** Skills in identifying, analyzing, and classifying common web application vulnerabilities (e.g., SQL Injection, XSS, CSRF, Access Control flaws).
4. **Web Exploitation Techniques:** Hands-on experience in ethically exploiting web vulnerabilities to understand their impact.
5. **Secure Coding Practices:** Ability to apply defensive coding techniques to prevent common web application attacks, especially focusing on **Input Validation** and **Output Encoding**.
6. **Security Tool Proficiency:** Competence in using industry-standard tools like web proxies (Burp Suite, OWASP ZAP), vulnerability scanners, and network reconnaissance tools.
7. **Problem Solving & Analytical Thinking:** Ability to diagnose security flaws, devise exploitation

strategies, and formulate effective countermeasures.

8. **Ethical Hacking Mindset:** Developing a responsible and ethical approach to discovering and reporting vulnerabilities.
9. **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 Course**	**L-T- P- C**	**3**	**0**	**0**	**3**
Version No.	1.0					
Course Pre-requisites	CSE2502 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	To develop a comprehensive understanding of penetration testing by enabling students to learn testing methodologies, identify system vulnerabilities, perform security assessments, and recommend effective mitigation strategies.					
Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain principles and techniques for information gathering and vulnerability detection in target systems. [Understand] **CO2:** Analyze security threats and vulnerabilities in SDN networks and web applications. [Apply] **CO3:** Apply exploitation techniques for assessing security in mobile applications and wireless networks. [Apply] **CO4:** Utilize tools such as Metasploit and Meterpreter to automate penetration testing tasks. [Apply] **CO5:** Conduct basic risk assessments based on identified vulnerabilities and their potential impact. [Apply]					
Course Content:						
Module 1	**Information Gathering, Host Discovery and Evading Techniques**					**12 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**	**12 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**	**12 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**	**9 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:

1. **Nmap:** Network scanner for information gathering and vulnerability assessment.

2. **Nessus (Conceptual/Trial):** Vulnerability scanner for identifying security weaknesses.
3. **Burp Suite (Community Edition):** Web proxy for intercepting and manipulating web traffic.
4. **OWASP ZAP:** Open-source web application security scanner.
5. **Aircrack-ng Suite (Conceptual/Lab Environment):** Wireless network auditing tools.
6. **Metasploit Framework:** Penetration testing framework for exploitation and post-exploitation.
7. **Kali Linux / Parrot OS:** Penetration testing distributions with pre-installed tools.
8. **Virtualization Software:** (e.g., VirtualBox, VMware Workstation Player) for creating isolated lab environments.
9. **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.

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/

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:

1. **Information Gathering & Reconnaissance:** Ability to effectively gather information about target systems and networks.
2. **Vulnerability Assessment:** Skills in identifying, analyzing, and classifying security weaknesses using various tools and techniques.
3. **Exploitation Techniques:** Hands-on experience in ethically exploiting vulnerabilities across different platforms (web, mobile, network, wireless).
4. **Penetration Testing Methodologies:** Understanding and applying structured approaches to penetration testing.
5. **Security Tool Proficiency:** Competence in using industry-standard penetration testing tools (e.g., Nmap, Burp Suite, Metasploit, Aircrack-ng).
6. **Risk Assessment:** Ability to evaluate the likelihood and impact of vulnerabilities and prioritize remediation efforts.
7. **Reporting & Communication:** Skills in documenting findings and communicating security risks effectively to technical and non-technical audiences.
8. **Ethical Hacking Mindset:** Developing a responsible, legal, and ethical approach to security testing.
9. **Problem Solving & Analytical Thinking:** Ability to diagnose complex security flaws, devise exploitation paths, and formulate effective countermeasures.
10. **Advanced Exploitation:** Practical knowledge of using frameworks like Metasploit for automating attacks and post-exploitation.

Course Code: **CSE3523**	**Course Title: Introduction to Fintech** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	FIN1002 Essentials of Finance					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a comprehensive understanding of financial technology by enabling students to learn fintech systems and concepts, explore digital payments and financial innovations, analyze fintech applications, and evaluate emerging trends in the financial ecosystem.					
Course	On successful completion of this course, students will be able to:					

Outcomes	**CO1:** Explain the historical evolution and development of financial technology. [Understand] **CO2:** Analyze the impact of financial technology on the financial services industry and business models. [Analyze] **CO3:** Explain the fundamental technical concepts underlying financial technology systems. [Understand] **CO4:** Apply fintech tools and technologies to solve problems in areas such as payments, lending, and digital banking. [Apply] **CO5:** Identify, interpret, and evaluate emerging technological trends in the financial services sector. [Analyze]

Course Content:

Module 1	**Introduction to FinTech**	**7 Sessions**

Topics:

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

Module 2	**FinTech Reshaping Financial Services Industry-I**	**8 Sessions**

Topics:

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

Module 3	**FinTech as disruptor empowering Financial Services Industry-II**	**8 Sessions**

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**	**7 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

List of Laboratory Tasks: **30 Sessions**

Module 1: Introduction to FinTech (Lab Sheets 1–4)

Week 1: Lab Sheet 1 (FinTech Data Analysis – Evolution Trends)

Level 1:
Write a Python program to load a CSV dataset of FinTech investments and display summary statistics (mean, median, growth rate).

Level 2:
Develop a Python program to visualize FinTech evolution trends (bar chart/line chart) across years using Matplotlib.

Week 2: Lab Sheet 2 (Financial Data Processing)

Level 1:
Write a Python program to read transaction data and filter records based on transaction type (credit/debit).

Level 2:
Create a Python program to classify transactions into categories (shopping, bills, salary) using conditional logic.

