PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PROGRAMME REGULATIONS & CURRICULUM

2024-28

PRESIDENCY SCHOOL OF COMPUTER SCIENCE & ENGINEERING

BACHELOR OF TECHNOLOGY (B.TECH.)

COMPUTER SCIENCE AND TECHNOLOGY (DEVOPS)

PRESIDENCY UNIVERSITY

Presidency University Act, 2013 of the Karnataka Act No. 41 of 2013 | Established under Section 2(f) of UGC Act, 1956
Approved by AICTE, New Delhi

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Program Regulations and Curriculum

2024-2028

BACHELOR OF TECHNOLOGY (B.Tech.) in

COMPUTER SCIENCE AND TECHNOLOGY (DEVOPS)

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

(As amended up to the 24th Meeting of the Academic Council held on 3rd August 2024. This document supersedes all previous guidelines)

Regulations No: PU/AC-23.9/SOCSE03/CDV/2023-2027

Resolution No.10 of the 24th Meeting of the Academic Council held on 03rd August 2024, and ratified by the Board of Management in its 24th Meeting held on 05th August, 2024.

AUGUST-2024

Table of Contents

PART A – PROGRAM REGULATIONS

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

1.1 Vision of the University

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

1.2 Mission of the University

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

1.3 Vision of Presidency School of Computer Science and Engineering

To be a value-based, practice-driven Presidency School of Computer Science and Engineering, committed to developing globally competent engineers, dedicated to developing cutting-edge technology to enhance the 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 Credit System (CBCS) with focus on Social Project Based Learning, Industrial Training, and Internship to enable the students to become eligible and fully equipped for employment in industries, choose higher studies or entrepreneurship.

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

3. Short Title and Applicability

a. These Regulations shall be called the Bachelor of Technology Degree Program Regulations and Curriculum 2024-2028.
b. These Regulations are subject to, and pursuant to the Academic Regulations.
c. These Regulations shall be applicable to the ongoing Bachelor of Technology Degree Programs of the 2024-2028 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, 2024-2028;*

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

gg. *"PSCS" means the Presidency School of Computer Science and 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 2024-2028 are subject to, and, pursuant to the Academic Regulations. These Program Regulations shall be applicable to the following ongoing Bachelor of Technology (B.Tech.) Degree Programs of 2024-2028 offered by the Presidency School of Computer Science an d Engineering (PSCS):

1. B.Tech. Computer Science and Engineering

2. B. Tech. Computer Science and Technology (Big Data)

3. B. Tech. Computer Science and Engineering (Block Chain)

4. B. Tech. Computer Science and Technology (DevOps)

5. B. Tech. Computer Science and Engineering (Cyber Security)

6. B. Tech. Computer Science and Engineering (Internet of Things)

7. B. Tech. Computer Science and Engineering (Data Science)

8. B. Tech. Computer Science and Technology [Artificial Intelligence and Machine Learning]

9. B. Tech. Information Science and Technology [Artificial Intelligence and Data Science]

10. B. Tech. Computer Science and Information Technology

11. B. Tech. Computer Science and Engineering (Networks)

12. B. Tech. Computer Engineering

13. B. Tech. Information Science and Engineering [Artificial Intelligence and Robotics]

14. B. Tech. Computer Science and Engineering (Artificial Intelligence and Machine Learning)

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 16.1 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 19.0 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 able to:

PEO1. Demonstrate success as Computer Science and Engineering with innovative skills, moral and ethical values.

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

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

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

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

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

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

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

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

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

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

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

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

8.2 Program Specific Outcomes (PSOs):

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

PSO 01: Problem Analysis: Identify, formulate, research literature, and analyse complex engineering problems related to software development & project management methodologies, Computing, DevOps tools and practices for substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PSO 02: Design/development of Solutions: Design solutions for complex engineering problems related to software development & project management methodologies, Computing, DevOps tools and practices and design system components or processes

that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PSO 03: Modern Tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities related to software development & project management methodologies.

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/BE/BS 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. (Computer Science and Technology-Dev Ops) 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 (Computer Science and Technology-Dev Ops) 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:

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

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

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

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

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

11 Change of Branch / Discipline / Specialization

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

12.5 Assessment Components and Weightage

** - Shall be decided at School level

2

Table 1: 12.5 Assessment Components and Weightage											
S. No	Credit Structure [L-T-P-C]	Percentage/ Marks	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 * Except for full stack courses
		Marks	25	25	25	25	50	50	-	200	
3	1-0-4-3	Percentage	-	25%	10%	40%	5%	20%	-	100 %	Mid-Term & End Term by School
		Marks	-	25	10	40	5	20	-	100	
4	2-0-4-4	Percentage	12.50%	12.50 %	10%	15%	20%	30%	-	100 %	*Mid-Term & End Term by CoE
		Marks	25	25	20	30	40	60	-	200	
5	0-0-4-2	Percentage	-	50%	-	-	-	-	50%	100 %	Project evaluated by IC at School level
		Marks	-	50	-	-	-	-	50	100	
6	0-0-2-1	Percentage	-	100%	-	-	-	-	-	100 %	Only CA at School Level
		Marks	-	100	-	-	-	-	-	100	
7	3-0-2-4	Percentage	12.50%	12.50 %	15%	10%	30%	20%	-	100 %	Mid-Term & End Term by CoE
		Marks	25	25	30	20	60	40	-	200	
8	2-0-0-2	Percentage	25%	-	25%	-	50%	-	-	100 %	Mid-Term & End Term by CoE
		Marks	50	-	50	-	100	-	-	200	

*CSE3150-Front End Full stack development

CSE3151-Java Full Stack Development

CSE3152-.Net Full Stack development

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

Normally, for Practice/Skill based Courses, without a defined credit structure (L–T–P) [NTCC], but with assigned Credits (as defined in Clause 5.2**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.

2.1 Minimum Performance Criteria:

2.1.1 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.

2.1.2 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.

2.1.3 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 Clauses 8.9.1 and 8.9.2 of academic regulations) in the "Make-Up Examinations" of the concerned Course. Further, the student has an option to re-register for the Course and clear the same in the summer term/ subsequent semester if he/she wishes to do so, provided the Course is offered.

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

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

3.1 The transfer of credits shall be examined and recommended by the Equivalence Committee (Refer ANNEXURE B of academic regulations) and approved by the Dean - Academics.

3.2 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.

3.3 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:

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

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

3.3.3 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.

3.3.4 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.

3.3.5 A student shall request for transfer of credits only from such approved Courses as mentioned in Sub-Clause 3.3.2 above.

3.3.6 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.

3.3.7 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.

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

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

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

3.3.9 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 for the award of the concerned Degree.

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

3.4 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.

PART B – PROGRAM STRUCTURE

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

The B.Tech. (Computer Science and Technology-Dev Ops) Program Structure (2024-2028) totalling 160 credits. Table 3 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. (Computer Science and Technology-Dev Ops) 2024-2028: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets

Sl. No.	Baskets	Credit Contribution
1	Humanities and Social Sciences including Management Courses (HSMC)	10
2	Basic Science Courses (BSC)	19
3	Engineering Science Courses (ESC)	23
4	Professional Core Courses (PCC)	68
5	Professional Elective Courses (PEC)	18
6	Open Elective Courses (OEC)	06
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 60% out of the total credits of 160 for B.Tech. (Computer Science and Technology-Dev Ops) 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.

PART C – CURRICULUM STRUCTURE

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

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

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

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

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

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

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

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).

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

S.No	Course Code	Course Name	L	T	P	C
1	ENG1002	Technical English	1	0	2	2
2	PPS1001	Introduction to soft skills	0	0	2	1
3	DES1146	Introduction to Design Thinking	1	0	0	1
4	ENG2001/ FRLXXXX	Advanced English / Foreign Language courses	1	0	2	2
5	PPS1012	Enhancing Personality Through Soft Skills	0	0	2	1
6	MGTXXXX	Managerial Economics and Financial Analysis	3	0	0	3
					Total No. of Credits	**10**

Table 3.2 : List of Basic Science Courses (BSC)						
S.No	Course Code	Course Name	L	T	P	C
1	MAT1001	Calculus and Linear Algebra	3	0	2	4
2	PHY1002	Optoelectronics and Device Physics	2	0	2	3
1	MAT1003	Applied Statistics	1	0	2	2
1	MAT2501	Integral Transforms and Partial Differential Equations	3	0	0	3
1	MAT2602	Numerical Computations	3	0	0	3
2	MAT2605	Discrete Mathematics	4	0	0	4
					Total No. of Credits	**19**

	Table 3.3 : List of Engineering Science Courses (ESC)					
S.No	Course Code	Course Name	L	T	P	C
1	MEC1006	Engineering Graphics	2	0	0	2
2	CSE1004	Problem Solving Using C	1	0	4	3
3	ECE2007	Digital Design	2	0	2	3
4	CIV1008	Basic Engineering Sciences	2	0	0	2
5	CSE1006	Problem Solving using JAVA	1	0	4	3
6	EEE1007	Basics of Electrical and Electronics Engineering	3	0	2	4
7	ECE2010	Innovative Projects Using Arduino	-	-	-	1
8	CSE1500	Computational Thinking Using Python	2	0	2	3
9	CSE2510	Competitive Programming and Problem Solving	0	0	4	2
10	MEC1006	Engineering Graphics	2	0	0	2
					Total No. of Credits	**23**

Table 3.4 : List of Professional Core Courses (PCC)						
S.No	Course Code	Course Name	L	T	P	C
1	CDV1700	DevOps Foundations	3	0	0	3
2	CSE2502	Operating Systems	3	0	0	3

3	CSE2514	Operating Systems Lab	0	0	2	1
4	CSE1508	Data Structures	3	0	0	3
5	CSE2501	Computer Organization and Architecture	3	0	0	3
6	CSE2000	Software Design and Development	3	0	0	3
7	CSE1509	Data Structures Lab	0	0	4	2
8	CSE1512	Analysis of Algorithms	3	1	0	4
9	CSE1514	Object Oriented Programming Using Java	3	0	0	3
10	CSE1510	Database Management Systems	3	0	0	3
11	**CSE1504**	Web Technologies	2	0	0	2
12	**CSE1505**	Web Technologies Lab	0	0	2	1
13	CDV2500	Software Project Management	3	0	0	3
14	CSE1515	Object Oriented Programming Using Java Lab	0	0	4	2
15	CSE1511	Database Management Systems Lab	0	0	2	1
16	**CSE1513**	Analysis of Algorithms Lab	0	0	2	1
17	CDV2501	Agile Structures and Frameworks	3	0	0	3
18	CDV2502	DevOps Tools Internals	3	0	0	3
19	CDV2503	DevOps Tools Internals Laboratory	0	0	2	1
20	CDV2504	Software Testing	3	0	0	3
21	CDV2505	Software Testing Lab	0	0	2	1
22	CSE1700	Essentials of AI	3	0	0	3
23	CSE1701	Essentials of AI Lab	0	0	4	2
24	CSE2508	Mobile Application Development	2	0	0	2
25	CSE2509	Mobile Application Development Lab	0	0	4	2
26	CDV2506	Continuous Integration and Continuous Delivery (CI/CD) Pipelines	3	0	0	3
27	CDV2507	Continuous Integration and Continuous Delivery (CI/CD) Pipelines Lab	0	0	2	1
28	CDV2508	Edge Computing and DevOps	3	0	0	3
29	CDV2509	DevSecOps	3	0	0	3
					Total No. of Credits	**68**

Table 3.5 : List of course in Project Work basket (PRW)					
S.No	Course Name	L	T	P	C
1	Capstone Project	0	0	0	10
2	Internship	0	0	0	2
3	Mini Project	0	0	0	4
				Total No. of Credits	**16**

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

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

18.1. Internship

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

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

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

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

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

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

18.2 Mini Project

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

18.2.1 The Mini Project 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 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 project work confirms to the University that the project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.3 Capstone Project

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

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

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

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

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

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

18.4 Research Project / Dissertation

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

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

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

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

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 - DevOps with Cloud Computing

S.No	Course Code	Course Name	L	T	P	C
1	CDV3400	Fundamentals of Cloud Computing	3	0	0	3
2	CDV3401	Version Control with Git & GitHub/GitLab	3	0	0	3
3	CDV3402	Serverless Computing	3	0	0	3
4	CDV3403	Infrastructure as Code (IaC) with Terraform & Ansible	3	0	0	3
5	CDV3404	Cloud Automation & Orchestration	3	0	0	3

6	CDV3405	Cloud Infrastructure and Systems Software	3	0	0	3
Track 2 - DevOps with AI						
1	CDV3406	AI-Powered Chatbots & Automation in DevOps	3	0	0	3
2	CDV3407	AIOps – AI-Driven DevOps & IT Operations	3	0	0	3
3	CDV3408	Predictive Analytics for System Monitoring & Performance Optimization	3	0	0	3
4	CDV3409	AI-Enabled Cloud Cost Optimization	3	0	0	3
5	CDV3410	AI-Powered Incident Management & Root Cause Analysis	3	0	0	3
6	CDV3411	Reinforcement Learning for Self-Healing DevOps Pipelines	3	0	0	3
Track 3 - DevOps with Fin Tech						
1	CDV3412	DevOps Fundamentals for Financial Services	3	0	0	3
2	CDV3413	Introduction to FinTech & Digital Banking	3	0	0	3
3	CDV3414	Microservices Architecture for Banking Applications	3	0	0	3
4	CDV3415	Automated Testing & Monitoring in FinTech	3	0	0	3
5	CDV3416	DevSecOps in Financial Services	3	0	0	3
6	CDV3417	Blockchain & Smart Contracts for Financial Transactions	3	0	0	3

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

Table 3.7 : Open Elective Courses Baskets: Minimum Credits to be earned from this Basket is 12

Sl. No.	Course Code	Course Name					Type of Skill / Focus	Course Caters to	Prerequisites / Corequisites	Antirequisites	Future Courses that need this as a Prerequisite
Chemistry Basket											
1	CHE1003	Fundamentals of Sensors					S	ES	-	-	-
2	CHE1004	Smart materials for IOT					S	ES	-	-	-
3	CHE1005	Computational Chemistry					S	ES	-	-	-
4	CHE1006	Introduction to Nano technology					S	ES	-	-	-
5	CHE1007	Biodegradable electronics					S	ES	-	-	-
6	CHE1008	Energy and Sustainability					S	ES	-	-	-
7	CHE1009	3D printing with Polymers					S	ES	-	-	-
8	CHE1010	Bioinformatics and Healthcare IT					S	ES	-	-	-
9	CHE1011	Chemical and Petrochemical catalysts					S	ES	-	-	-
10	CHE1012	Introduction to Composite materials					S	ES	-	-	-
11	CHE1013	Chemistry for Engineers					S	ES	-	-	-
12	CHE1014	Surface and Coatings technology					S	ES	-	-	-
13	CHE1015	Waste to Fuels					S	ES	-	-	-
14	CHE1016	Forensic Science					S	ES	-	-	-
Civil Engineering Basket											
1	CIV1001	Disaster mitigation and management					S	-	-	-	-
2	CIV1002	Environment Science and Disaster Management					FC	-	-	-	-
3	CIV2001	Sustainability Concepts in Engineering					S	-	-	-	-
4	CIV2002	Occupational Health and Safety					S	-	-	-	-
5	CIV2003	Sustainable Materials and Green Buildings					EM	-	-	-	-
6	CIV2004	Integrated Project Management					EN	-	-	-	-
7	CIV2005	Environmental Impact Assessment					EN	-	-	-	-