Week 3: Lab Sheet 3 (Simple FinTech Dashboard)

Level 1:
Develop a Python script to display account balance and recent transactions.

Level 2:
Build a basic dashboard using Streamlit to visualize user financial data.

Week 4: Lab Sheet 4 (API Integration – Financial Data)

Level 1:
Write a Python program to fetch real-time currency exchange rates using a public API.

Level 2:
Develop a program to convert currencies dynamically and display historical trends.

Module 2: FinTech in Payment & Lending (Lab Sheets 5–8)

Week 5: Lab Sheet 5 (Digital Wallet Simulation)

Level 1:
Develop a Python program to simulate a digital wallet with add money and pay features.

Level 2:
Extend the wallet system to include transaction history and balance validation.

Week 6: Lab Sheet 6 (KYC Verification System)

Level 1:
Write a Python program to validate user details (email, phone, ID format).

Level 2:
Develop a mini KYC system that stores verified users in a database (SQLite).

Week 7: Lab Sheet 7 (Loan Eligibility System)

Level 1:
Write a Python program to check loan eligibility based on income and credit score.

Level 2:

Implement a scoring model to classify applicants into low, medium, high risk.

Week 8: Lab Sheet 8 (P2P Lending Platform Simulation)

Level 1:
Develop a Python program to match lenders and borrowers.

Level 2:
Extend it to calculate interest, repayment schedule, and risk scoring.

Module 3: Wealth & Insurance FinTech (Lab Sheets 9–12)

Week 9: Lab Sheet 9 (Investment Portfolio Tracker)

Level 1:
Write a Python program to track stock investments and calculate total portfolio value.

Level 2:
Enhance the program to calculate profit/loss and visualize portfolio distribution.

Week 10: Lab Sheet 10 (Robo-Advisory System)

Level 1:
Develop a rule-based Python system to suggest investments based on risk profile.

Level 2:
Implement a basic ML model (e.g., decision tree) to recommend investment strategies.

Week 11: Lab Sheet 11 (Insurance Premium Calculator)

Level 1:
Write a Python program to calculate insurance premium based on age and health factors.

Level 2:
Extend the system to simulate claim processing and approval logic.

Week 12: Lab Sheet 12 (Fraud Detection System)

Level 1:
Write a Python program to detect suspicious transactions using threshold rules.

Level 2:
Implement anomaly detection using a machine learning algorithm (e.g., Isolation Forest).

Module 4: FinTech Technologies & Innovations (Lab Sheets 13–15)

Week 13: Lab Sheet 13 (Blockchain Simulation)

Level 1:
Write a Python program to create a simple blockchain with blocks and hashing.

Level 2:
Extend the blockchain to validate transactions and prevent tampering.

Week 14: Lab Sheet 14 (Big Data Analytics for Finance)

Level 1:
Use Pandas to process large financial datasets and compute insights.

Level 2:
Implement predictive analytics using regression to forecast financial trends.

Week 15: Lab Sheet 15 (FinTech Web Application)

Level 1:
Develop a simple web application (Flask/Streamlit) for transaction management.

Level 2:
Extend the application to include authentication, API integration, and analytics dashboard.

Project work/Assignment:

1. **Assignment 1 on (Module 1 and Module 2)**
2. **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

Course Code: CSE3524	**Course Title: Banking Technology** **Type of Course: Lab Integrated Course**	**L-T-P-C**	**2**	**0**	**2**	**3**
Version No.	1.0					
Course Pre-requisites	FIN1002 Essentials of Finance					
Anti-requisites	NIL					
Course Description	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.					
Course Objective	To develop a comprehensive understanding of modern banking by enabling students to understand banking systems and technologies, analyze digital banking operations, explore payment systems and their security aspects, and evaluate emerging innovations in the banking sector.					

Course Outcomes	On successful completion of this course, students will be able to: **CO1:** Explain the evolution of banking technology and its impact on modern banking operations. [Understand] **CO2:** Explain the role of delivery channels such as ATMs, mobile banking, and UPI in digital banking systems. [Understand] **CO3:** Evaluate the effectiveness of risk management, treasury operations, and data center management in banking systems. [Evaluate] **CO4:** Analyze the functioning of centralized banking and payment systems such as RTGS, NEFT, and SFMS. [Analyze] **CO5:** Explain blockchain and cryptocurrency concepts and analyze recent core banking software systems. [Analyze]

Course Content:

Module 1	**Branch Operation and Core Banking**	**Assignment**	**-**	**6 Sessions**
Topics: 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.				
Module 2	**Delivery Channels**	**Assignment**	**-**	**7 Sessions**
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**	**-**	**6 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**	**-**	**6 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	**Assignment** -	**5 Sessions**

Topics:

Techniques Block Chain and Bit-coin – Crypto currency Analysis of Recent Core Banking Software-Case study.

List of Laboratory Tasks: **30 Sessions**

Module 1: Branch Operation & Core Banking (Lab Sheets 1–3)

Week 1: Lab Sheet 1 (Core Banking System – Basic Operations)

Level 1:

Write a Python program to simulate basic banking operations such as account creation, deposit, and withdrawal.

Level 2:

Extend the program to maintain multiple customer accounts with unique IDs and transaction history.

Week 2: Lab Sheet 2 (Centralized Banking Simulation)

Level 1:

Develop a Python program to simulate centralized banking where multiple branches access a common database.

Level 2:

Implement concurrency handling (using threads) to simulate multiple users accessing the system simultaneously.

Week 3: Lab Sheet 3 (Branch Transaction Management)

Level 1:

Write a program to record daily branch transactions and compute total deposits and withdrawals.

Level 2:

Develop a reporting system to generate end-of-day summaries for each branch.