8	CIV2006	Infrastructure Systems for Smart Cities					EN	-	-	-	-
9	CIV2044	Geospatial Applications for Engineers					EM	-	-	-	-
10	CIV2045	Environmental Meteorology					S	-	-	-	-
11	CIV3046	Project Problem Based Learning					S	-	-	-	-
12	CIV3059	Sustainability for Professional Practice					EN	-	-	-	-
Commerce Basket											
1	COM2001	Introduction to Human Resource Management					F	HP/ GS	-	-	-
2	COM2002	Finance for Non Finance					S	-	-	-	-
3	COM2003	Contemporary Management					F	-	-	-	-
4	COM2004	Introduction to Banking					F	-	-	-	-
5	COM2005	Introduction to Insurance					F	-	-	-	-
6	COM2006	Fundamentals of Management					F	-	-	-	-
7	COM2007	Basics of Accounting					F	-	-	-	-
Computer Science Basket											
1	CSE2002	Programming in Java					S/EM	-	-	-	-
2	CSE2003	Social Network Analytics					S	GS	-	-	-
3	CSE2004	Python Application Programming					S/ EM	-	-	-	-
4	CSE2005	Web design fundamentals					S/ EM/ EN	-	-	-	-
5	CSE3111	Artificial Intelligence : Search Methods For Problem Solving					S/ EM/ EN	-	-	-	-
6	CSE3112	Privacy And Security In Online Social Media					S/ EM/ EN	-	-	-	-
7	CSE3113	Computational Complexity					S/ EM/ EN	-	-	-	-
8	CSE3114	Deep Learning for Computer Vision					S/ EM/ EN	-	-	-	-
9	CSE3115	Learning Analytics Tools					S/ EM/ EN	-	-	-	-
Design Basket											
1	DES1001	Sketching and Painting					S	-	-	-	-
2	DES1002	Innovation and Creativity					F	-	-	-	-

3	DES1121	Introduction to UX design					S	-	-	-	-
4	DES1122	Introduction to Jewellery Making					S	-	-	-	-
5	DES1124	Spatial Stories					S	-	-	-	-
6	DES1125	Polymer Clay					S	-	-	-	-
7	DES2001	Design Thinking					S	-	-	-	-
8	DES1003	Servicability of Fashion Products					F	ES	-	-	-
9	DES1004	Choices in Virtual Fashion					F	ES, GS, HP	-	-	-
10	DES1005	Fashion Lifestyle and Product Diversity					F	ES, GS, HP	-	-	-
11	DES1006	Colour in Everyday Life					F	ES	-	-	-
12	DES2080	Art of Design Language					S	-	-	-	-
13	DES2081	Brand Building in Design					S	-	-	-	-
14	DES2085	Web Design Techniques					S	-	-	-	-
15	DES2089	3D Modeling for Professionals					S	-	-	-	-
16	DES2090	Creative Thinking for Professionals					S	-	-	-	-
17	DES2091	Idea Formulation					S	-	-	-	-
Electrical and Electronics Basket											
1	EEE1002	IoT based Smart Building Technology					S	-	-	-	-
2	EEE1003	Basic Circuit Analysis					S	-	-	-	-
3	EEE1004	Fundamentals of Industrial Automation					S	-	-	-	-
4	EEE1005	Electric Vehicles & Battery Technology					S	-	-	-	-
5	EEE1006	Smart Sensors for Engineering Applications					S	-	-	-	-
Electronics and Communication Basket											
1	ECE1003	Fundamentals of Electronics					F	-	-	-	-
2	ECE1004	Microprocessor based systems					F	-	-	-	-
3	ECE3089	Artificial Neural Networks					S	-	-	-	-
4	ECE3097	Smart Electronics in Agriculture					F/EM	-	-	-	-
5	ECE3098	Environment Monitoring Systems					F/EM	-	-	-	-
6	ECE3102	Consumer Electronics					F/EM	-	-	-	-

7	ECE3103	Product Design of Electronic Equipment					S/F/EM / EN	-	-	-	-
8	ECE3106	Introduction to Data Analytics					F/EM	-	-	-	-
9	ECE3107	Machine Vision for Robotics					F/EM	-	-	-	-
English Basket											
1	ENG1008	Indian Literature					-	GS/HP	-	-	-
2	ENG1009	Reading Advertisement					S	-	-	-	-
3	ENG1010	Verbal Aptitude for Placement					S	-	-	-	-
4	ENG1011	English for Career Development					S	-	-	-	-
5	ENG1012	Gender and Society in India					-	GS/HP	-	-	-
6	ENG1013	Indian English Drama					-	-	-	-	-
7	ENG1014	Logic and Art of Negotiation					-	-	-	-	-
8	ENG1015	Professional Communication Skills for Engineers					-	-	-	-	-
DSA Basket											
1	DSA2001	Spirituality for Health					F	HP	-	-	-
2	DSA2002	Yoga for Health					S	HP	-	-	-
3	DSA2003	Stress Management and Well Being					F	-	-	-	-
Kannada Basket											
1	KAN1001	Kali Kannada					S	-	-	-	-
2	KAN1003	Kannada Kaipidi					S	-	-	-	-
3	KAN2001	Thili Kannada					S	-	-	-	-
4	KAN2003	Pradharshana Kale					S	-	-	-	-
5	KAN2004	Sahithya Vimarshe					S	-	-	-	-
6	KAN2005	Anuvadha Kala Sahithya					S	-	-	-	-
7	KAN2006	Vichara Manthana					S	-	-	-	-
8	KAN2007	Katha Sahithya Sampada					S	-	-	-	-
9	KAN2008	Ranga Pradarshana Kala					S	-	-	-	-
Foreign Language Basket											
1	FRL1004	Introduction of French Language					S	S	-	-	-
2	FRL1005	Fundamentals of French					S	S	-	-	-

3	FRL1009	Mandarin Chinese for Beginners					S	S	-	-	-
Law Basket											
1	LAW1001	Introduction to Sociology					2	F	HP	-	-
2	LAW2001	Indian Heritage and Culture					2	F	HP/GS	-	-
3	LAW2002	Introdcution to Law of Succession					2	F	HP/GS	-	-
4	LAW2003	Introduction to Company Law					2	F	HP	-	-
5	LAW2004	Introduction to Contracts					F	HP	-	-	-
6	LAW2005	Introduction to Copy Rights Law					F	HP	-	-	-
7	LAW2006	Introduction to Criminal Law					F	HP	-	-	-
8	LAW2007	Introduction to Insurance Law					F	HP	-	-	-
9	LAW2008	Introduction to Labour Law					F	HP	-	-	-
10	LAW2009	Introduction to Law of Marriages					F	HP/GS	-	-	-
11	LAW2010	Introduction to Patent Law					F	HP	-	-	-
12	LAW2011	Introduction to Personal Income Tax					F	HP	-	-	-
13	LAW2012	Introduction to Real Estate Law					F	HP	-	-	-
14	LAW2013	Introduction to Trademark Law					F	HP	-	-	-
15	LAW2014	Introduction to Competition Law					F	HP	-	-	-
16	LAW2015	Cyber Law					F	HP	-	-	-
17	LAW2016	Law on Sexual Harrassment					F	HP/GS	-	-	-
18	LAW2017	Media Laws and Ethics					F	HP/GS	-	-	-
Mathematics Basket											
1	MAT2008	Mathematical Reasoning					S	-	-	-	-
2	MAT2014	Advanced Business Mathematics					S	-	-	-	-
3	MAT2041	Functions of Complex Variables					S	-	-	-	-
4	MAT2042	Probability and Random Processes					S	-	-	-	-
5	MAT2043	Elements of Number Theory					S	-	-	-	-
6	MAT2044	Mathematical Modelling and Applications					S	-	-	-	-
Mechanical Basket (not to be offered for Mechanical Department students)											

1	MEC1001	Fundamentals of Automobile Engineering					F	-	-	-	-
2	MEC1002	Introduction to Matlab and Simulink					S/EM	-	-	-	-
3	MEC1003	Engineering Drawing					S	-	-	-	-
4	MEC2001	Renewable Energy Systems					F	ES	-	-	-
5	MEC2002	Operations Research & Management					F	-	-	-	-
6	MEC2003	Supply Chain Management					S/EM/EN	-	-	-	-
7	MEC2004	Six Sigma for Professionals					S/EM	-	-	MEC2008	-
8	MEC2005	Fundamentals of Aerospace Engineering					F	-	-	-	-
9	MEC2006	Safety Engineering					S/EM	ES	-	-	-
10	MEC2007	Additive Manufacturing					F/EM	-	-	-	-
11	MEC3069	Engineering Optimisation					S/EM	-	-	-	-
12	MEC3070	Electronics Waste Management					F/S	ES	-	-	-
13	MEC3071	Hybrid Electric Vehicle Design					S/EM	ES	-	-	-
14	MEC3072	Thermal Management of Electronic Appliances					S/EM	-	-	-	-
15	MEC3200	Sustainable Technologies and Practices					S/EM	-	-	-	-
16	MEC3201	Industry 4.0					S/EM	-	-	-	-
Petroleum Basket											
1	PET1011	Energy Industry Dynamics					FC	ES	-	NIL	-
2	PET1012	Energy Sustainability Practices					FC	ES	-	NIL	-
Physics Basket											
1	PHY1003	Mechanics and Physics of Materials					FC / SD				
2	PHY1004	Astronomy					FC				
3	PHY1005	Game Physics					FC / SD				
4	PHY1006	Statistical Mechanics					FC				
5	PHY1007	Physics of Nanomaterials					FC				
6	PHY1008	Adventures in nanoworld					FC				

7	PHY2001	Medical Physics					FC	ES			
8	PHY2002	Sensor Physics					FC / SD				
9	PHY2003	Computational Physics					FC				
10	PHY2004	Laser Physics					FC	ES			
11	PHY2005	Science and Technology of Energy					FC	ES			
12	PHY2009	Essentials of Physics					FC				
Management Basket- I											
1	MGT2007	Digital Entrepreneurship					S/EM/EN	-	-	-	-
2	MGT2015	Engineering Economics					S	-	-	-	-
3	MGT2023	People Management					S/EM/EN	HP	-	-	-
Management Basket- II											
1	MGT1001	Introduction to Psychology					F	HP	-	-	-
2	MGT1002	Business Intelligence					EN	-	-	-	-
3	MGT1003	NGO Management					S	-	-	-	-
4	MGT1004	Essentials of Leadership					EM/EN	GS/HP	-	-	-
5	MGT1005	Cross Cultural Communication					S/EM/EN	HP	-	-	-
6	MGT2001	Business Analytics					S/EM/EN	-	-	-	-
7	MGT2002	Organizational Behaviour					F	HP	-	-	-
8	MGT2003	Competitive Intelligence					S	-	-	-	-
9	MGT2004	Development of Enterprises					S/EM/EN	-	-	-	-
10	MGT2005	Economics and Cost Estimation					S/EM	-	-	-	-
11	MGT2006	Decision Making Under Uncertainty					S	-	-	-	-
12	MGT2008	Econometrics for Managers					S	-	-	-	-
13	MGT2009	Management Consulting					S/EM/EN	-	-	-	-
14	MGT2010	Managing People and Performance					S/EM/EN	HP/GS	-	-	-

15	MGT2011	Personal Finance					F	-	-	-	-
16	MGT2012	E Business for Management					S/EM	-	-	-	-
17	MGT2013	Project Management					EN / EM	GS/ HP/ES	-	-	-
18	MGT2014	Project Finance					EN / EM	HP	-	-	-
19	MGT2016	Business of Entertainment					EM/ EN	-	-	-	-
20	MGT2017	Principles of Management					S/EM/ EN	-	-	-	-
21	MGT2018	Professional and Business Ethics					S/EM/ EN	HP	-	-	-
22	MGT2019	Sales Techniques					S/EM/ EN	HP	-	-	-
23	MGT2020	Marketing for Engineers					S/EM/ EN	HP	-	-	-
24	MGT2021	Finance for Engineers					S/EM/ EN	HP	-	-	-
25	MGT2022	Customer Relationship Management					S/EM/ EN	HP	-	-	-
Media Studies Basket											
1	BAJ3050	Corporate Filmmaking and Film Business					EM	HP	-	-	-
2	BAJ3051	Digital Photography					EM	HP	-	-	-
3	BAJ3055	Introduction to News Anchoring and News Management					EM	-	-	-	-

21. List of NPTEL Courses for B.Tech. Compputer Science and Engineering- Block Chain

Sl. No	Course Code	Course Name	Total Credits	L-T-P-C
1	CSE3111	Artificial Intelligence : Search Methods For Problem Solving	3	3-0-0-3
2	CSE3112	Privacy And Security In Online Social Media	3	3-0-0-3
3	CSE3113	Computational Complexity	3	3-0-0-3
4	CSE3114	Deep Learning for Computer Vision	3	3-0-0-3
5	CSE3115	Learning Analytics Tools	3	3-0-0-3
6	CSE502	Technical Skills in JAVA	3	0-0-6-3
7	CSE503	Technical Skills in Python	3	0-0-6-3
8	CSE504	Comprehensive Technical Skills	5	0-0-10-5
9	CSE505	The Joy Of Computing Using Python	3	3-0-0-3
10	CSE3119	Coding Skills in Python	3	3-0-0-3
11	CSE3121	Parallel Computer Architecture	3	3-0-0-3
12	CSE3124	Games and Information	3	3-0-0-3
13	CSE3140	Introduction To Industry 4.0 And Industrial Internet Of Things	3	3-0-0-3
14	CSE3142	Affective Computing	3	3-0-0-3
15	CSE3112	Privacy and Security in Online Social Media	3	3-0-0-3
16	CSE3196	Foundations of Cyber Physical Systems	3	3-0-0-3
17	CSE3197	Getting Started with Competitive Programming	3	3-0-0-3
18	CSE3198	GPU Architectures And Programming	3	3-0-0-3
19	CSE3199	Artificial Intelligence: Knowledge Representation And Reasoning	3	3-0-0-3
20	CSE3200	Programming in Modern C++	3	3-0-0-3
21	CSE3201	Circuit Complexity Theory	3	3-0-0-3
22	CSE3202	Basics of Computational Complexity	3	3-0-0-3
23	CSE3212	ion to Computer and Network Performance Analysis Using Queuing	1	1-0-0-1
24	CSE3213	C Programming And Assembly Language	1	1-0-0-1
25	CSE3214	Python For Data Science	1	1-0-0-1
26	CSE3215	Software Conceptual Design	1	1-0-0-1
27	CSE3117	Industrial Digital Transformation	3	3-0-0-3
28	CSE3118	Blockchain for Decision Makers	3	3-0-0-3
29	CSE3349	Technology for Lawyers	3	3-0-0-3
30	CSEXXXX	Deep Learning for Natural Language Processing	3	3-0-0-3
31	CSEXXXX	Machine Learning for Engineering and science applications	3	3-0-0-3
32	CSEXXXX	Algorithms in Computational Biology and Sequence Analysis	3	3-0-0-3
33	CSEXXXX	Introduction to Large Language Models (LLMs)	3	3-0-0-3
34	CSEXXXX	Quantum Algorithms and Cryptography	3	3-0-0-3

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

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Basket
Semester 1 - Physics Cycle						**19**	**26**	
1	MAT1001	Calculus and Linear Algebra	3	0	2	4	5	BSC
2	PHY1002	Optoelectronics and Device Physics	2	0	2	3	4	BSC
3	MEC1006	Engineering Graphics	2	0	0	2	2	ESC
4	ENG1002	Technical English	1	0	2	2	3	HSMC
5	PPS1001	Introduction to soft skills	0	0	2	1	2	HSMC
6	CSE1004	Problem Solving Using C	1	0	4	3	5	ESC
7	ECE2007	Digital Design	2	0	2	3	4	ESC
8	DES1146	Introduction to Design Thinking	1	0	0	1	1	HSMC
Semester 2 - BES Cycle						**15**	**23**	
1	MAT1003	Applied Statistics	2	0	0	2	2	BSC
2	CHE1018	Environmental Science	1	0	2	0	3	MAC
3	CIV1008	Basic Engineering Sciences	2	0	0	2	2	ESC
4	CSE1006	Problem Solving using JAVA	1	0	4	3	5	ESC

5	ENG2001/ FRLXXXX	Advanced English / Foreign Language courses	1	0	2	2	3	HSMC
6	PPS1012	Enhancing Personality Through Soft Skills	0	0	2	1	2	HSMC
7	EEE1007	Basics of Electrical and Electronics Engineering	3	0	2	4	5	ESC
8	LAW1007	Indian Constitution and Professional Ethics for Engineers	1	0	0	0	1	MAC
9	ECE2010	Innovative Projects Using Arduino	-	-	-	1	0	ESC
Semester 3						**27**	**29**	
1	MAT2501	Integral Transforms and Partial Differential Equations	3	0	0	3	3	BSC
2	CDV1700	DevOps Foundations	3	0	0	3	3	PCC
3	CSE2502	Operating Systems	3	0	0	3	3	PCC
4	CSE2514	Operating Systems Lab	0	0	2	1	2	PCC
5	CSE1508	Data Structures	3	0	0	3	3	PCC
6	CSE2501	Computer Organization and Architecture	3	0	0	3	3	PCC
7	CSE1500	Computational Thinking Using Python	2	0	2	3	2	ESC
8	CSE2000	Software Design and Development	3	0	0	3	3	PCC
9	MGTXXXX	Managerial Economics and Financial Accounting	3	0	0	3	3	HSMC
10	CSE1509	Data Structures Lab	0	0	4	2	4	PCC
Semester 4						**27**	**32**	
1	MAT2602	Numerical Computations	3	0	0	3	3	BSC
2	MAT2013	Discrete Mathematics	4	0	0	4	4	BSC
3	CSE1512	Analysis of Algorithms	3	1	0	4	4	PCC
4	CSE1514	Object Oriented Programming Using Java	3	0	0	3	3	PCC
5	CSE1510	Database Management Systems	3	0	0	3	3	PCC
6	**CSE1504**	Web Technologies	2	0	0	2	2	PCC
7	**CSE1505**	Web Technologies Lab	0	0	2	1	2	PCC
8	CDV2500	Software Project Management	3	0	0	3	3	PCC
9	CSE1515	Object Oriented Programming Using Java Lab	0	0	4	2	4	PCC
10	CSE1511	Database Management Systems Lab	0	0	2	1	2	PCC
11	**CSE1513**	Analysis of Algorithms Lab	0	0	2	1	2	PCC
Semester 5						**25**	**29**	
1	CDV2501	Agile Structures and Frameworks	3	0	0	3	3	PCC
2	CDV2502	DevOps Tools Internals	3	0	0	3	3	PCC
3	CDV2503	DevOps Tools Internals Laboratory	0	0	2	1	2	PCC

4	CDV2504	Software Testing	3	0	0	3	3	PCC
5	CDV2505	Software Testing Lab	0	0	2	1	2	PCC
6	CSE1700	Essentials of AI	3	0	0	3	3	PCC
7	CSE1701	Essentials of AI Lab	0	0	4	2	4	PCC
8	CSE2508	Mobile Application Development	2	0	0	2	2	PCC
9	CSE2509	Mobile Application Development Lab	0	0	4	2	4	PCC
10	CDVXXXX	Professional Elective – I	3	0	0	3	3	PEC
11	CSE7000	Internship	-	-	-	2	0	PRW
Semester 6						**21**	**26**	
1	CDV2506	Continuous Integration and Continuous Delivery (CI/CD) Pipelines	3	0	0	3	3	PCC
2	CDV2507	Continuous Integration and Continuous Delivery (CI/CD) Pipelines Lab	0	0	2	1	2	PCC
3	CDV2508	Edge Computing and DevOps	3	0	0	3	3	PCC
4	CDV2509	DevSecOps	3	0	0	3	3	PCC
5	CDVXXXX	Professional Elective – II	3	0	0	3	3	PEC
6	CDVXXXX	Professional Elective – III	3	0	0	3	3	PEC
7	PPSXXXX	Industry Preparedness Program	2	0	0	0	2	MAC
8	XXXXXXX	Open Elective – I	3	0	0	3	3	OEC
9	CSE2510	Competitive Programming and Problem Solving	0	0	4	2	4	ESC
Semester 7						**16**	**12**	
1	CBCXXXX	Professional Elective – IV	3	0	0	3	3	PEC
2	CBCXXXX	Professional Elective – V	3	0	0	3	3	PEC
3	CBCXXXX	Professional Elective – VI	3	0	0	3	3	PEC
4	XXXXXXX	Open Elective – II	3	0	0	3	3	OEC
5	CSE7100	Mini Project				4	0	PRW
Semester 8						**10**	**0**	
1	CSE7300	Capstone Project	-	-	-	10	0	PRW
						160		

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.

Sample Catalogue is given below for reference:

IV. Course Catalogues:

Each course shall have a course catalogue with the following details:

i) Pre –Requisites of the course
ii) Course Description
iii) Course Outcome
iv) Course Content
iv) Reference Resources.

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

Course Code: MAT1001	Course Title: Calculus and Linear Algebra Type of Course: Basic Sciences Theory	L-T-P-C	3	0	2	4
Version No.	3.0					
Course Pre-requisites	Basic Concepts of Limits, Differentiation, Integration					
Anti-requisites	NIL					
Course Description	The course focuses on the concepts of calculus and linear algebra with reference to specific engineering problems. The course is of both conceptual and analytical type in nature. The lab sessions associated with the course are concerned with acquiring an ability to use the MATLAB software.					
Course Objective	The objective of the course is Skill Development of student by using Problem Solving Techniques.					
Course Out Comes	On successful completion of the course the students shall be able to: 1) Comprehend the knowledge of applications of matrix principles. 2) Understand the concept of partial derivatives and their applications. 3) Apply the principles of integral calculus to evaluate integrals. 4) Adopt the various analytical methods to solve differential equations. 5) Demonstrate the use of MATLAB software to deal with a variety of mathematical problems.					

Course Content:				
Module 1	Linear Algebra			10 Sessions

Review: Types of matrices, elementary transformations, rank of a matrix, normal form, Solution of systems of linear equations: (Homogenous and non-homogenous system) AX = O and AX = B using rank method.

Linear Algebra:

Eigenvalues and Eigenvectors of a real matrix – Characteristic equation – Properties of Eigenvalues and Eigenvectors – Cayley-Hamilton theorem – Diagonalization of matrices – Reduction of a quadratic form to canonical form by orthogonal transformation – Nature of quadratic forms.

Engineering Applications of Linear Algebra.

Module 2	Partial Derivatives			10 Sessions

Review: Differential calculus with single variable.

Partial Derivatives:

Homogeneous functions and Euler‘s theorem, Total derivative, Change of variables, Jacobians, Partial differentiation of implicit functions, Taylor‘s series for functions of two variables, Maxima and minima of functions of two variables, Lagrange‘s method of undetermined multipliers.

Engineering Applications of partial derivatives.

Module 3	Advanced Integral calculus			12 Sessions

Review: Integral calculus for single integrals.

Advanced Integral calculus:

Beta and Gamma functions–interrelation-evaluation of integrals using gamma and beta functions; error function-properties. Multiple Integrals- Double integrals – Change of order of integration – Double integrals in polar coordinates – Area enclosed by plane curves, evaluation of triple integrals-change of variables between Cartesian and cylindrical and spherical polar co-ordinates.

Engineering applications of partial derivatives.

Module 4	Ordinary Differential Equations	Assignment	Programming	12 Sessions

Review: First order and first-degree Ordinary Differential Equations, Method of separation of variables, Homogeneous and Non- Homogeneous Equations reducible to Homogeneous form.

Linear Differential Equations, Bernoulli's Differential Equation, Exact and Non- Exact Differential Equations, Higher order Differential Equation with constant coefficients and with right hand side of the form eax, sinax, cosax, eaxf(x), xnf(x) etc., Linear equations with variable coefficients such as Cauchy Equation and Lagrange's Equation, D-operators and Inverse D- operators, Method of Variation of Parameters.

Engineering applications of differential equations.

List of Laboratory Tasks:

Introductory Task: Introduction to usage of the software and simple programming tasks. [3 Sessions]

Experiment N0 1: Solution of Simple differentiation with single variable and use of chain Rule.

Experiment No. 2: Solution based on application of Tailors' Series using software

Experiment No. 3: Application of Maxima and Minima condition using software.

Experiment No. 4 Computation of different functions for a specific problem

Experiment No. 5 Computation of Area under a curve.

Experiment No. 6 Solution of a set of simultaneous equations in matrix method

Experiment No. 7 Computation of Eigen Values and Eigen Vectors.

Experiment No. 8 Solution of Partial Differential equation

Experiment No. 9 solution using Cauchy Equation and Lagrange's Equation

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: MatLab, Zylink.
Assignment:
List at least 3 sets of Matrix Applications concerning the respective branch of Engineering and obtain the solution using MATLAB. 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 Sankara Rao, Introduction to Partial differential equations, Prentice Hall of India, edition, 2011 B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

References:

Victor Henner, Tatyana Belozerova, Mickhail Khenner, Ordinary and Partial Differential Equations, CRC Press, Edition, 2013.

Walter Ledermann, Multiple integrals, Springer, 1st edition

Lay, Linear Algebra ansd its applications, 3rd Ed., 2002, Pearson Education India.

Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition

MatLab usage manual

E-resources/ Web links:

1. https://nptel.ac.in/courses/109104124
2. https://nptel.ac.in/courses/111106051
3. https://nptel.ac.in/courses/111102137
4. https://www.cuemath.com/learn/mathematics/algebra-vs-calculus/
5. https://stanford.edu/~shervine/teaching/cs-229/refresher-algebra-calculus
6. https://math.hmc.edu/calculus/hmc-mathematics-calculus-online-tutorials/linear-algebra/
7. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
8. https://www.scu.edu.au/study-at-scu/units/math1005/2022/

Topics relevant to the development of Foundation Skills: All solution methods

Topics relevant to development of Employability skills: Use of Matlab software.

Course Code: PHY1002	Course Title: Optoelectronics and Device Physics Type of Course: Engineering Sciences Theory	L-T-P-C	2-0-2-3
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, experimental and analytical skills. The associated 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 Out Comes	On successful completion of the course the students shall be able to: CO1: Describe the concepts of semiconductors, magnetic materials and superconductors. CO2: Apply the concept of materials in the working of optoelectronic and magnetic devices. CO3: Discuss the quantum concepts used in advanced microscopy and quantum computers. CO4: Explain the applications of lasers and optical fibers in various technological fields. CO5: Interpret the results of various experiments to verify the concepts used in optoelectronics and advanced devices. [Lab oriented].		
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Optoelectronics and device physics “and attain Skill Development through Experiential Learning techniques		
Course Content:			

Module 1	Fundamentals of Materials.	Assignment	Plotting of magnetization (M) v/s Magnetic field (H) for diamagnetic, paramagnetic and ferromagnetic materials using excel/ origin software.	7 Sessions
Topics: Concept of energy bands, charge carriers, carrier concentration, concept of Fermi level, Hall effect, Magnetic materials, Superconductors:				
Module 2	Advanced Devices and applications	Assignment	Data collection on efficiency of solar cells.	8 Sessions
Topics: p-n junctions, Zener diode, transistor characteristics, Optoelectronic devices:, Solar cells, I-V characteristics, and LEDs				
Module 3	Quantum concepts and Applications	Term paper	Seminar on quantum computers.	8 Sessions
Topics: Planck's quantum theory, applications of Quantum theory: de-Broglie hypothesis, matter waves, properties. de-Broglie wavelength associated with an electron. Heisenberg's uncertainty principle. Schrodinger time independent wave equation. Particle in a box				
Module 4	Lasers and Optical fibers	Term paper	Case study on medical applications of Lasers.	7 Sessions
Topics: Interactions of radiations with matter, Characteristics of laser, conditions and requisites of laser, Modern day applications of laser: LIDAR, LASIK, Cutting, Welding and Drilling. 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.				
List of Laboratory Tasks: 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 current vs voltage characteristics of CdS photo-resistor at constant irradiance and To measure the photo-current as a function of the irradiance at constant voltage.

Level 1 To study the current vs voltage characteristics of CdS photo-resistor at constant irradiance.

Level 2: To measure the photo-current as a function of the irradiance at constant voltage.

Experiment No. 8: 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. 9: 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. 10: To determine the magnetic susceptibility of a given diamagnetic and paramagnetic substances using Quincke's method.

Level 1: To determine the magnetic susceptibility of a given diamagnetic substance.

Level 2: To determine the magnetic susceptibility of a given paramagnetic substance.

Experiment No. 11: 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. 12: 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.

Targeted Application & Tools that can be used:

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.

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

Engineering Physics by Avadhanalu, Revised edition, S. Chand Publications,2018.

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:

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=553045&site=ehost-live

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=833068&site=ehost-live

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=323988&site=ehost-live

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1530910&site=ehost-live

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=486032&site=ehost-live
Topics relevant to “SKILL DEVELOPMENT”: Fundamentals of materials, Lasers and optical fibers. for Skill Development through Participative Learning Techniques. This is attained through the Assignment/ Presentation as mentioned in the assessment component in course handout.

Course Code: MEC1006	Course Title: Engineering Graphics Type of Course: Engineering Science & Theory Only	L- T-P-C	2	0	0	2
Version No.	1.2					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	The course is designed with the objective of giving an overview of engineering graphics. It is introductory in nature and acquaints the students with the techniques used to create engineering drawings. The course emphasizes on projection of points, lines, planes and solids and isometric projections.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Engineering Graphics” and attain SKILL DEVELOPMENT through Problem solving methodologies.					
Course Outcomes	On successful completion of this course the students shall be able to: Demonstrate competency of Engineering Graphics as per BIS conventions and standards. Comprehend the theory of projection for drawing projections of Points, Lines and Planes under different conditions. Prepare multiview orthographic projections of Solids by visualizing them in different positions. Prepare pictorial drawings using the principles of isometric projections to visualize objects in three dimensions.					
Course Content:						
Module 1	Introduction	Assignment	Standard technical drawing	02 Sessions		

	to Drawing			
Topics: Introduction, drawing instruments and their uses, relevant BIS conventions and standards, Lettering, Line conventions, dimensioning, Selection of drawing sheet size and scale. [02 Hours: Comprehension Level]				
Module 2	Orthographic projections of Points, Straight Lines and Plane Surfaces	Assignment	Projection methods Analysis	10 Sessions
Topics: Introduction, Definitions – Elements of projection and methods of projection, Planes of projection, reference line and conventions adopted. First angle and third angle projections. Projection of Points in all 4 quadrants. Projections of Straight Lines (located in first quadrant/first angle projection only): True and apparent lengths, true and apparent Inclinations to reference planes. (No application problems). Projection of Plane surfaces (First angle projection): Regular plane surfaces – triangle, square, rectangle, pentagon, hexagon and circle – in different positions inclined to both the planes using change of position method only. [10 Hours: Application Level]				
Module 3	Orthographic Projections of Solids	Assignment	Multi-view drawing Analysis	10 Sessions
Topics: Introduction, Projection of right regular prisms, pyramids, cone, hexahedron and tetrahedron in different positions (Problems resting on HP only and First angle projection). [10 Hours: Application Level]				
Module 4	Isometric Projections of Solids (Using isometric scale only)	Assignment	Spatial Visualization	8 Sessions

Topics: Introduction, Isometric scale, Isometric projections of right regular prisms, cylinders, pyramids, cones and their frustums, spheres and hemispheres, hexahedron (cube), and combination of 2 solids, conversion of orthographic view to isometric projection of simple objects. [8 Hours: Application Level]
Text Book: 1.N. D. Bhatt, “Engineering Drawing: Plane and Solid Geometry,” Charotar Publishing House Pvt. Ltd. References: K.R. Gopalakrishna, “Engineering Graphics”, Subhash Publishers, Bangalore. D. M. Kulkarni, A. P. Rastogi, A. K. Sarkar, “Engineering Graphics with AutoCAD,” Prentice Hall. D. A. Jolhe, “Engineering Drawing with Introduction to AutoCAD,” Tata McGraw Hill. Web resources: https://nptel.ac.in/courses/112103019
Topics relevant to “SKILL DEVELOPMENT”: Projection in first and third angle for SKILL DEVELOPMENT through Problem Solving methodologies. This is attained through the assessment component mentioned in the course handout.