Module 2: Delivery Channels (Lab Sheets 4–6)

Week 4: Lab Sheet 4 (ATM Simulation System)

Level 1:

Develop a Python program to simulate ATM operations (PIN validation, withdrawal, balance inquiry).

Level 2:

Extend the ATM system to include mini statement generation and transaction limits.

Week 5: Lab Sheet 5 (Internet/Mobile Banking System)

Level 1:

Create a basic web interface using Flask to display account details.

Level 2:

Extend the application to allow fund transfer between accounts with validation.

Week 6: Lab Sheet 6 (UPI Payment Simulation)

Level 1:

Write a Python program to simulate UPI payments using virtual IDs.

Level 2:

Enhance the system to include QR code-based payments and transaction logs.

Module 3: Back Office Operations (Lab Sheets 7–9)

Week 7: Lab Sheet 7 (Inter-Branch Reconciliation System)

Level 1:

Write a Python program to compare transactions between two branches and identify mismatches.

Level 2:

Develop an automated reconciliation system that resolves discrepancies.

Week 8: Lab Sheet 8 (Risk Management System)

Level 1:

Develop a program to calculate risk score based on transaction patterns.

Level 2:

Implement anomaly detection for suspicious transactions using basic ML techniques.

Week 9: Lab Sheet 9 (CRM System for Banking)

Level 1:

Create a program to store and manage customer details.

Level 2:

Develop a CRM dashboard showing customer segmentation and interaction history.

Module 4: Interbank Payment Systems (Lab Sheets 10–12)

Week 10: Lab Sheet 10 (NEFT/RTGS Simulation)

Level 1:

Write a Python program to simulate NEFT transactions with batch processing.

Level 2:

Extend the system to simulate RTGS with real-time fund transfer and priority handling.

Week 11: Lab Sheet 11 (Secure Messaging – SFMS Simulation)

Level 1:

Develop a program to send encrypted messages between banks.

Level 2:

Implement secure message authentication using hashing techniques.

Week 12: Lab Sheet 12 (Payment Gateway Integration)

Level 1:

Create a simple payment gateway interface to process payments.

Level 2:

Extend it to include payment status tracking and failure handling.

Module 5: Contemporary Banking Technologies (Lab Sheets 13–15)

Week 13: Lab Sheet 13 (Blockchain for Banking)

Level 1:

Write a Python program to create a simple blockchain for recording transactions.

Level 2:

Extend the blockchain to validate transactions and ensure immutability.

Week 14: Lab Sheet 14 (Cryptocurrency Transaction System)

Level 1:

Develop a program to simulate cryptocurrency wallet transactions.

Level 2:

Implement digital signature verification for secure transactions.

Week 15: Lab Sheet 15 (Core Banking Software Mini Project)

Level 1:

Develop a mini banking application integrating account management and transactions.

Level 2:

Extend the application with authentication, reporting dashboard, and security features.

Project work/Assignment:

Assignment 1 on (Module 1 and Module 2)

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/

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: **ISE2510**	**Course Title- Ethics of AI Type of Course: Theory**	**L- T- P-C**	**2**	**0**	**0**	**2**
Version No.	1.0					
Course Pre-requisites	CSE2264					
Anti-requisites	NIL					
Course Description	This course explores the ethical, legal, and societal implications of Artificial Intelligence (AI) in modern society. As AI technologies increasingly influence decision-making, behavior, and governance, it is critical to understand the responsibilities of developers, users, and policymakers in ensuring ethical outcomes. Students will examine key topics such as algorithmic bias, fairness, transparency, data privacy, accountability, and the role of regulation in guiding AI development. Through philosophical inquiry, real-world case studies, and critical analysis, the course equips students with the knowledge and tools to assess ethical challenges in AI systems and contribute to the creation of responsible and human-centered AI solutions.					
Course Outcomes	At the end of the course, the student will be able to: **CO1:** Explain the fundamental ethical theories and their relevance to Artificial Intelligence systems. [Understand]					

	CO2: Identify and analyze ethical issues such as bias, fairness, transparency, and accountability in AI systems. [Analyze]	
	CO3: Evaluate the ethical implications of AI on privacy, surveillance, and data governance. [Evaluate]	
	CO4: Examine real-world AI applications through case studies and apply ethical reasoning to assess their societal impact. [Analyze]	
	CO5: Propose frameworks and best practices for developing responsible, transparent, and human-centered AI systems. [Create]	
Course Content:		
Module 1	**Foundations of AI Ethics**	**8 Sessions**
Topics: Topics : Introduction to Ethics: Moral theories and frameworks,Relevance of ethics in AI systems,Overview of AI capabilities and impact,Stakeholders in AI development and deployment, Case studies: Early ethical dilemmas in AI applications		
Module 2	**Bias, Fairness, and Accountability**	**8 Sessions**
Topics: Understanding algorithmic bias and discrimination, Fairness in machine learning models, Accountability and transparency in AI systems, Auditing AI for ethical compliance,Case studies: Finance, and law enforcement		
Module 3	**Privacy, Surveillance, and Data Ethics**	**7 Sessions**
Topics: Data privacy and protection principles, Surveillance technologies and ethical concerns, Informed consent and user autonomy, Data governance and ethical data usage. Case studies: Social media platforms, facial recognition		
Module-4	**Responsible AI and Governance**	**7 Sessions**
Topics: Explainability and interpretability in AI, Transparency, Trust and accountability in AI, systems Regulation and legal frameworks (e.g., EU AI Act, IEEE standards),Human-in-the-loop and value-sensitive design,Future of ethical AI: Aligning AI with human values,Case studies: Autonomous systems, healthcare AI		
Project work/Assignment:		

Assignment 1 on (Module 1 and Module 2)

Assignment 2 on (Module 3 and Module 4)

REFERENCE MATERIALS:

TEXTBOOKS

1. Virginia Dignum – Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way, Springer, 2019.
2. Mark Coeckelbergh – AI Ethics, MIT Press, 2020.

REFERENCES

1. Wendell Wallach and Colin Allen – Moral Machines: Teaching Robots Right From Wrong, Oxford University Press, 2008.
2. Shannon Vallor – Technology and the Virtues: A Philosophical Guide to a Future Worth Wanting, Oxford University Press, 2016.3.
3. Cathy O'Neil – Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, Crown Publishing, 2016.
4. Nick Bostrom – Superintelligence: Paths, Dangers, Strategies, Oxford University Press, 2014

Course Code: ISE2508	Course Title: Natural Language Processing Type of Course: PCC Theory	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 basics of Natural Language Processing methods with specific emphasis on modern applications. The course will teach students different concepts of natural language processing, such as word representations, text representations, part-of-speech tagging, word sense disambiguation, parsing, etc. Topics: Word representations, Part-of-Speech tagging, chunking, parsing, text classification, sentiment analysis, named entity recognition, and machine translation..					
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: Define different problems related to natural language processing. [Knowledge] Discuss using NLP techniques for different applications. [Comprehension] Propose solutions for a particular NLP problem using different machine learning					

	and deep learning techniques. [Application] Learn to use different NLP tools and packages. [Application]		
Course Content:			
Module 1	Introduction to Natural Language Processing		
Definition of Natural Language Processing; Description of various NLP tasks; Sentence and word boundary detection; Introduction to word representation, PoS tagging, Chunking and Parsing, and text classification; Introduction to NLP applications like Sentiment Analysis, Named Entity Recognition, and Machine Translation			
Module 2	Word and Text Representation		
Introduction to Word Embeddings; Creation of word embeddings using Skipgram; Using word embeddings like GloVe / fastText; Cross-lingual word embeddings (Eg. MUSE); Pre-trained monolingual and multilingual language models. Text Representations Using BoW, feature-based, Kernel, embedding-based representations.			
Module 3	Part-of-Speech Tagging, Chunking and Parsing		
Sequence Labeling and Hidden Markov Model; Viterbi Algorithm; Part-of-Speech Tagging; Using NLTK and Spacy for PoS Tagging; Building a PoS Tagger; Chunking and Constituency Parsing; Using Parser from NLTK.			
Module 4	NLP Applications		
Lexical Resource Creation – Creation and evaluation. Agreement metrics Sentiment Analysis – Definitions, Challenges (Sarcasm, Thwarting, etc.) Named-Entity Recognition – Definition, Relationship between NER and PoS tagging Machine Translation – Definition, Challenges, Approaches and Paradigms, Evaluation Techniques.			
Targeted Application & Tools that can be used: Google Colab NLTK Huggingface Transformers			
Project work/Assignment:			
Group project on some NLP Task like text classification, 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, 2023 (3rd Edition Draft). Aditya Joshi, Pushpak Bhattacharyya. “Natural Language Processing”, Wiley Publication, 2023 (1st Edition).
References: 1. Chris Manning and Hinrich Schütze. “Foundations of Statistical Natural Language Processing”, MIT Press. Cambridge, MA. 1999. Weblinks: NPTEL online course: https://nptel.ac.in/courses/106106211 Latest edition of Text Book: https://web.stanford.edu/~jurafsky/slp3/

Course Code:ISE2509	**Course Title:** Natural Language Processing Lab Course Type: PCC Lab	L-T-P-C	0	0	2	1
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces the basics of Natural Language Processing methods with specific emphasis on modern applications. The course will teach students different concepts of natural language processing, such as word representations, text representations, part-of-speech tagging, word sense disambiguation, parsing, etc. Topics: Word representations, Part-of-Speech tagging, chunking, parsing, text classification, sentiment analysis, named entity recognition, and machine translation.					
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: **CO1:** Process data from text files. [Apply] **CO2:** Use NLP tools like NLTK and Spacy to extract information from the text files. [Apply] **CO3:** Build systems using supervised learning such as Naïve Bayes Classifier, Logistic Regression, etc. to solve NLP tasks such as analogy completion, language identification, and sentiment analysis. [Apply] **CO4:** Implement a Hidden Markov Model (HMM) to solve a sequence labelling task such as Part-of-Speech Tagging (PoS Tagging), Named Entity Recognition (NER), etc. **CO5:** Use Huggingface Transformers and MarianMT Model to perform Machine					

	Translation and evaluate the performance. [Application]

List of Laboratory Tasks:

Introduction to Using Word Representations and NLP Tools

Complex Word Identification

Sentiment Analysis and Named Entity Recognition

Lexical Simplification

Cross-Lingual NLP

Extracting PoS features

Building PoS Tagger

Building HMM for Title Case Conversion

Machine Translation Using Transformers

Lexical Candidate Generation and Ranking Using Pre-trained Language Models

Targeted Application & Tools that can be used:

Google Colab

NLTK

Huggingface Transformers

Project work/Assignment:

Group project on some NLP Task like text classification, 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, 2023 (3rd Edition Draft).

Aditya Joshi, Pushpak Bhattacharyya. “Natural Language Processing”, Wiley Publication, 2023 (1st Edition).

References:

1. Chris Manning and Hinrich Schütze. “Foundations of Statistical Natural Language Processing”, MIT Press. Cambridge, MA. 1999.