Course Code: ENG1002	Course Title: Technical English Type of Course: Humanities Science / Theory	L-T-P-C	1-0-2-2
Version No.	V. 3		
Course Pre-requisites	Intermediate Level English		
Course Anti-requisites	NIL		

Course Description	Technical English course is designed to equip students with the language skills necessary for effective communication in technical and scientific contexts. The course focuses on the specialized vocabulary, writing styles, and communication techniques used in various technical fields, including engineering and information technology.			
Course Objectives	The objective of this course is to develop the learners' EMPLOYABILITY SKILLS by using EXPERIENTIAL LEARNING and PARTICIPATIVE LEARNING TECHNIQUES.			
Course Outcomes	On successful completion of the course, the students shall be able to: Develop proficiency in using technical vocabulary and terminology. Apply language skills for better speaking skills in technical fields. Write technical descriptions Demonstrate writing skills in writing technical documents such as reports, manuals, and articles.			
Course Content:				
Module 1	Fundamentals of Technical Communication	Worksheets& Quiz	Vocabulary building	9 Classes
Introduction to Technical English Differences between Technical English and General English Technical Writing Basics Technical Vocabulary				
Module 2	Technical Presentation	Presentations	Speaking Skills	12 Classes
Introduction Planning the Presentation Creating the Presentation Giving the Presentation				
Module 3	Technical Description	Assignment	Group Presentation	12 Classes
Product Description				

Process Description

User Manuals

Transcoding: Diagrams, charts and images

Module 4	Technical Writing	Assignment	Writing Skills	12 Classes

Email Writing

Persuasive and Descriptive Language

Professional Email Etiquette

Writing clear and concise technical emails

Communicating technical information effectively

Technical Report Writing

Types of technical reports (Lab reports, research reports, etc.)

Components of technical reports

Writing an abstract and executive summary

Structure and content organization

Transcoding: diagrams, charts and images

List of Laboratory Tasks:

Module-1

Level 1: Worksheets

Level 2: Worksheets

Module 2

Level 1: Preparing Presentation

Level 2: Giving Presentation (Individual)

Module-3

Level 1: Product Description & User Manual

Level 2: Process Description & Transcoding

Module 4

Level 1: Email Writing

Level 2: Report Writing

Targeted Applications & Tools that can be used:

Flipgrid

Quizzes

Youtube Videos

Podcast

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

Bring out the essence of technical communication with reference to the conventions of technical communication, with examples

Prepare a technical presentation on the importance of Technical Communication and its relevance in a technical field, with real-life examples.

The following individual, as well as group Assignments, will be given to the students.

Presentation

Describing a product/process

Individual Reports

Text Books

Kumar, Sanjay; Pushpalatha. English Language and Communication Skills for Engineers. Oxford University Press. 2018.

Brieger, Nick and Alison Paul. Technical English Vocabulary and Grammar.

https://nmetau.edu.ua/file/technical_english_vocabulary_and_grammar.pdf

Reference Book:

Chauhan, Gajendra Singh, and Kashmiramka, Smita, Technical Communication. Cengage Publication. 2018.

Sunder Jain. Technical Report Writing. Centrum Press, 2013.

John Bowden. “Writing a Report: How to Prepare, Write & Present Really Effective Reports?”. 9th Edition 2011

Comfort, Jeremy et. al. 1984. Business Reports in English. Cambridge University Press.

Sharma, R.C. and K. Mohan. 2011. Business Correspondence and Report Writing, Fourth Edition. Tata McGraw Hill.

Web Resources:

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=JSTOR1_3307.

https://puniversity.informaticsglobal.com:2282/ehost/detail/detail?vid=5&sid=3a77d69b-abe5-4681-b39d-32dfdcb8f4a5%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=154223466&db=iih Last,Suzan, et. al. Technical Writing Essentials. University of Victoria, British Columbia, 2019 (E- Book) Wambui, Tabita Wangare, et al. Communication Skills- Volume 1, LAP LAMBRET, USA, 2012 (E Book)
Topics Relevant to the Development of Employability Skills: Speaking Skills, Writing Skills, Critical Thinking and Critical Analysis, and Group Communication.

Course Code: PPS 1001	Course Title: Introduction to Soft Skills Type of Course: Practical Only Course	L-T-P-C	0	2	1
Version No.	1.0				
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.				
Anti-requisites	NIL				
Course Description	This course is designed to enable students understand soft skills concepts and improve confidence, communication and professional skills to give the students a competitive advantage and increase chances of success in the professional world. 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 “Soft Skills” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.				
Course Out Comes	On successful completion of this course the students shall be able to: CO1: Recognize significance of soft skills				

	CO2: Illustrate effective communication while introducing oneself and others CO3: List techniques of forming healthy habits CO4: Apply SMART technique to achieve goals and increase productivity		
Course Content:			
Module 1	INTRODUCTION TO SOFT SKILLS	Classroom activity	04 Hours
Topics: Setting Expectations, Ice Breaker, Significance of soft skills, Formal grooming, punctuality			
Module 2	EFFECTIVE COMMUNICATION	Individual Assessment	10 Hours
Topics: Different styles of communication, Difference between hearing and listening, Effective communication for success, Email etiquette, Self-introduction framework, Video introduction, email- writing, Resume Building- Digital, Video, Traditional.			
Module 3	HABIT FORMATION	Worksheets & Assignment	4 Hours
Topics: Professional and personal ethics for success, Identity based habits, Domino effect, Habit Loop, Unlearning, standing up for what is right			
Module 4	Goal setting & Time Management	Goal sheet	8 Hours
A session where students will be introduced to Time management, setting SMART Goals, Introduction to OKR Techniques, Time Management Matrix, steps to managing time through outbound group activity, making a schedule, Daily Plan and calendars (To Do List), Monitoring/charting daily activity			
Targeted Application & Tools that can be used: LMS			
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course			
Individual Assessment LMS MCQ			
The topics related to Skill Development: Communication and professional grooming, Goal setting and presentation for skill development through participative learning techniques. This is attained through assessment component mentioned in course handout.			

Course Code: CSE1004	Course Title: Problem Solving Using C Type of Course: School Core Lab Integrated.			L-T-P-C	1	0	4	3
Version No.	1.0							
Course Pre-requisites	NIL							
Anti-requisites	NIL							
Course Description	The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs and applications in C. ACAlso by learning the basic programming constructs they can easily switch over to any other language in future.							
Course Object	The objective of the course is to familiarize the learners with the concepts of Problem Solving Using C and attain Employability through Problem Solving Methodologies.							
Course Outcomes	On successful completion of this course the students shall be able to: Write algorithms and to draw flowcharts for solving problems Demonstrate knowledge and develop simple applications in C programming constructs Develop and implement applications using arrays and strings Decompose a problem into functions and develop modular reusable code Solve applications in C using structures and Union Design applications using Sequential and Random Access File Processing.							
Course Content:								
Module 1	Introduction to C Language	Quiz	Problem Solving	9 Hrs.				
Topics: Introduction to Programming – Algorithms – Pseudo Code - Flow Chart – Compilation – Execution – Preprocessor Directives (#define, #include, #undef) - Overview of C – Constants, Variables and Data types – Operators and Expressions – Managing Input and Output Operations – Decision Making and Branching - Decision Making and Looping.								
Module 2	Introduction to Arrays and Strings	Quiz	Problem Solving	9 Hrs.				

Topics:

Arrays: Introduction – One Dimensional Array – Initialization of One Dimensional Arrays – Example Programs – Sorting (Bubble Sort, Selection Sort) – Searching (Linear Search) - Two Dimensional Arrays – Initialization of Two Dimensional Arrays. Example Programs – Matrix operations. Strings: Introduction – Declaring and Initializing String

Variables – Reading Strings from Terminal – Writing String to Screen – String Handling Functions.

Module 3	Functions and Pointers	Quiz	Problem Solving	9 Hrs.

Topics:

Functions: Introduction – Need for User-defined functions – Elements of User-Defined Functions: declaration, definition and function call–Categories of Functions – Recursion. Pointers: Introduction – Declaring Pointer Variables – Initialization of Variables – Pointer Operators – Pointer Arithmetic – Arrays and Pointers – Parameter

Passing: Pass by Value, Pass by Reference.

Module 4	Structures and Union	Quiz	Problem Solving	9 Hrs.

Topics:

Structures: Introduction – Defining a Structure – Declaring Structure Variable – Accessing Structure Members – Array of Structures – Arrays within Structures – Union: Introduction – Defining and Declaring Union – Difference Between

Union and Structure.

Module 5	File handling	Case Study	Problem Solving	9 Hrs.

Topics:

Files: Defining and Opening a File – Closing a File – Input / Output Operations on File – Random Access Files

List of Practical Tasks Lab Sheet 1 (Module I)

Programs using IO Statements, Conditional Statements and Looping Statements

Lab Sheet 2 (Module II)

Programs using Arrays and Strings

Lab Sheet 3 (Module III)

Programs using Functions and Pointers

Lab Sheet 4 (Module IV)

Programs using Structures and Unions

Lab Sheet 5 (Module V)

Programs using Files
Text Book(s): 1. E. Balaguruswamy, “Programming in ANSI C”, 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.
Reference Book(s): Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020. ReemaThareja, “Programming in C”, Oxford University Press, Second Edition, 2016. Kernighan, B.W and Ritchie,D.M, “The C Programming language”, Second Edition, Pearson Education, 2015 Schildt Herbert, “C: The Complete Reference”, Tata McGraw Hill Education, 4th Edition, 2014. 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: ECE2007	Course Title: Digital Design Type of Course: Theory &Integrated Laboratory	L-T-P-C	2	0	2	3
Version No.	2.0					
Course Pre-requisites	[1] Elements of Electronics/Electrical Engineering, 2] Basic concepts of number representation, Boolean Algebra					
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 focusing on both combinational and sequential logic circuits. The course emphasizes on minimization techniques for making canonical and low-cost digital circuit implementations. This course deals with analysis and design of digital electronic circuits. The course also creates a foundation for future courses which includes Computer Architecture, Microprocessors, Microcontrollers, and Embedded Systems etc. The course enhances the Design, Implementation and Programming abilities through laboratory tasks. The associated laboratory provides an opportunity to verify the theoretical knowledge.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of Digital Design and attain the SKILL DEVELOPMENT through EXPERIENTIAL LEARNING.			
Course Outcomes	On successful completion of this course the students shall be able to: Describe the concepts of number systems, Boolean algebra and logic gates. Apply minimization techniques to simplify Boolean expressions. Demonstrate the Combinational circuits for a given logic Demonstrate the Sequential and programmable logic circuits Implement various combinational and sequential logic circuits using gates.			
Course Content:				
Module 1	Fundamentals of Number systems- Boolean algebra and digital logic	Application Assignment	Data Analysis task	06 classes
Topics: Review of Number systems and logic gates, Number base conversions, Overview of Boolean functions and simplifications, two, three, four variable K-Maps- Don't care conditions- Both SOP and POS- Universal Gates (NAND & NOR) Implementations. Introduction to HDL.				
Module 2	Boolean function simplification	Application Assignment	Data Analysis task	08 Classes
Topics: Introduction to Combinational circuits, Analysis, Design procedure, Binary Adder and Subtractor, Magnitude comparator, Parity generator and checker, Multiplexers-				

Demultiplexers, Decoders, Encoders and Priority Encoders, HDL Models of combinational circuits.

Module 3	Combinational Logic circuits:	Application Assignment	Programming Task & Data Analysis task	08 Classes

Topics:

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

List of Laboratory Tasks:

Experiment N0 1: Verify the Logic Gates truth table

Level 1: By using Digital Logic Trainer kit

Level 2: By using Analog devices like RPS, Volt meter, Resistors and ICs

Experiment No. 2: Verify the Boolean Function and Rules

Level 1: By using Digital Logic Trainer kit

Level 2: By using Analog devices like RPS, Volt meter, Resistors and ICs

Experiment No. 3: Design and Implementations of HA/FA

Level 1: By using basic logic gates and Trainer Kit

Level 2: By using Universal logic gates and Trainer Kit

Experiment No. 4: Design and Implementations of HS/FS

Level 1: By using basic logic gates and Trainer Kit

Level 2: By using Universal logic gates and Trainer Kit

Experiment No. 5: Design and Implementations of combinational logic circuit for specifications

Level 1: Specifications given in the form of Truth table

Level 2: Specification should be extracted from the given scenario

Experiment No. 6: Study of Flip flops

Experiment No. 7: Design and Implementations of sequential logic circuit for specifications

Level 1: Specifications given in the form of Truth table

Level 2: Specification should be extracted from the given scenario

Experiment No.8: HDL coding for basic combinational logic circuits

Level 1: Gate level Modeling

Level 2: Behavioral Modeling

Experiment No.9: HDL coding for basic sequential logic circuit

Level 1: Gate level Modeling

Level 2: Behavioral Modeling

Targeted Application & Tools that can be used:

Digital electronics is the foundation of all modern electronic devices such as cellular phones, MP3 players, laptop computers, digital cameras, high definition televisions, Home Automation, Communication in systems in industries

Professionally Used Software: HDL/VHDL/Verilog HDL/ OOPS

Text Book(s):

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

Thomas L. Floyd “DIGITAL LOGIC DESIGN” , Pearson Education, fourth edition.

Reference(s):

Reference Book(s):

R1. Jain, R. P., “Modern Digital Electronics”, McGraw Hill Education (India), 4th Edition

R2. Roth, Charles H., Jr and Kinney Larry L., “Fundamentals of logic Design”, Cengage Learning, 7th

Edition

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

eBook1: Mano, M. Morris and Ciletti Michael D., “Digital Design”, Pearson Education.

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

eBook2:Floyd “DIGITAL LOGIC DESIGN” fourth edition- ePub, eBook- [PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org.

NPTEL Course- NPTEL :: Electrical Engineering - NOC:Digital Electronic Circuits

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

Lab Tutorial: Multisim Tutorial for Digital Circuits - Bing video

CircuitVerse - Digital Circuit Simulator online

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

Digital Design 5: LOGISIM Tutorial & Demo

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

E-content:

Z. Xin-Li and W. Hong-Ying, "The Application of Digital Electronics in Networking Communication," 2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2016, pp. 684-687, doi: 10.1109/ICMTMA.2016.168.

An encoding technique for design and optimization of combinational logic circuit DipayanBhadra;Tanvir Ahmed Tarique;Sultan Uddin Ahmed;Md. Shahjahan;KazuyukiMurase2010 13th International Conference on Computer and Information Technology (ICCIT)

A. Matrosova and V. Provkin, "Applying Incompletely Specified Boolean Functions for Patch Circuit Generation," 2021 IEEE East-West Design & Test Symposium (EWDTS), 2021, pp. 1-4, doi: 10.1109/EWDTS52692.2021.9581029.

A. Matrosova, V. Provkin and E. Nikolaeva, "Masking Internal Node Faults and Trojan Circuits in Logical Circuits," 2019 IEEE East-West Design & Test Symposium (EWDTS), 2019, pp. 1-4, doi: 10.1109/EWDTS.2019.8884434.

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

Course Code: DES1146	Course Title: Introduction to Design Thinking Type of Course: Theory	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 the students shall be able to: Understand the concept and importance of Design Thinking. Differentiate between traditional problem-solving and Design Thinking. Identify the core stages of the Design Thinking process.					
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	Visual journal, book of essays,		Visual output generation, by Visual Journal and narrative development.		3 hours

		context-specific assignment/project			
Topic Definition and Introduction to Design Thinking Understand the Design Thinking Process					
Module 2	Design Thinking in Action	Visual journal, book of essays, context-specific assignment/project		Visual output generation, by visual journal and narrative development.	12 hours
Topics: Introduction to the steps of Design Thinking Process Understand use cases of Design thinking 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: Design ideation tools like Miro , SCAMPER etc. Research Tools for Human Centric Design using forecasting tools like WGSN Feedback tools like Google Forms , etc. Expert Lectures					
Text Book					

Thinking Design by S Balaram. New Delhi [India]: Sage Publications Pvt. Ltd. 2010. eBook., Database: eBook Collection (EBSCOhost)

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

Design Thinking by Clarke, Rachel Ivy. Series: Library Futures, Vol. 4. Chicago: ALA Neal-Schuman. 2020. eBook., Database: eBook Collection (EBSCOhost)

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

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

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

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

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%3F

Query%3Ddesign%2Bthinking%26so%3Drel&ab_segments=0%2FSYC-6168%2Ftest&refreqid=fastly-default%3A0d5b607b163f60876ca973ed90e22b1c&seq=1#metadata_info_tab_contents

Course Code: MAT1003	Course Title: Applied Statistics Type of Course: School Core	L-T-P-C	1	0	2	2
Version No.	3.0					
Course Pre-requisites	None					
Anti-requisites	None					
Course Description	The goal of this course is to provide a firm understanding of probability and statistics by means of a thorough treatment of descriptive statistics, probability and probability distributions keeping in mind the future courses having statistical, quantitative and probabilistic components. The course covers topics such as descriptive statistics, probability, rules for probability, random variables and probability distributions, standard discrete and continuous probability distributions.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Applied Statistics” and attain Skill Development Through Problem Solving techniques.					
Expected Outcome:	At the end of this course, students will be in a position to apply the techniques of descriptive statistics effectively interpret the ideas of probability and conditional probability demonstrate the knowledge of probability distributions Compute statistical parameters, correlation and regression, probability and sampling distributions using R software.					

Module 1	Descriptive Statistics	Assignment	Coding needed	10 classes
Introduction to Statistics, Data and statistical thinking, review of basic statistical parameters, Covariance, Correlation, Types of Measures of Correlation - Karl Pearson's Correlation Coefficient, Spearman Rank Correlation, linear regression, Multi linear regression .				
Module 2	Probability			6 classes
Introduction to Probability, Probability of an event, Addition Principle, Multiplication law, Conditional Probability, Total Probability and Baye's theorem with examples				
Module 3	Random Variables and Probability Distributions		Coding needed	14 classes
Introduction to Random variables, Discrete Random Variables and Continuous Random Variables, Probability Distributions, Probability Mass Function and Probability Density Function, Various Probability distributions, Binomial, Negative Binominal (Self Study), Poisson, Normal and Exponential distributions				
Module 4	Sampling Theory		Coding needed	15 classes
Introduction to Sampling Theory, Population, Statistic, Parameter, Sampling Distribution, Standard Error. Testing of Hypothesis, Types of Errors, Critical Region, level of Significance. Difference between Parametric and Non-parametric Tests, Large Sample Tests: Z-Test for Single Mean and Difference of Means (Self Study), Small Sample Tests: Student's t-Test for Single Mean and Difference of Means, F-Test, Chi-Square Test.				
Targeted Application & Tools that can be used: The objective of the course is to familiarize students with the theoretical concepts of probability and statistics and to equip them with basic statistical tools to tackle engineering and real-life problems. Tools used: R Software / MS-Excel				
Text Book				

Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying E Ye, Probability and Statistics for Engineers and Scientists, Pearson Education, 2016.
References James T. McClave, P. George Benson and Terry Sincich, Statistics for Business and Economics, 2018. David R. Anderson, Dennis J. Sweeney, Thomas A. Williams, Essentials of Modern Business Statistics with Microsoft Excel, 2020. David R. Anderson, Dennis J. Sweeney, Thomas A. Williams, Essentials of Statistics for Business and Economics, 2019. Douglas C. Montgomery and George C. Runger, Applied Statistics and Probability for Engineers, John Wiley and Sons, 2018. Richard A. Johnson, Miller and Freund's Probability and Statistics for Engineers, 2018. Kishor S Trivedi, Probability and Statistics with reliability, Queuing and Computer Science Applications, John Wiley & Sons, 2008.
Topics relevant to SKILL DEVELOPMENT: The goal of this course is to provide a firm understanding of probability and statistics by means of a thorough treatment of descriptive statistics, probability and probability distributions keeping in mind the future courses having statistical, quantitative and probabilistic components. The course covers topics such as descriptive statistics, probability, rules for probability, random variables and probability distributions, standard discrete and continuous probability distributions for Skill Development through Problem Solving methodologies. This is attained through assessment component mentioned in course handout.

Course Code: CHE1018	Course Title: Environmental Science	L- T- P- C	1	0	2	0
	Type of Course: School Core- Theory and Lab	Contact hours	1	0	2	3
Version No.	2.0					

Course Pre-requisites	NIL			
Anti-requisites	NIL			
Course Description	This course emphasizes the need to conserve biodiversity and adopt a more sustainable lifestyle by utilizing resources in a responsible way. Topics covered include basic principles of ecosystem functions; biodiversity and its conservation; human population growth; water resources, pollution; climate change; energy resources, and sustainability; Sustaining human societies, policies, and education. 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 “Environmental Science” and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques.			
Course Outcomes	On successful completion of this course the students shall be able to: Appreciate the historical context of human interactions with the environment and the need for eco-balance. Describe basic knowledge about global climate change with particular reference to the Indian context. Understand biodiversity and its conservation Develop an understanding on types of pollution and ways to protect the environment Learn about various strategies on Global environmental management systems			
Course Content:				
Module 1	Humans and the Environment	Assignment	Data Collection	01 class
Topics: The man-environment interaction: Mastery of fire; Origin of agriculture; Emergence of city-states; Great ancient civilizations and the environment. Self-learning topics: Humans as hunter-gatherers; Industrial revolution and its impact on the environment; Environmental Ethics and emergence of environmentalism.				
Module 2	Natural Resources and Sustainable Development	Assignment		03 Classes

Topics:

Overview of natural resources: Definition of resource; Classification of natural resources- biotic and abiotic, renewable and non-renewable. Water resources: Types of water resources- fresh water and marine resources;

Soil and mineral resources: Important minerals; Mineral exploitation Soil as a resource and its degradation.

Energy resources: Sources of energy and their classification, renewable and non-renewable sources of energy; Advantages and disadvantages.

Self- learning topics: Availability and use of water resources; Environmental impact of over-exploitation, issues and challenges.; Environmental problems due to extraction of minerals and use; Sustainable Development Goals (SDGs)- targets, indicators, and challenges for SDGs.

Module 3	Environmental Issues: Local, Regional and Global	Case study		02 Classes

Topics:

Environmental Pollution: Types of Pollution- air, noise, water, soil, municipal solid waste, hazardous waste; Trans- boundary air pollution; Acid rain; Smog.

Land use and Land cover change: land degradation, deforestation, desertification, urbanization. Global change: Ozone layer depletion; Climate change

Self -learning topics: Environmental issues and scales

Module 4	Conservation of Biodiversity and Ecosystems	Assignment		02 Classes

Topics:

Biodiversity-Introduction, types, Species interactions, Extinct, endemic, endangered and rare species, Threats to biodiversity: Natural and anthropogenic activities.

Self-learning topics: Mega-biodiversity, Hot-spots, Major conservation policies. Biodiversity loss: past and current trends, impact.

Module 5	Environmental Pollution and Health	Case study		03 Classes

Topics:

Pollution, Definition, point and nonpoint sources of pollution, Air pollution- sources, major air pollutants, health impacts of air pollution.

Water pollution– Pollution sources, adverse health impacts on human and aquatic life and mitigation, Water quality parameters and standards.

Soil pollution and solid waste- Soil pollutants and their sources, solid and hazardous waste, Impact on human health.

Self-learning topics: Noise pollution, Thermal and radioactive pollution.

	Module 6	Climate Change: Impacts, Adaptation and Mitigation	Assignment/case		02 Classes

Topics:

Understanding climate change: Natural variations in climate; Projections of global climate change with special reference to temperature, rainfall and extreme events; Importance of 1.5 °C and 2.0 °C limits to global warming; Impacts

Vulnerability and adaptation to climate change: Observed impacts of climate change on ocean and land systems; Sea level rise, changes in marine and coastal ecosystems; Impacts on forests and natural ecosystems; Indigenous knowledge for adaptation to climate change.

Self-learning topics: Mitigation of climate change: Synergies between adaptation and mitigation measures; National and international policy instruments for mitigation.

Module 7	Environmental Management	Case study	Data analysis	02 Classes

Topics:

Environmental management system: ISO 14001; Environmental risk assessment Pollution control and management; Waste Management- Concept of 3R (Reduce, Recycle and Reuse) and sustainability.

Self-learning topics: Environmental audit and impact assessment; Eco labeling /Eco mark scheme

Module 8	Environmental Treaties and Legislation	Case study	Data analysis	01 Classes

Topics:

Major International Environmental Agreements: Convention on Biological Diversity (CBD), Major Indian Environmental Legislations: Environmental Protection Act, Forest Conservation Act, Public awareness.

Self-learning topics: Paris Agreement, Conference of the Parties (COP), India's status as a party to major conventions: Air (Prevention and Control of Pollution) Act, Water (Prevention and control of Pollution) Act, Wildlife Protection Act.

List of laboratory tasks : Any eight experiments will be conducted

Determination of total alkalinity of a water sample (knowledge)

Estimation of water hardness by EDTA method and its removal (by zeolite/ ion exchange method) (Comprehensive)

Estimation of copper from industrial effluents by colorimetric method (Comprehensive)

Estimation of iron from industrial effluents by titrimetric method/potentiometric method (Comprehensive)

Estimation of nickel from industrial effluents by titrimetric method (Comprehensive)

Estimation of chloride in drinking water by titrimetric method (Comprehensive)

Estimation of fluoride in ground water by colorimetric method (Comprehensive)

Determination of calcium in aqueous solution (Comprehensive)

Determination of Total Dissolved Salts, conductivity and pH of a water samples (Knowledge)

Determination of Chemical oxygen demand in the industrial effluent. (Comprehensive)

Biological oxygen demand of waste water sample (Comprehensive)

Determination of dissolved oxygen of an industrial effluent (Comprehensive)

Quality monitoring analysis of a soil sample (knowledge)

Flame photometric estimation of Sodium and potassium (Application)

Gas Chromatographic analysis of volatile organic compounds (Application)

Targeted Application & Tools that can be used:

Application areas are Energy, Environment and sustainability

Tools: Statistical analysis of environmental pollutants using excel, origin 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.)

Lab evaluation/Assignment

End Term Exam

Self-learning

Assignment 1: Write a Statement of Environment report of your town/city/state/country

Assignment 2: Individual students will carry out the analyses of polluted solid, liquid, and gaseous samples and propose suitable mitigation measures. A detailed and in-depth report needs to be submitted for each case. This may include preparation of reagents, sample preparation (extraction), chemical analysis carried out, instruments and tools used, data collected and processed, inferences made and conclusions arrived at. Necessary support is given in the form of

lab manual and reference links to e-books.

Text Book

G. Tyler Miller and Scott Spoolman (2020), Living in the Environment, 20th Edition, Cengage Learning, USA

Krishnamurthy, K.V. (2003) Text book of Biodiversity, Science Publishers, Plymouth, UK.

Jackson, A.R. & Jackson, J.M. (2000), Environmental Science: The natural environment and human impact, Pearson Education.

Reference Books

Fisher, Michael H. (2018) An Environmental History of India- From Earliest Times to the Twenty-First Century, Cambridge University Press.

William P. Cunningham and Mary Ann Cunningham (2017), Principles of Environmental Science: Inquiry & Applications, 8th Edition, McGraw-Hill Education, USA.

Sinha N., (2020) Wild and Wilful. Harper Collins, India.

www.ipcc.org; https://www.ipcc.ch/report/sixth-assessment-report-cycle/

Theodore, M. K. and Theodore, Louis (2021) Introduction to Environmental Management, 2nd Edition. CRC Press.

Richard A. Marcantonio, Marc Lame (2022). Environmental Management: Concepts and Practical Skills. Cambridge University Press.

E-resources:

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AB_1_06082022_18126

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AB_1_06082022_8761

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AJ_1_02082022_3333

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AB_1_06082022_3063

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AB_1_06082022_20719

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AB_1_06082022_16824

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AB_1_06082022_3954

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO_AB_1_06082022_491

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=CUSTOM_PACKAGE_16012023_WORLD_BUSINESS_COUNCIL_SUSTAINABLE_488

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=CUSTOM_PACKAGE_16012023_WORLD_BUSINESS_COUNCIL_SUSTAINABLE_583

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=SP_RINGER_INDEST_1_171

https://presiuniv.knimbus.com/user#/searchresult?searchId=3R%20principle&_t=1687427221129 https://presiuniv.knimbus.com/user#/searchresult?searchId=eco%20labelling&_t=1687427279979 https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=TE_XTBOOK_LIBRARY01_06082022_395&xIndex=4 https://www.ugc.gov.in/oldpdf/modelcurriculum/env.pdf

Course Code: CIV1008	Course Title: Basic Engineering Sciences Type of Course: Theory Only	L-T-P-C	2	0	0	2
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This basic course on engineering science is designed to introduce students to the fields of civil, mechanical and petroleum engineering. Student will be exposed to various fields in civil engineering and different manufacturing techniques in addition to machinery for power production and consumption. Additionally, students will be getting an overview of various sectors of oil & gas industries. This course acquaints students to basics of Industry 4.0 and Construction 4.0. The course aims to enable students to appreciate the multidisciplinary nature of engineering design and operations in the current era with mechanization and digitization transforming every aspect of engineering.					
Course Objective	The objective of the course is skill development of student by using Participative Learning techniques.					
Course Outcomes	On successful completion of this course the students shall be able to: 1] Recognize the significance of various disciplines in Civil Engineering 2] Discuss the recent evolutions in Civil Engineering					

	3] Explain various energies, energy generating machineries and energy consumption machineries 4] Describe the fundamental concept and terminology associated with the Petroleum Industry 5] Distinguish between conventional and modern manufacturing techniques.			
Course Content:				
Module 1	Introduction to various fields in Civil Engineering	Assign ment	Case studies on different Civil Engineering Projects	6 Sessions
Topics: Introduction to Civil Engineering: Definition, scope and branches of Civil Engineering, Role of Civil Engineer, Overview of Infrastructure.				
Module 2	Current Trends and Evolution in Civil Engineering	Assign ment	Article Review	6 Sessions
Topics: Mechanization in Construction, Application of Digital Technologies in Planning, Design, execution, monitoring and maintenance of Construction. Overview of Smart Cities.				
Module 3	Power Production and Consumption Machinery	Assign ment & Quiz	Data Collection	6 Sessions
Topics: Energy and its types, Engines and their applications, Pumps-Compressors and their applications.				
Module 4	Overview of Petroleum Engineering	Assign ment & Quiz	Article Review	6 Sessions
Overview of the Petroleum Industry, Importance of Petroleum Engineering, lifecycle of Petroleum products, Classifications of E&P activities: Key difference between Offshore and Onshore, Onshore facilities, offshore platforms, Digitization of petroleum engineering				
Module 5	Industry 4.0	Assign ment & Quiz	Data Collection	6 Sessions
Topics: Conventional manufacturing process: Metal forming, metal removal and metal joining process.				