Weblinks:

NPTEL online course: https://nptel.ac.in/courses/106106211

Latest edition of Text Book: https://web.stanford.edu/~jurafsky/slp3/

Course Code: EEE1251	**Course Title: Basics of Electrical and Electronics Engineering Laboratory** **Type of Course: ESC - Laboratory**	**L-T-P-C**	0	0	2	1
Version No.	**1.0**					
Course Pre-requisites	**Nil**					
Anti-requisites	**Nil**					
Course 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.					
Course 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.					
Basic skill sets required for the laboratory	**The students shall be able to develop:** 14) An attitude of enquiry. 15) Confidence and ability to tackle new problems. 16) Ability to interpret events and results. 17) Ability to work as a leader and as a member of team. 18) Assess errors and eliminate them. 19) Observe and measure physical phenomenon. 20) Write Reports. 21) Select suitable equipment, instruments and materials. 22) Locate faults in systems. 23) Skills for setting and handling equipment. 24) The ability to follow standard test procedures. 25) An awareness of the need to observe safety precautions. 26) To judge magnitudes without actual measurement.					
Course Outcomes	On successful completion of the course, the students shall be able to: 1. **Demonstrate** the behavior of passive elements and fundamental electrical circuit laws 2. **Determine** the power factor, impedance, and phase relationship of R and RL loads under AC supply conditions. 3. **Develop** electrical circuits and three-phase supply systems using MATLAB/Simulink to study resistance, voltage, and current characteristics. 4. **Analyze** the performance characteristics of electrical machines, including single-phase transformers and three-phase induction motors, under different operating conditions. 5. **Examine** the output characteristics and performance parameters of half-wave and full-wave rectifiers.					

	6. **Illustrate** the performance characteristics of Zener diode and MOSFET under different operating conditions.
Course Content:	List of Laboratory Tasks: **Experiment No 1:**Familiarization of passive elements. Level 1: Identify and understand the function and ratings of basic passive elements. Level 2: Study the behavior of passive elements in simple electrical circuits by measuring voltage, current, and impedance. **Experiment No 2:**Experimental Verification of Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL). Level 1: Study and Verify KVL and KCL for the given electrical Circuit. **Experiment No 3:**Determination of Power Factor, and Impedance for R and RL Loads. Level 1: Determine the power factor and impedance of R and RL loads under steady-state AC supply conditions. Level 2: Analyze the effect of inductance on power factor, impedance, and phase angle for RL loads under varying load conditions. **Experiment No 4:**Modeling and Simulation of effective resistances in different circuit connections Using Simulink. Level 1: Model and simulate series and parallel resistor circuits in Simulink and determine their effective resistance. Level 2:Analyze the effect of different resistor combinations on total resistance, current distribution, and voltage drop. **Experiment No 5:**Simulation and analysis of line and phase voltages in Star-connected and Delta- connected supply systems. Level 1: Model and simulate star-connected and delta-connected three-phase supply systems and observe line and phase voltages. Level 2: Analyze and compare the relationship between line and phase voltages in star and delta connections. **Experiment6:** Determination of transformation ratio of a Single-Phase Transformer Under different tapping conditions. Level 1: Determine the transformation ratio of a single-phase transformer under no-load condition for various tap settings. Level 2: Analyze the effect of different tapping conditions on secondary voltage, regulation, and transformer performance. **Experiment No. 7:** Study of the speed-torque characteristics of a Three-Phase Induction Motor. Level 1: Study and analyze the speed–torque characteristics of a three-phase induction motor. Level 2: Evaluate the effect of supply voltage and current on motor performance parameters such as slip, and torque. **Experiment 8:** Performance Analysis of Half-Wave and Full-Wave Rectifier Circuits Using Simulink. Level 1: Model and simulate half-wave and full-wave rectifier circuits in Simulink and analyze their output waveforms. Level 2: Evaluate and compare rectifier performance parameters such as ripple factor, efficiency, and output voltage. **Experiment 9:** Study of the characteristics of a Zener Diode. Level 1: Study the forward and reverse V–I characteristics of a Zener diode using DC analysis. Level 2: Analyze the Zener breakdown characteristics and determine the Zener voltage and regulation behavior.

	Experiment 10: Study of the Output and Transfer Characteristics of a MOSFET. Level 1: Study, analyze and implement the output characteristics of a MOSFET under different gate-source voltage conditions and observe their results. Level 2: Analyze the transfer characteristics of a MOSFET and evaluate key parameters such as threshold voltage and transconductance.

Targeted 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: MATLAB SIMULINK

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

Course Material

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

Text Book:

1. D. P. Kothari and I. J. Nagrath, Basic Electrical Engineering, 4th Edition, McGraw Hill Education, New Delhi, 2019.
2. B. L. Theraja and A. K. Theraja, A Textbook of Electrical Technology, Volume-I: Basic Electrical Engineering in SI Units, 24th Edition, S. Chand Publishing, New Delhi, 2024.

References

1. John Hiley, Keith Brown and Ian McKenzie-Smith, Hughes Electrical and Electronic Technology, 12th Edition, Pearson Education, 2016.
2. S. K. Bhattacharya and D. Devaraj, Basic Electrical and Electronics Engineering, 2nd Edition, Pearson Education India, 2017.
3. T. K. Nagsarkar and M. S. Sukhija, Principles of Basic Electrical Engineering, 1st Edition, Oxford University Press, 2018.
4. Robert L. Boylestad and Louis Nashelsky, Electronic Devices and Circuit Theory, 12th Edition, Pearson Education, 2019.
5. A. K. Maini and Varsha Agrawal, Electronic Devices and Circuits, 2nd Edition, Wiley India, 2019.
6. Adel S. Sedra and Kenneth C. Smith, Microelectronic Circuits, 8th Edition, Oxford University Press, 2019.