Modern Manufacturing process: 3D Printing / Additive Manufacturing.
Targeted Application & Tools that can be used: Application Areas include design and implementation of Smart City projects, Infrastructure maintenance, Power production, IC engines, Electric vehicles, onshore and offshore exploration and production activities
Project work/Assignment:
Assignment 1: Collect data and prepare report on various Mega Projects in Civil Engineering Assignment 2: Review Articles on current evolutions in Civil Engineering. Assignment 3: Collect data related to renewable energy generation (Wind, Solar) Assignment 4: Prepare an energy consumption chart for a compressor or pumps. Assignment 5: Prepare a report on role of 3D printing across various industries. Assignment 6: Prepare an assignment on geopolitical influence on oil and gas industries.
Text Book: T1. Elements of Civil and Mechanical Engineering, L.S. Jayagopal & R Rudramoorthy, Vikas Publishers T2. Elements of Mechanical Engineering, by VK Manglik T3. Fundamentals of Oil & Gas Industry for Beginners by Samir Dalvi, Notion Press; 1st edition
References K.P. Roy, S.K. Hajra Choudhury, Nirjhar Roy, “Elements of Mechanical Engineering”, Media Promoters and Publishers Pvt Ltd, Mumbai. Nontechnical Guide to Petroleum Geology, Exploration, Drilling & Production by Norman J. Hyne, PennWell Books; 3rd Revised edition Web-resources: Basic Civil Engineering https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2706932&site=ehost-live Post-parametric Automation in Design and Construction https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1155197&site=ehost-live Smart Cities : Introducing Digital Innovation to Cities

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1993146&site=ehost-live

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

Mechanical Engineering

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO106_REDO_1705

Additive Manufacturing: Opportunities, Challenges, Implications

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1134464&site=ehost-live

Society of Petroleum Engineers (SPE)

https://www.spe.org/en/

PetroWiki: A comprehensive online resource created by the Society of Petroleum Engineers that provides information on various aspects of petroleum engineering.

https://petrowiki.spe.org/PetroWiki

Rigzone: A resource for news and information about the oil and gas industry, including job postings and industry trends.

https://www.rigzone.com/

Topics relevant to the development of SKILLS:

Engines-Turbines and their applications.

Mechanization in Construction.

Digitization in Petroleum Industries

Course Code: CSE1006	Course Title: Problem Solving using JAVA Type of Course: Lab Integrated	L- T-P-C	1	0	4	3
Version No.	2.0					
Course Pre-requisites	CSE1004 – Problem-Solving Using C					
Anti-requisites	Nil					

<table>
<tr><td>Course Description</td><td colspan="4">This course introduces the core concepts of object-oriented programming. This course has theory and lab component which emphasizes understanding the implementation and application of object-oriented programming paradigm. It helps the student to build real-time secure applications by applying these concepts and also for effective problem-solving. The students interpret and understand the need for object-oriented programming to build applications.</td></tr>
<tr><td>Course Objective</td><td colspan="4">The objective of the course is to familiarize the learners with the concepts of Problem-Solving using JAVA and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques</td></tr>
<tr><td>Course Out Comes</td><td colspan="4">On successful completion of the course, the students shall be able to:
C.O. 1: Describe the basic programming concepts. [Knowledge]
C.O. 2: Apply the concept of classes, objects and methods to solve problems. [Application]
C.O. 3: Apply the concept of arrays and strings. [Application]
C.O. 4: Implement inheritance and polymorphism in building secure applications. [Application]
C.O. 5: Apply the concepts of interface and error handling mechanism. [Application]</td></tr>
<tr><td>Course Content:</td><td colspan="4"></td></tr>
<tr><td>Module 1</td><td>Basic Concepts of Programming and Java</td><td>Assignment</td><td>Data Collection/Interpretation</td><td>12 Sessions</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 2</td><td>Classes, objects, methods and Constructors</td><td>Case studies / Case let</td><td>Case studies / Case let</td><td>12 Sessions</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 3</td><td>Arrays, String and String buffer</td><td>Quiz</td><td>Case studies / Case let</td><td>14 Sessions</td></tr>
</table>

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	Quiz	Case studies / Case let	14 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	Quiz	Case studies / Case let	14 Sessions
Input/output Operation in Java(java.io Package), Streams and the new I/O Capabilities, Understanding Streams, working with File Objects, 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.				
List of Laboratory Tasks: P1 - Problem Solving using Basic Concepts. P2 - Problem Solving using Basic Concepts and Command Line Arguments. P3 - Programming assignment with class, objects, methods and Constructors. P4 - Programming assignment with method overloading. P5 - Programming assignment with constructor overloading. P6 - Programming assignment with Static members and static methods. P7 - Programming assignment with Nested classes. P8 - Programming assignment using Arrays. P9 - Programming assignment using Strings. P10 - Programming assignment using String Builder. P11 - Programming assignment using Inheritance and super keyword. P12 - Programming assignment using Method overriding and Dynamic method invocation. P13 - Programming assignment using Final keywords. P14 - Programming assignment using Abstract keywords. P15 - Programming assignment using Interface. P16 - Programming assignment using Interface.				

P17 - Programming assignment CharacterStream Classes

P18 - Programming assignment Read/Write Operations with File Channel

Targeted Application & Tools that can be used : JDK /eclipse IDE/ net Beans IDE.

Text Book

T1 Herbert Schildt, “The Complete Reference Java 2”, Tata McGraw Hill Education.

References

R1: Cay S Horstmann and Cary Gornell, “CORE JAVA volume I-Fundamentals”, Pearson

R2: James W. Cooper, “Java TM Design Patterns – A Tutorial”, Addison-Wesley Publishers.

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 the development of “Skill Development”:

Static Polymorphism

Method overloading, constructors

constructor overloading

this keyword

static keyword and Inner classes

Inheritance and Polymorphism.

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

ENG2001	Advanced English	L-T-P-C	1	0	2	2

Version No.	1.3			
Course Pre-requisites	ENG1002 Technical English			
Anti-requisites	NIL			
Course Description	The course emphasizes on technical communication at advanced level by exploring critical reading, technical presentation and review writing. The purpose of the course is to enable learners to review literature in any form or any technical article and deliver technical presentations. Extensive activities in practical sessions equip to express themselves in various forms of technical communications. Technical presentations and the module on career setting focus on learners' area of interests and enhance their English language writing skills to communicate effectively.			
Course Out Come	On successful completion of the course the students shall be able to: Develop a critical and informed response reflectively, analytically, discursively, and creatively to their reading. Communicate effectively, creatively, accurately and appropriately in their writing. Deliver technical presentations Design resume and create professional portfolio to find a suitable career			
Course Content: Theory				
Module 1	Critical Reasoning and Writing	Writing Essays	Critical Reading	4 Classes
Topics: A Catalog of Reading Strategies The Myth of Multitasking A Guide to Writing Essays Speculating about Causes or Effects Is Google Making Us Stupid (Self Study)				
Module 2	Technical Presentation	Presentation	Oral Skills	3 Classes
Topics: Planning the presentation Creating the presentation				

Giving the presentation				
Module 3	Writing Reviews	Prezi	Review Writing	4 Classes
Topics: Review Writing Short film reviews Advanced English Grammar (Self Study)				
Module 4	Starting your Career	Online Writing Lab	Writing Skills	4 Classes
Topics: Preparing a Resume Writing Effective Application Letter Creating a Professional Portfolio				
Course Content: Practical Sessions				
Module 1	Critical Reasoning and Writing		8 Classes	
Reading and Analyzing Level 1 – Annotation Level 2 - Assumptions Writing Narrative Essays Level 1 – Draft 1 Level 2 – Draft 2				
Module 2	Technical Presentation		10 Classes	
Fishbowl In Fishbowl, students form concentric circles with a small group inside and a larger group outside. Students in the inner circle engage in an in-depth discussion, while students in the outer circle listen and critique content, logic, and group interaction. Level 1 – within group Level 2 – Among 2 group Technical Group Presentation				
Module 3	Writing Reviews		Classes	
Practice Worksheets Level 1 – Eliminating the Passive Voice				

Level 2 – Simple, compound and complex sentences Writing Short Film Reviews		
Module 4	Starting your Career	Classes
Collaborative Project Job search and writing report Writing Resume		
Module 1-4	Academic Journal	2 Classes
Academic Journal Writing Level 1- Mid Term Level 2 – End Term		
Targeted Application & Tools that can be used: Writing reports, Review writing, Group Discussion, Dyadic interviews, Grammarly.com		
Project work/Assignment:		
Academic Journal – Assignment In Academic Journal (CIJ), students compile task and activities completed in each module and submit to the instructor at the middle and end of the semester.		
References Hering, Heik. How to Write Technical Reports: Understanding Structure, Good Design, Convincing Presentation. Springer. Johnson, Richard. (2010) Technical Communication Today. Pearson, 2015 Rice B. Adelrod, Charles R. Cooper and Ellen C. Carillo. (2020) Reading Critically Writing Well: A Reader and Guide. Beford/St. Martin's Macmillan Learning, New York. The Princeton Review. (2010) MCAT Verbal Reasoning & Writing. The Princeton Review, Inc. https://www.hitbullseye.com/Strong-and-Weak-Arguments.php Accessed on 10 Dec 2021 https://www.inc.com/guides/how-to-improve-your-presentation-skills.html Accessed on 10 Dec 2021		
Topics Relevant to "employability": Critical Reasoning, Presentation, Review Writing and Starting Career Topics Relevant to "Human Values and Professional Ethics": Critical reasoning		

Course Code: PPS1012	Course Title: Enhancing Personality through Soft Skills Type of Course: Practical Only Course	L- T - P- C	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.					
Anti-requisites	NIL					
Course Description	This course is designed to enable students understand soft skills concepts and improve confidence, communication and professional skills to give the students a competitive advantage and increase chances of success in the professional world. 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 “Personality Development through Soft Skills” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.					
Course Out Comes	On successful completion of this course the students shall be able to: CO 1 Identify the stages of team formation (Remember) CO 2 Demonstrate effective presentation skills (Apply) CO3 Prepare professional social media profile (Apply)					

Course Content:			
Module 1	Team Building	Classroom and outbound team building activities.	6 Hours
Topics: Importance of team, stages of Team Formation, Trust and collaboration, Virtual Team. Activity: Team Building outbound activity			
Module 2	Art of Questioning	Role plays	4 Sessions
Topics: Framing Questions, 5W1H Technique, Open-ended and Close-ended questions, Funnel technique, Probing questions, Leading questions			
Module 3	Presentation Skills	Practice and evaluation of individual / group presentation	10 Sessions
Topics: Content development, Delivery techniques, Audience Analysis, Timing and Pacing, handling questions and challenges. Activity: Individual presentations and team presentation			
Module 4	Professional Brand Building	Brand Framework Activity	4 Sessions
Topics: Personal brand definition, Crafting a compelling LinkedIn profile, Networking strategies. Activity: Create a basic online profile			
Module 5	Recap / Revision /Feedback Session		1 Session
Targeted Application & Tools that can be used: TED Talks You Tube Links Activities			

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course
Presentation Evaluation
Targeted Application & Tools that can be used: TED Talks YouTube Links Videos by L&D Team shared on Edhitch/YouTube.com LMS Assignments proposed for this course Evaluation on Presentation Assignment on LinkedIn Post YouTube Links: https://youtu.be/z__jxoczNWc (Steve Jobs Introducing the iPhone 4 in June 2010) References “Talk Like TED - The 9 Public-Speaking Secrets of the World's Top Minds” By Carmine Gallo St. Martin's Press Copyright © 2014 Carmine Gallo All rights reserved. ISBN: 978-1-250-04112-8 “The Presentation Secrets of Steve Jobs: How to Be Insanely Great in Front of Any Audience” MP3 CD – Import, 22 April 2014 “The Definitive Book of Body Language: The Hidden Meaning Behind People's Gestures and Expressions” Hardcover – Illustrated, 25 July 2006 “Crucial Conversations: Tools for Talking When Stakes Are High” Paperback – Import, 1 July 2002 Web links: https://www.wordstream.com/blog/ws/2014/11/19/how-to-improve-presentation-skills https://www.cbs.de/en/blog/15-effective-presentation-tips-to-improve-presentation-skills/ https://hbr.org/2022/05/the-art-of-asking-great-questions Topics relevant to development of “SKILL”: Art of Presentation, Team building, Art of questioning and Personal Branding for Skill Development through Participative

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

Course Code: EEE1007	Course Title: Basics of Electrical and Electronics Engineering. Type of Course: Engineering Science - Theory & Integrated Laboratory	L-T-P-C	3	0	2	4
Version No.	1.0					
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 emphasis on the characteristics and applications of Electrical and Electronics devices, working, analysis and design of electrical circuits using both active & passive components, fundamentals of electrical machines and basics of transistors and its application. The associated laboratory provides an opportunity to validate the concepts taught and enhances the ability to visualize the 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.					
Course Outcomes	On successful completion of this course the students shall be able to: Explain basic laws of Electrical Engineering to compute voltage, currents and other parameters in the circuits. Discuss various fundamental parameters appearing in the characteristics of semiconductor devices and their applications. Summarize the operations of different biasing configurations of BJTs and amplifiers. Summarizethe performance characteristics and applications of various electrical Machines.					

	Demonstrate the working of electrical machines to observe performance characteristics Demonstrate the working of electronic circuits to obtain the V-I Characteristics of various semiconductor devices.			
Course Content:				
Module 1	Introduction to Electrical Circuits	Assignment/ Quiz	Numerical solving Task	10 Sessions
DC Circuits: Concept of Circuit and Network, Types of elements, Network Reduction Techniques- Series and parallel connections of resistive networks, Star–to-Delta Transformations, Mesh Analysis, 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	Assignment/ Quiz	Memory Recall based Quizzes	11 Sessions
Mass Action Law, Charge densities in a semiconductor, Types of SC, Junction diodes - Ideal and practical behaviour, Modelling the Diode Characteristic, and Diode applications like rectifiers, Clipping and clamping circuits. Zener diode, characteristics and its applications like voltage regulator.				
Module 3	Fundamentals of Electrical Machines	Assignment/ Quiz	Memory Recall-based Quizzes	12 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. Special Machines: Introduction to special electrical machines and its applications.				
Module 4	Transistors and its Applications	Assignment/ Quiz	Numerical solving Task	12 Sessions
Transistor characteristics, Current components, BJT Configurations (CB, CC, CE configurations) and their current gains. Operating point, Biasing & stabilization				

techniques: Fixed Bias, Voltage divider bias and its stability factor and load line analysis. Single and multistage amplifier, Darlington pair.

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.

List of Laboratory Tasks:

Experiment No 1: Verification of KVL and KCL for a given DC circuit.

Level 1: Study and Verify KVL and KCL for the given electrical Circuit.

Level 2: For the same circuit considered in level 1, perform the simulation using NI LabVIEW/Multisim/MATLAB.

Experiment No 2: Analyse AC series circuits – RL, RC and RLC .

Level 1: Conduct an experiment to perform and verify the impedance, current and power of Series RL and RC circuits

Level 2:

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 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 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 Zin input impedance and Zout 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: Matlab/Multisim/ PSpice

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..

Text Book(s):

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

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.

A.P.Malvino, Electronic Principles,7thEdition, Tata McGraw Hill,2007

J. Millman, C. C. Halkias and C. D. Parikh, “Millman’s Integrated Electronics”, McGraw Hill Education, 2nd Edition.

Basics of Electrical & Electronics Laboratory Manual.