Online Resources:

1. https://presidencyuniversity.linways.com
2. https://www.digimat.in/nptel/courses/video/108105112/L01 "Fundamentals of Electrical Engineering - Basic Concepts, Examples"
3. 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
4. NPTEL lectures on "Rectifiers", by Prof. G. Bhuvaneswari, IIT Delhi https://nptel.ac.in/courses/108102145
5. NPTEL Lectures on "Electrical Machines II", by Prof. G. Sridhara Rao, IIT Madras. https://nptel.ac.in/courses/108106072

Topics relevant to "EMPLOYABILITY SKILLS": 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.

Topics relevant to "HUMAN VALUES & PROFESSIONAL ETHICS": Standard test methods, safety procedure.

Course Code: MAT1211	**Course Title: Linear Algebra, Differential Equations and Statistical Methods** **Type of Course:1] School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces fundamental mathematical concepts from linear algebra, differential calculus, ordinary differential equations and statistics. The content includes matrices, systems of linear equations, eigenvalues and eigenvectors, functions of several variables, differential equations, correlation and regression analysis, and curve fitting techniques. The course combines theoretical developments with illustrative applications drawn from scientific and engineering settings. The course emphasizes the application of these mathematical tools in solving problems in science and engineering. It extends student's problem-solving abilities through analytical and computational approaches, with focus on both theoretical understanding and applications.					
Course Objective	The objective of this course is to provide students with a strong mathematical foundation required for engineering and scientific studies. The course introduces fundamental concepts from linear algebra, differential calculus, ordinary differential equations, and statistical methods that are widely used in modeling, analysis and problem solving. It aims to familiarize students with mathematical techniques for representing engineering systems, studying relationships among variables, describing dynamic processes and interpreting data. The course serves as a foundation to prepare students for advanced engineering subjects that rely on mathematical modeling, computation and analytical methods.					
Course Out Comes	On successful completion of the course the students shall be able to: **CO1:**Use linear algebra concepts to evaluate properties of matrices and quadratic forms. (L3) **CO2:** Apply matrix-based methods to solve systems of linear equations. (L3) **CO3:** Use fundamental theorems and analytical techniques of differential calculus to single variable functions. (L3) **CO4:** Apply the concepts of partial derivatives to multivariable functions. (L3) **CO5:** Solve first order ordinary differential equations using analytical methods.(L3) **CO6:**Employ statistical techniques for data analysis and to interpret relationshipsbetween variables. (L3)					
Course Content:						
Module 1	**Fundamentals of Linear Algebra**					**(15L+5T=20 Sessions)**
Matrices, echelon forms, rank of matrix, Systems of linear equations: Consistency, solution to system of linear homogeneous and non-homogeneous equations- Gauss elimination method, Gauss-Jordan method (Self-study), LU decomposition method, Network analysis using linear systems, Design of traffic patterns, Electrical circuits. Eigenvalue and eigenvector, diagonalization of symmetric matrices, quadratic forms.						
Module 2	**Differential Calculus**					**(15L+5T=20 Sessions)**
Introduction, limit, continuity, Rolle's theorem, mean value theorems, Indeterminate forms and L'Hôpital's rule. Partial derivatives, Euler's theorem, total derivative, Jacobians, Taylor's and Maclaurin's theorems, maxima, minima and saddle points, method of Lagrange multipliers, Current Growth in an RL Circuit, Velocity Estimation, Fuel consumption optimization, Cost function optimization.						
Module 3	**Ordinary Differential Equations**					**(6L+2T=8 Sessions)**
Definition, types of differential equations-variable separable, homogeneous, exact, linear and Bernoulli's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type, Newton's law of cooling, Current in RL circuit.						
Module 4	**Statistical Methods and Curve Fitting**					**(9L+3T=12 Sessions)**

Measures of central tendency, moments, skewness, and kurtosis. Correlation - Karl Pearson's coefficient of correlation, rank correlation, multiple correlation. Regression analysis - lines of regression, multiple regression. Principle of least square, curve fitting- straight line, parabola and exponential curves, Stress and strain curve fitting, Feature selection and Prediction models.

Targeted Applications &Tools that can be used:

- Linear Algebra is applied to solve systems of equations in circuits, traffic flow, and network analysis, with eigenvalues and quadratic forms aiding stability and optimization studies.
- Differential Calculus supports process speed analysis, velocity estimation, and optimization techniques such as minimizing fuel consumption and cost functions.
- Differential Equations model dynamic behaviours like current in RL circuits, heat transfer in materials, and growth or decay in environmental and nuclear engineering.
- Statistical Methods and Curve Fitting enable effective data analysis through correlation, regression, and curve fitting, supporting predictive modeling and interpretation of experimental data such as stress-strain relationships.

Software: MATLAB, Python, R Programming, Mathematica, Scilab, Octave

Assignment:

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. B. S. Grewal, *Higher Engineering Mathematics*, 44th ed. New Delhi, India: Khanna Publishers, 2017.
2. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2018.

References:

1. P. V. O'Neil, *Advanced Engineering Mathematics*, 8th ed. Boston, MA, USA: Cengage Learning, 2018.
2. R. I. Miller and J. E. Freund, *Probability and Statistics for Engineers*, 9th ed. Noida, India: Pearson Education Ltd., 2017.
3. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
4. V. Henner, T. Belozerova, and M. Khenner, *Ordinary and Partial Differential Equations*. Boca Raton, FL, USA: CRC Press, 2013.

E-resources/ Web links:

1. https://nptel.ac.in/courses/111105121
2. https://nptel.ac.in/courses/111106146
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_106839
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_61605
5. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_134719
6. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=TEXTBOOK_LIBRARY01_06082022_174

Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of differential calculus, differential equations and statistical methods 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.