Reference Book (s):

John Hiley, Keith Brown and Ian McKenzie Smith, "HUGHES Electrical and Electronic Technology", 10th Edition (Indian Edition published by Dorling Kindersley), Pearson,2011

Samarajit Ghosh, "Fundamentals of Electrical and Electronics Engineering", 2nd Edition, Prentice Hall India, 2007.

K Uma Rao, A Jaya Lakshmi, “Basic Electrical engineering” IK International publishing house Pvt. Ltd

R. L. Boylestad and L. Nashelsky, “Electronic Devices and Circuit Theory”, Pearson Education India 7th Edition.

A K. Maini, V. Agrawal, “Electronic Devices & Circuits”, Wiley, 2nd Edition

A.S Sedra, K. C. Smith, “Microelectronic Circuits”, Oxford University Press, 6th Edition

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

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

https://www.digimat.in/nptel/courses/video/108105112/L01 “Fundamentals of Electrical Engineering-Basic Concepts, Examples”

Seminar Topic: https://nptel.ac.in/courses/108/105/108105153/ “Electrical Measurements”

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

Video lectures on “Analog Electronics” by Prof. S.C. Dutta Roy, IIT Delhi https://nptel.ac.in/courses/108/102/108102095/

Video lectures on “Diodes”, by Prof.ChitralekhaMahanta, IIT Guwahati,

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

“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 CurrentVoltageModeling," 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

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

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

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

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 Electrical and electronics circuit parameters, performance operation of Machines, and semiconductor devices for Skill Developmentthrough Experiential Learning techniques. This is attained through assessment component mentioned in course plan.

Course Code: CSE1006	Course Title: Problem Solving using JAVA Type of Course: Integrated	L- T- P- C	1	0	4	3
Version No.	2.0					
Course Pre-requisites	CSE1004 – Problem Solving Using C					

Anti-requisites	Nil			
Course Description	This course introduces the core concepts of object-oriented programming. This course has theory and lab component which emphasizes on understanding the implementation and application of object-oriented programming paradigm. It helps the student to build real time secure applications by applying these concepts and also for effective problem solving. The students interpret and understand the need for object oriented programming to build applications.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of Problem-Solving using JAVA and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques			
Course Out Comes	On successful completion of the course the students shall be able to: C.O. 1: Describe the basic programming concepts. [Knowledge] C.O. 2: Apply the concept of classes, objects and methods to solve problems. [Application] C.O. 3: Apply the concept of arrays and strings. [Application] C.O. 4: Implement inheritance and polymorphism building secure applications. [Application] C.O. 5: Apply the concepts of interface and error handling mechanism. [Application]			
Course Content:				
Module 1	Basic Concepts of Programming and Java	Assignment	Data Collection/Interpretation	12 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	Case studies / Case let	Case studies / Case let	12 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	Quiz	Case studies / Case let	14 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	Quiz	Case studies / Case let	14 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	Quiz	Case studies / Case let	14 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.				

List of Laboratory Tasks:

P1 - Problem Solving using Basic Concepts.

P2 - Problem Solving using Basic Concepts and Command Line Arguments.

P3 - Programming assignment with class, objects, methods and Constructors.

P4 - Programming assignment with method overloading.

P5 - Programming assignment with constructor overloading.

P6 - Programming assignment with Static members and static methods.

P7 - Programming assignment with Nested classes.

P8 - Programming assignment using Arrays.

P9 - Programming assignment using Strings.

P10 - Programming assignment using String Builder.

P11 - Programming assignment using Inheritance and super keyword.

P12 - Programming assignment using Method overriding and Dynamic method invocation.

P13 - Programming assignment using Final keywords.

P14 - Programming assignment using Abstract keywords.

P15 - Programming assignment using Interface.

P16 - Programming assignment using Interface.

P17 - Programming assignment CharacterStream Classes

P18 - Programming assignment Read/Write Operations with File Channel

Targeted Application & Tools that can be used : JDK /eclipse IDE/ net Beans IDE.

Text Book

T1 Herbert Schildt, “The Complete Reference Java 2”, Tata McGraw Hill Education.

References

R1: Cay S Horstmann and Cary Gornell, “CORE JAVA volume I-Fundamentals”, Pearson

R2: James W. Cooper, “Java TM Design Patterns – A Tutorial”, Addison-Wesley Publishers.

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”:

Static Polymorphism

Method overloading, constructors

constructor overloading

this keyword

static keyword and Inner classes

Inheritance and Polymorphism.

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

Course Code: FRL1002	Course Title: Basic French Type of Course: Open Elective	L- T-P- C	2-0-0-2

Version No.	4.0			
Course Pre-requisites	Not Applicable			
Anti-requisites	Not Applicable			
Course Description	This Course is for beginners and gives an introduction of the French Language (basic grammar, conjugation, daily used vocabulary words, and basic conversations) and French culture. This Course is designed to build up all of the basic skills of French listening, reading, speaking, and writing introduced in the lessons. Besides, this Course offers an access to the French world, helping students to break cultural boundaries and raise cultural literacy.			
Course Objective	This course is designed to improve the learners Employability skills by using participative learning techniques to develop students' language proficiency and cross-cultural competence by active and participatory teaching methods.			
Course Outcomes	On successful completion of the course the students shall be able to: Identify the basics of French Grammar, vocabulary and Conjugation Apply the basics strategies of listening, reading, speaking and writing skills Use of French on everyday topics such as greetings, personal information, time and schedule Practice conversations in French language with peer speakers in different situations			
Course Content:	Learning of Basic French skills			
Module 1	Greetings and Introducing yourself		[Remember]	6 Periods
Chapter 1. Greetings Objectives: Greetings, introducing yourself, how to welcome someone, Grammar: Construction of a sentence, the days of the weeks and the months Chapter 2. Introducing yourself Objectives: Introduce oneself / ask for someone's personal information. Grammar: Mas or Fem noun, adjectives, present tense of the 1st group Usage of audio visual files				

Module 2	Expressing likes/dislikes and introducing someone		[Apply]	6 Periods
Chapter 3. Expressing likes and dislikes Objectives: How to expressing what you like and dislike. Grammar: Negative form, singular and plural. Culture: The polite way to address people in French Assignment Chapter 4. Introducing someone Objectives: How to describe someone, Grammar: Vocabulary of the family, Demonstrative adjectives, Present tense of verbs of the 2nd and 3rd group				
Module 3	Inviting someone and asking questions		[Apply]	9 Periods
Chapter 5. Inviting someone Objectives: How to invite someone, accept or refuse the invitation, Read the time, Grammar: Future tense, Interrogation. Culture: The art of accepting and declining an invitation politely in French Internal Chapter 6. Asking for information, Objectives: How to ask for information, giving information				
Module 4	Making a reservation and giving directions		[Apply]	9 Periods
Chapter 7: Making a Reservation Objectives: How to make a reservation, future tense Chapter 8 : Giving directions Objectives: How to ask for directions, Imperative tense Group discussions				
Targeted Application & Tools that can be used				

Project work /Assignments Assignment (Essay writing / presentation) Internal Group work / Group discussions
Text Book L’Atelier 1 - - Méthode de Français--- Niveau A1 (Didier – 2019) Festival 1- - Méthode de Français--- Niveau A1 (CLE International – 2005)
References Learning materials designed by the instructor
Topics relevant to development of ‘Employability Skills’ through participative learning techniques: Foreign language proficiency and cross-cultural competence by active and participatory teaching methods.

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

ECE2010</td><td colspan="4">Course Title: Innovative Projects using Arduino</td><td>L- T- P- C</td><td>-</td><td>-</td><td>-</td><td>1</td></tr>
<tr><td>Version No.</td><td colspan="9">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="9">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="9">NIL</td></tr>
<tr><td>Course Description</td><td colspan="9">This course is designed to provide an in-depth understanding of Arduino microcontrollers and their application in various real time projects involving sensors. Throughout the course, students will learn the fundamentals of Arduino programming and gain hands-on experience with a wide range of sensors. Students will explore how to connect and interface sensors with Arduino 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 and sensors.</td></tr>
<tr><td>Course Objective</td><td colspan="9">The objective of the course is Employability Skills of student by using PARTICIPATIVE LEARNING techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="9">On successful completion of the course the students shall be able to

Explain the main features of the Arduino prototype board

Demonstrate the hardware interfacing of the peripherals to Arduino system.

Understand the types of sensors and its functions

Demonstrate the functioning of live projects carried out using Arduino system.</td></tr>
<tr><td>Course Content:</td><td colspan="9"></td></tr>
<tr><td>Module 1</td><td colspan="3">Basic concepts of Arduino</td><td>Hands-on</td><td colspan="2">Interfacing Task and Analysis</td><td colspan="2">4 Sessions</td></tr>
<tr><td colspan="10">Topics:

Introduction to Arduino, Pin configuration and architecture, Device and platform features, Concept of digital and analog ports, Familiarizing with Arduino Interfacing Board, API’s , Introduction to Embedded C and Arduino platform, Arduino Datatypes and variables, Arduino i/o Functions, Arduino Communications, Arduino IDE, Various Cloud Platforms.</td></tr>
</table>

Module 2	Sensory Devices	Hands-on	Interfacing Task and Analysis	4 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 Tinkercad Simulator.

Topics: Types of Arduino boards, 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 and sensors can be applied. The flexibility and affordability of Arduino, combined with the wide range of sensors available, allow for endless possibilities in creating innovative projects.

Professionally Used Software: students can use open SOURCE Softwares Arduino IDE and Tincker CAD

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

Textbook(s):

Monk Simon “Programming Arduino: Getting Started with Sketches”, Mc Graw Hill Publications Second Edition

References

Reference Book(s)

1. Neerparaj Rai “Arduino Projects for Engineers” BPB publishers,first edition, 2016.

2. Ryan Turner ”Arduino Programming ” Nelly B.L. International Consulting Ltd. first edition,2019.

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

Arduino trending Projects < https://www. https://projecthub.arduino.cc/>

Introduction to Arduino < https://onlinecourses.swayam2.ac.in/aic20_sp04/preview>

Case studies on Wearable technology< https://www.hticiitm.org/wearables>

E-content:

Cattle Health Monitoring System Using Arduino and IOT (April 2021| IJIRT | Volume 7 Issue 11 | ISSN: 2349-6002)

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.

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.

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.

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

Course Code: MAT2501	Course Title: Integral Transforms and Partial Differential Equations Type of Course:1] School Core	L-T- P- C	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Calculus and Differential Equations					
Anti-requisites	NIL					
Course Description	This course aims to introduce various transform techniques such as Laplace transform, Fourier transform and Z-transform in addition to expressing functions in terms of Fourier series. The course covers applications of Laplace transform to LCR circuits and solutions of different equations using Z-transform. The course also deals with the analytical methods for solving partial differential equations and the classical applications of partial differential equations.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Transform Techniques, Partial Differential Equations” and attain Skill Development through Problem Solving Techniques.					
Course Out Comes	On successful completion of the course the students shall be able to: CO1 - Express functions in terms of uniformly convergent Fourier series. CO2 - Apply Laplace transform technique to solve differential equations. CO3 - Employ Z-transform techniques to solve difference equations. CO4 - Solve a variety of partial differential equations analytically.					
Course Content:						
Module 1	Laplace Transforms		(12 Classes)			
Definition and Laplace transform of elementary functions. Properties of Laplace transform, and Laplace transform of periodic function, unit-step function and Impulse function – related problems. Inverse Laplace transform of standard functions -						

problems, initial and final value theorem. Convolution theorem, solution of linear and simultaneous differential equations and LCR Circuit.

Module 2	Fourier Series	Assignment	(8 Classes)

Fourier Series: Periodic functions, Dirichlet's condition. Fourier series of periodic functions period 2π and arbitrary period. Half range Fourier series. Practical harmonic analysis.

Module 3	Fourier Transforms and Z - Transforms		(13 Classes)

Fourier Transforms: Definitions, infinite Fourier transforms, Fourier sine and cosine transforms, inverse Fourier transforms, Problems.

Difference equations and Z-transforms: Z-transforms – Basic definitions, Standard Z-transforms, Linearity property, Damping rule, Shifting rule, Initial value theorem, Final value theorem, Inverse Z-transforms. Difference equations – Basic definitions, Application of Z-transforms to solve difference equations.

Module 4	Partial Differential Equations	Assignment	(12 Classes)

Formation of PDE, Solution of non-homogeneous PDE by direct integration, Solution of homogeneous PDE involving derivative with respect to one independent variable only (Both types with given set of conditions) Method of separation of variables. (First and second order equations) Solution of Lagrange's linear PDE. of the type P p + Q q = R.

Applications of PDE: Derivation of one-dimensional wave and heat equations. Various possible solutions of these by the method of separation of variables. D'Alembert's solution of wave equation. Two-dimensional Laplace's equation – various possible solutions. Solution of all these equations with specified boundary conditions (Boundary value problems).

Targeted Application & Tools that can be used:

The objective of the course is to familiarize students with a variety of numerical techniques and the theoretical concepts of probability and statistics to equip them with the necessary numerical approaches and basic statistical tools to tackle engineering and real-life problems.

Assignment:

Newton-Raphson Methods, Gauss-Seidel Method, LU Decomposition, Trapezoidal Rule, Simpson's rule, Runge-Kutta 4th Order.

Text Book

Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition

B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

References:

Victor Henner, Tatyana Belozerova, Mickhail Khenner, Ordinary and Partial Differential Equations, CRC Press, Edition, 2013.

Walter Ledermann, Multiple integrals, Springer, 1st edition

E-resources/ Web links:

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_140238

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_233298

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_204892

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_246791

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_223548

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_134719

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_32614

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: CDV1700
Course Title: DevOps Foundations
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description

This course provides foundational knowledge of DevOps culture, principles, tools, and practices. It covers version control, continuous integration and delivery (CI/CD), containerization, cloud deployment, and infrastructure as code. Students gain an understanding of how DevOps improves collaboration, automation, and software delivery lifecycle in modern development environments.