Course Code: MAT2311	**Course Title: Vector Calculus, Numerical Methods and Sampling Distributions** **[Type of Course:1] School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces the fundamental concepts and methods of ordinary differential equations, numerical techniques, and statistical inference used in engineering and scientific applications. The course covers analytical methods for solving differential equations, numerical methods for solving linear and nonlinear equations, interpolation, numerical differentiation and integration, and numerical solutions of ordinary differential equations. It also introduces sampling distributions and hypothesis testing for statistical decision-making. The course enables students to apply mathematical and statistical techniques to solve computational and real-world engineering problems.					
Course Objective	The objectives of this course are to develop the ability to solve ordinary differential equations arising in scientific and engineering applications. Introduce numerical techniques for solving systems of linear equations and nonlinear algebraic equations. Apply interpolation, differentiation, and integration methods for approximating mathematical functions and evaluating definite integrals. Employ numerical methods for obtaining approximate solutions of ordinary differential equations. Apply concepts of sampling distributions and statistical inference to estimate population parameters and support decision-making. Apply hypothesis testing techniques to real-world problems involving means, proportions, variances, and associations among variables.					
Course Outcomes	After successful completion of the course, students will be able to: **CO1:** Employ vector differential operators to solve problems involving scalar and vector fields and classify them as solenoidal or irrotational (L3) **CO2:** Apply vector integration techniques and integral theorems to a solve engineering problems. (L3) **CO3:** Use numerical methods to solve systems of linear and nonlinear equations. (L3) **CO4:** Implement numerical techniques for interpolation, differentiation, integration, and ordinary differential equations. (L3) **CO5:** Utilize sampling distributions and statistical inference techniques to solve statistical problems. (L3) **CO6:** Perform hypothesis testing for statistical decision-making. (L3)					
Course Content:						
Module 1	**Vector Calculus**					**(14L+5T=19 Sessions)**
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, Surface and Volume Integrals, Green's Theorem in the plane, Divergence Theorem of Gauss, Stokes's Theorem. Problems on Image Edge Detection, Robot Path Planning and 3D Surface Modelling						
Module 2	**Solution of Linear Systems and non-linear of Equation**					**(9L+3T=12 Sessions)**
Introduction, Jacobi iteration method, Gauss-Seidel iteration method. Solution of Algebraic and Transcendental Equations: Bisection method, Regula-Falsi method, Newton-Raphson method, secant method, fixed point iteration method. Page Ranking, Network Analysis, Circuit Simulation, Root Finding, Load Balancing, and Computational Optimization Problems.						
Module 3	**Numerical Interpolations and Differential Equations**					**(10L+3T=13 Sessions)**

Newton's method, Divided difference method, Lagrange's method, Numerical differentiation, Numerical integration: Trapezoidal rule, Simpson's one-third rule, Simpson's three-eighth rule, Gaussian quadrature rule. Solution of ordinary differential equations: Taylor series method, modified Euler's method, Runge-Kutta method for 4th order. Problems on Signal Reconstruction, Sensor Data Interpolation and Trajectory Prediction.

Module 4	**Sampling Distributions**	**(12L+4T=16 Sessions)**

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, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes. A/B Testing, Fraud Detection, Spam Classification Evaluation, and Network Traffic Analysis.

Targeted Applications & Mathematical Tools:

- Analysis of fluid flow in open channels and pipelines, groundwater movement, stress and strain fields, terrain modelling, and 3D surface representation in civil engineering projects.
- Structural analysis, load distribution in trusses and frames, hydraulic network analysis, foundation design computations, and optimization problems in transportation and water resource systems.
- Estimation of terrain elevations, survey data interpolation, rainfall-runoff modelling, settlement prediction, traffic flow analysis, and numerical solutions of engineering differential equations.
- Quality control of construction materials, reliability analysis, traffic studies, environmental data analysis, hydrological investigations, and hypothesis testing in engineering experiments.
- Analysis of vector fields in fluid flow, electromagnetics, and heat transfer; image edge detection, robot path planning, and 3D surface modelling using gradients, divergence, curl, and integral theorems.

Software: MATLAB, Python, Microsoft Excel Solver, LINDO/LINGO, AutoCAD, ETABS, Scilab.

Assignment:

Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor.

Text Book

1. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, 7th ed. New Delhi, India: New Age International Publishers, 2019.
2. B. S. Grewal, *Higher Engineering Mathematics*, 45th ed. New Delhi, India: Khanna Publishers, 2022.
3. S. C. Gupta and V. K. Kapoor, *Fundamentals of Mathematical Statistics*, 12th ed. New Delhi, India: Sultan Chand & Sons, 2020.

References:

1. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, 7th ed. New Delhi, India: New Age International Publishers, 2019.
2. S. S. Sastry, *Introductory Methods of Numerical Analysis*, 5th ed. New Delhi, India: PHI Learning Pvt. Ltd., 2012.
3. J. Kiusalaas, *Numerical Methods in Engineering with Python 3*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2025.
4. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 8th ed. Hoboken, NJ, USA: John Wiley & Sons, 2023.
5. D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications*, 7th ed. Belmont, CA, USA: Cengage Learning, 2008.
6. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
7. R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*, 11th ed. Boston, MA, USA: Cengage Learning, 2021.
8. S. C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2018.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE

BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_53751

2. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=TEXTBOOK_LIBRARY01_06082022_117
3. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7313
4. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_228965
5. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_8838
6. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=SPRINGER_OPEN_ACCESS_BOOKS_10022025_1861
7. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7312

Topics relevant to SKILL DEVELOPMENT: Identification and solution of differential equations, application of numerical methods for algebraic and differential equations, implementation of interpolation and numerical integration techniques, statistical data analysis and hypothesis testing, and use of Python/MATLAB for engineering problem solving.

Course Code: MAT2321	**Course Title:Discrete Mathematical Structures** **Type of Course:1] School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces the fundamental concepts of Discrete Mathematics, providing the mathematical foundations essential for computer science and related disciplines. It covers sets, relations, functions, closures, equivalence relations, and primitive recursive functions, followed by partially ordered sets, lattices, and Boolean algebras. The course also explores graph theory, including graph types, isomorphism, Euler and Hamiltonian paths, circuits, and graph coloring. In addition, students study algebraic structures such as semigroups, monoids, and groups, residue arithmetic, and group codes. Emphasis is placed on developing logical reasoning, mathematical modeling, and problem-solving skills applicable to computing, information systems, and network analysis.						
Course Objective	The primary objective of this course is to provide students with a strong foundation in discrete mathematical structures and logical reasoning that underpin computer science and related disciplines. It aims to develop the ability to construct mathematical arguments, analyze discrete problems, and apply formal methods to problem solving. The course introduces fundamental concepts of logic, sets, relations, functions, Boolean algebra, graph theory, and algebraic structures such as semigroups, monoids, and groups, while also exploring formal languages and grammar. Through these topics, students will acquire the analytical and mathematical skills necessary for applications in algorithm design, digital systems, coding theory, cryptography, and other areas of computing.						
Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Apply propositional and predicate logic, logical equivalences, normal forms, inference rules, and proof techniques to analyze arguments and solve problems in mathematical reasoning and computer science. (L3) **CO2:** Apply the concepts of sets, relations, functions, closures, equivalence relations, and primitive recursive functions to model and solve problems in discrete mathematics and computer science. (L3) **CO3:** Employ partially ordered sets, lattices, Boolean algebras, and Boolean functions, and apply their properties to solve problems in discrete mathematical structures and logical systems. (L3) **CO4:** Apply the concepts of lattices, Boolean algebras, and Boolean functions to solve logical and computational problems. (L3) **CO5:** Apply graph theoretic concepts and techniques to solve problems in networks, connectivity, and optimization applications. (L3) **CO6:** Employ group theory and residue arithmetic techniques in coding, computation, and information processing. (L3)						
Course Content:						
Module 1	**Fundamentals of Logic**					**(11L+4T=14 Sessions)**
Propositional Logic, Propositional Logic, Equivalences, Normal forms, Inference rules, Introduction to Proofs, Conversion to clausal form, Predicate calculus, The Statement function, Inference theory of the Predicate Calculus. Compiler optimization and program correctness verification, Cybersecurity and cryptographic protocol verification.						
Module 2	**Relations and Functions**					**(10L+3T=13 Sessions)**
Sets and set-operations, Relations, Functions, Properties of Relations & Representations of Relation, closure of different types of relations, equivalence relations, primitive recursive function. Database Management Systems, Artificial Intelligence and Machine Learning.						
Module 3	**Boolean Algebra and Graph Theory**					**(13L+4T=17 Sessions)**
Partially Ordered Sets, Lattices, Boolean Algebras, Functions of Boolean Algebras, Boolean Functions as Boolean Polynomials. **Computer Networks:** Routing and switching logic. **Robotics:** Decision-making						

using Boolean logic.
Graph Theory-Basic Concept of Graph Theory, Types of Graphs, Isomorphism, Euler Paths and Circuits, Hamiltonian Paths and Circuits, Colouring of Graph. Social Networks (Facebook, Instagram, LinkedIn).and **GPS & Navigation:** Shortest path and route optimization.

Module 4	**Groups**	**(11L+4T=15 Sessions)**

Semi Groups, Monoids, Groups, The applications of Residue Arithmetic to Computers, Group codes. Functional programming and distributed systems. Compiler design and programming languages.

Targeted Application & Tools that can be used:

- Compiler optimization, program correctness verification, cybersecurity protocol verification, automated reasoning, formal software verification.
- Database Management Systems (DBMS), Artificial Intelligence (AI), Machine Learning (ML), knowledge representation, data modeling and query processing.
- Digital circuit design, computer networks (routing and switching), robotics decision-making systems, logic controller design, social network analysis, GPS and navigation systems, route optimization, communication networks, and recommendation systems.
- Cryptography, error-correcting codes, compiler design, distributed systems, residue arithmetic in computer architecture.

Software: MATLAB, Lean 4, Mathematica, Maple, Python, GeoGebra, Octave, R, Scilab.

Assignment:

Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor. Additionally, experiential learning activities will be conducted using Lean 4, enabling students to develop formal proofs, verify logical propositions, and gain hands-on experience in theorem proving and formal verification.

Text Book

1. K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2019.
2. F. Harary, *Graph Theory*. Reading, MA, USA: Addison-Wesley Publishing Company, 1969.

References:

1. S. Epp, *Discrete Mathematics with Applications*, 5th ed. Boston, MA, USA: Cengage Learning, 2019.
2. K. A. Ross and C. R. B. Wright, *Discrete Mathematics*, 6th ed. New Delhi, India: Pearson Education, 2018.
3. L. Mott, A. Kandel, and T. P. Baker, *Discrete Mathematics for Computer Scientists and Mathematicians*, 3rd ed. New Delhi, India: PHI Learning, 2016.

E-resources/ Web links:

6. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=TEXTBOOK_LIBRARY01_06082022_289
7. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_8919
8. https://nptel.ac.in/courses/106108227
9. https://nptel.ac.in/courses/111106102
10. https://nptel.ac.in/courses/111106113

Topics relevant to SKILL DEVELOPMENT: The course develops skills in logical reasoning, mathematical modeling, problem-solving, algorithmic thinking, and graph analysis through the study of logic, relations, Boolean algebra, and graph theory. It also enhances computational and analytical skills by applying algebraic structures, formal languages, recursion, and coding techniques to real-world computing problems.