Course Objectives

— Understand the principles and lifecycle of DevOps practices
— Explore version control, build tools, and CI/CD pipelines
— Learn containerization, orchestration, and cloud deployment basics
— Apply DevOps tools to automate and streamline the software development lifecycle

Course Outcomes
CO1 (Understand): Describe DevOps principles, processes, and benefits
CO2 (Analyze): Examine CI/CD practices and DevOps toolchains
CO3 (Apply): Implement version control, automated builds, and test pipelines
CO4 (Apply): Deploy containerized applications using basic orchestration

Course Content (45 Hours Total)

Module 1: Introduction to DevOps and Culture – 11 Sessions (Understand)
Evolution of DevOps, Agile and Lean influence, DevOps lifecycle, CALMS framework, Dev vs Ops mindset, DevOps metrics, Toolchain overview, DevSecOps introduction

Module 2: Version Control and CI/CD – 11 Sessions (Analyze)
Git basics and branching, GitHub/GitLab workflows, Build automation (Maven, Gradle), Continuous Integration concepts, Jenkins, GitHub Actions, Continuous Delivery pipelines, Code quality and static analysis

Module 3: Containerization and Deployment – 11 Sessions (Apply)
Containers vs VMs, Docker architecture, Dockerfile and images, Docker Compose, Container registry, Application deployment and rollback, Kubernetes overview

Module 4: Infrastructure and Monitoring – 12 Sessions (Apply)
Infrastructure as Code (IaC) with Terraform/Ansible, Cloud provisioning (AWS/GCP/Azure basics), Monitoring and logging with Prometheus, ELK Stack, Alerts and dashboards, DevOps use cases in industry

Textbooks
T1: Emily Freeman, *DevOps for Dummies*, Wiley, **2023**
T2: Mikael Krief, *Learning DevOps: Continuously Deliver Better Software*, Packt Publishing, **2023**

Reference Books
R1: Gene Kim, Jez Humble, Patrick Debois, *The DevOps Handbook*, IT Revolution, **2022**
R2: Len Bass et al., *DevOps: A Software Architect's Perspective*, Addison-Wesley, **2021**
R3: Viktor Farcic, *DevOps Tools for Java Developers*, O'Reilly Media, **2022**
R4: Jonathan Baier, *Getting Started with Kubernetes*, Packt Publishing, **2022**

Web Resources
W1: https://docs.docker.com
W2: https://www.jenkins.io
W3: https://learn.github.com
W4: https://kubernetes.io
W5: https://roadmap.sh/devops

<table>
<tr><td>Course Code:
CSE2502</td><td colspan="3">Course Title: Operating Systems

Type of Course: Program Core and Theory Only</td><td>L-T- P-C</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">CSE2009- Computer Organization, Problem solving using C

Students should have basic knowledge on computers, computer software & hardware, and Computer Organization. Prior programming experience in C is recommended.</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course introduces the concepts of operating system operations, operating system structure and its design and implementation. It covers the classical operating systems internal algorithms such as process scheduling, synchronization, deadlocks detection and

recovery and memory management. The course also enhances the problem solving, systems programming ability and case studies.</td></tr>
<tr><td>Course Object</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Operating Systems and attain Employability through Problem Solving Methodologies.</td></tr>
<tr><td>Course Out Comes</td><td colspan="8">On successful completion of the course the students shall be able to:

1] Describe the fundamental concepts of operating Systems and case studies. [Knowledge]
2] Demonstrate various CPU scheduling algorithms. .[Application]
3] Apply various tools to handle synchronization problems.[Application]4] Demonstrate deadlock detection and recovery methods [Application]
5] Illustrate various memory management techniques.[Application]</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td colspan="2">Introduction to Operating System</td><td colspan="2">Assignment</td><td colspan="2">Programming</td><td colspan="2">9 Hours</td></tr>
<tr><td colspan="9">Topics:

Introduction to OS , Operating-System Operations, Operating System Services, , System Calls and its types,

Operating System Structure, System Program and its types, Linkers and Loaders, Overview of OS design and implementation, Open-source operating system</td></tr>
<tr><td>Module 2</td><td colspan="2">Process Management</td><td colspan="2">Assignment/Case Study</td><td colspan="2">Programming/Simulation</td><td colspan="2">11 Hours</td></tr>
</table>

Topics:

Process Concept, Operations on Processes, Inter Process Communication, Communication in client-server systems (sockets, RPC, Pipes), Introduction to threads - Multithreading Models, Thread Libraries, Threading Issues, Process Scheduling– Basic concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF, RR and Priority.

Module 3	Process Synchronization and Deadlocks	Assignment	Programming	**11 Hours**

Topics:

The Critical-Section Problem- Peterson's Solution, Synchronization hardware, Semaphores, Classic Problems of Synchronization with Semaphore Solution- Producer-Consumer Problem, Reader-Writer problems, Dining Philosopher's Problem, . Introduction to Deadlocks, Necessary conditions for deadlock, Resource allocation Graph, Methods for handling deadlock: Deadlock Prevention and Implementation, Deadlock Avoidance and

Implementation, Deadlock detection & Recovery from Deadlock.

Module 4	Memory Management	Assignment	Programming/Simulation	**10 Hours**

Topics:
Introduction to Memory Management, Basic hardware-Base and Limit Registers, Memory Management Unit(MMU), Dynamic loading and linking, Swapping, Contiguous and Non-Contiguous Memory Allocation, Segmentation, Paging - Structure of the Page Table – Virtual Memory and Demand Paging – Page Faults and Page Replacement Algorithms, Copy-on-write, Allocation of Frames, Thrashing
Introduction to File system management: File System Interface (access methods, directory structures), File system implementation.

Targeted Application:
Application area is traffic management system, banking system, health care and many more systems wherein 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.

Project work/Assignment

. Demonstrate process concepts in LINUX OS.
. Simulation of CPU scheduling algorithms.
. Develop program to demonstrate use of Semaphores in threads.
. Develop program to demonstrate use of deadlock avoidance algorithms.
. Develop program to demonstrate use of page replacement algorithms.
. 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
2.

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

5. https://www.os-book.com/OS9/
6. https://pages.cs.wisc.edu/~remzi/OSTEP/
7. https://codex.cs.yale.edu/avi/os-book/OS10/index.html

Course Code: CSE1508	**Course Title:** Data Structures **Type of Course: Theory**		3		0	3 3
Version No.	1.0					
Course Pre-requisites						
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	The objective of the course is SKILL DEVELOPMENT of student by using EXPERIENTIAL LEARNING techniques					
Course Out Comes	**On successful completion of the course the students shall be able to:** **CO1 :** Describe the concept of basic data structure, stacks, queues, and arrays and their operations. **[Understand]** **CO2:** Utilize linked lists for real-time scenarios. **[Apply]**					

	CO3: Apply an appropriate non-linear data structure for a given scenario. **[Apply]** **CO4:** Demonstrate different searching and sorting techniques. **[Apply]**			
Course Content:				
Module 1	**Introduction to Data Structure and Linear Data Structure – Stacks and Queues**	Assignment	Program activity	**9 Hours**
Introduction –Introduction to Data Structures, Types and concept of Arrays . **Stack** –Concepts and representation, Stack operations, stack implementation using array and Applications of Stack. **Queues** –Representation of queue, Queue Operations, Queue implementation using array, Types of Queue and Applications of Queue.				
Module 2	**Linear Data Structure – Linked List**	Assignment	Program activity	**12 Hours**
Topics: Linked List - Singly Linked List, Operation on linear list using singly linked storage structures, Circular List, Applications of Linked list. **Recursion** - Recursive Definition and Processes.				
Module 3	**Non-linear Data Structures – Trees**	Assignment	Program activity	**12 Hours**
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 – AVL Trees ,Binary Serach Tree , .Heaps , Expression Tree ,Red Black Tree				
Module 4	**Non-linear Data Structures**	Assignment	Program	**6 Hours**

	– Graphs and Hashing		activity	

Topics: Graphs: Basic Concept of Graph Theory and its Properties, Representation of Graphs . ADT, Elementary graph operations, Minimum Cost spanning trees, Shortest path and Transitive closure.

Hashing: Introduction, Static Hashing, Dynamic Hashing

Module 5	**Searching & Sorting**	Assignment	Program activity	**6 Hours**

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

List of Laboratory Tasks:

Lab sheet -1

Level 1: Prompt the user, read input and print messages.Programs using class, methods and objects

Level 2: Programming Exercises on fundamental Data structure - Arrays based on Scenario.

Lab sheet -2

Level 1: Programming Exercises on Stack and its operations

Level 2: Programming Exercises on Stack and its operations with condition

Lab sheet -3

Level 1: Programming on Stack application infix to postfix Conversion

Level 2: -

Lab sheet -4

Level 1: Programming on Stack application – Evaluation of postfix

Lab sheet -5

Level 1: Programming Exercises on Queues and its operations with conditions

Level 2: -

Lab sheet -6

Level 1: Programming Exercises on Linked list and its operations.

Level 2: Programming Exercises on Linked list and its operations with various positions

Lab sheet -7

Level 1: Programming Exercises on Circular Linked list and its operations.

Level 2: Programming Exercises on Circular Linked list and its operations with various positions

Lab sheet -8

Level 1: Programming Exercises on factorial of a number

Level 2: Programming the tower of Hanoi using recursion

Lab sheet -9

Level 1: -

Level 2: Programming the tower of Hanoi using recursion

Lab sheet -10

Level 1: Programming Exercise on Doubly linked list and its operations

Level 2: -

Lab sheet -11

Level 1: Program to Construct Binary Search Tree and Graph

Level 2: Program to traverse the Binary Search Tree in three ways)in–order, pre–order and post–order(and implement BFS and DFS

Lab sheet -12

Level 1: Program to Implement the Linear Search & Binary Search

Level 2: Program to Estimate the Time complexity of Linear Search

Lab sheet -13

Level 1: Program to Implement and Estimate the Time complexity of Selection Sort

Level 2: Program to Implement and Estimate the Time complexity of Insertion Sort

Lab sheet -14 (Beyond syllabus activity)
Level 1: Program to Construct AVL Tree
Level 2:
Lab sheet -15 (Beyond syllabus activity)
Level 1: Program to Construct RED BLACK Tree

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: **CSE2501**	**Course Title: Computer Organization and Architecture** **Type of Course: Program Core, Theory based**	**L-T-P-C**	3	0	0	3
Version No.	2.0					
Course Pre-requisites	CSE2015 - Digital Design					
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		The objective of the course is to familiarize the learners with the concepts of **Computer Organization and Architecture** and attain **Skill Development** through **Participative Learning** techniques.
Course Outcomes		On successful completion of the course the students shall be able to: 1] Describe the basic components of a computer and their interconnections. [Remember] 2] Explain Instruction Set Architecture and Memory Unit [Understand] 3] Apply appropriate techniques to carry out selected arithmetic operations [Apply] 4] Explain the organization of memory and processor sub-system [Understand]
Course Content:		

Module 1	Basic Structure of computers	Assignment	Data Analysis task	**12 Sessions**

Topics:

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

Memory Instructions.

Module 2	Instruction Set Architecture and Memory Unit	Assignment	Analysis, Data Collection		**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	Case Study	Data analysis task		**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	Assignment	Analysis, Data Collection		**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. Web References: 1. NPTEL Course on “Computer architecture and organization” IIT Kharagpur By Prof. Indranil Sengupta, Prof. Kamalika Datta. https://nptel.ac.in/courses/106105163 2. NPTEL Courseon “Computer Organization”, IIT Madras By Prof. S. Raman. https://nptel.ac.in/courses/106106092 3. 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: CSE1500	**Course Title: Computational Thinking using Python** **Type of Course: Engineering Science Theory Integrated**	**L-T-P-C**	2	0	0	2
Version No.	1.0					
Course Pre-requisites	• •					
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 Object	The objective of the course is to familiarize the learners with the concepts of **Computational Thinking using Python** and attain **Skill Development** through **Participative Learning** techniques.				
Course Out Comes	On successful completion of the course the students shall be able to: 1) Describe algorithmic solutions for basic computing issues.. (Understand) 2) Explain data types and operators. (Understand) 3) Demonstrate control structures and Functions. (Apply) 4) Apply the data structures for the given data. (Apply) 5) Demonstrate the file operations. (Apply)				
Course Content:					
Module 1	**Computational Thinking And Problem Solving**	Assignment		Programming	**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**	Assignment		Programming	**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**	Assignment		Programming	**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**	Assignment		Programming	**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**	Assignment		Programming	**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).
	Project work/Assignment:
1.	**2. Assignment 1 on (Module 1 and Module 2)** **3. Assignment 2 on (Module 3 and Module 4 & 5)**
	Text Book 1) Paul Deitel and Harvey Deitel, "Python for Programmers", Pearson Education, 1st Edition, 2021 2) Eric Matthes, Python Crash Course,: A Hands-On, Project-Based Introduction to Programming, 3rd Edition, 2023

	References 1.Allen B. Downey, "Think Python: How to Think like a Computer Scientist", 2nd Edition, O'Reilly Publishers, 2016. 2. Karl Beecher, "Computational Thinking: A Beginner's Guide to Problem Solving and Programming", 1st Edition, BCS Learning & Development Limited, 2017. **Web Resources** W1. https://onlinecourses.nptel.ac.in/noc20_cs70/preview
	Topics relevant to development of "Employability": Data structures using python. **Topics relevant to "PROFESSIONAL ETHICS":** Naming and coding convention for simple programs using python.

Course Code: **CSE1501**	**Course Title: Computational Thinking using Python Lab** **Type of Course: Engineering Science Laboratory Integrated**		**L - T - P - C**	0	0	4	2
Version No.		1.0					
Course Pre-requisites	•	•					
Anti-requisites		NIL					
Course Description		The course efficiently introduces fundamental ideas and practical knowledge including control structures, functions, and tuples through hands on sessions. It also introduces dynamic programming like handling exceptions and file operations. The course covers Python dictionaries, classes, and objects for creating					

		user-defined datatypes, such binary search, in terms of data structures.
Course Object		The objective of the course is to familiarize the learners with the concepts of **Computational Thinking using Python Lab** and attain **Skill Development** through **Experiential Learning** techniques.
Course Out Comes		On successful completion of the course the students shall be able to: 6) Explain algorithms to solve fundamental computational problem. (Understand) 7) Illustrate the use of different data types and operators in Python. (Apply) 8) Demonstrate conditionals, loops, and functions to address problem-solving tasks. (Apply) 9) Utilize appropriate data structures to efficiently manage and process data. (Apply) 10)Perform file handling tasks such as reading, writing, and modifying files in Python.. (Apply)
Course Content:		

Module 1	**Computational Thinking And Problem Solving**	Assignment		Programming	**6 Sessions**

Lab sheet :

Introduction to Python Programming.

Demonstration of Colabs or Jupiter environment.

Demonstrate Input function.

Demonstrate int method.

Demonstrate data types

Demonstrate operators.

Demonstrate simple programs for python environment.

Python program that takes an integer input and calculates the sum of its digits.

Module 2	**Datatypes, Expressions, Statements**	Assignment		Programming	**14 Sessions**

Topics:

Python program to count the number of times a given character appears in a string.

Python program to identify the data types of given variables.

A grocery store needs a billing system, write a python program that calculates the total bill amount based on the price of individual items and their quantities.

A car rental company wants to calculate the mileage (km per liter) for different vehicles based on distance traveled and fuel consumed. Write a Python program to calculate the mileage.

A company wants to calculate the net salary of an employee after deducting tax (10%) and provident fund (5%) from the gross salary. Write a Python program to calculate net salary.

In a student grading system where the final grade depends on whether the student has passed both the written and practical exams. You need to check if the student has passed based on certain conditions:

- **The student must score at least 40 in the written exam.**
- **The student must score at least 50 in the practical exam.**
- **The student must have attended at least 75% of the classes**

Write a Python program to check if a person is eligible to vote. The criteria are:

- **The person must be at least 18 years old.**
- **The person should be a citizen of the country.**

Write a Python program to classify a person into age groups:

- **Child: 0 to 12 years**
- **Teenager: 13 to 19 years**
- **Adult: 20 years and above**

A user authentication system that checks if the user is authorized based on certain conditions, like having a correct password and being over the age of 18. Write a Python program for the above scenario.

In a student registration system, a student must meet certain criteria to be eligible for course registration:

- **The student must have a GPA of 3.0 or above.**
- **The student must have completed the prerequisite course (True/False).**

Module 3	**Control flow, Functions, Strings**	Assignment		Programming	**16 Sessions**

Lab Sheet:

An e-commerce store that offers discounts based on the following criteria:

- **The customer must be a loyal customer (i.e., True).**
- **The total purchase amount must be greater than $100.**

Python program to print the Fibonacci sequence up to n terms

Python program to print the Fibonacci sequence up to n terms using Recursion.

Apply slicing on the given data or dictionary.

Python Programs to create array and print the array.

Python program to check if a given number is an Armstrong number. An Armstrong number for a 3-digit number is one where the sum of the cubes of its digits is equal to the number itself.

The media platform wants to count the number of words in user-submitted posts to enforce character limits or to analyze the length of posts. .

In a bookstore inventory system, You need to implement a feature that checks if a book title is a palindrome. The bookstore wants to offer special discounts for books with titles that are palindromes. You need to create a Python function that reads the book title and determines if it's a palindrome.

In a library management system. The library has a database of books identified by unique numbers (IDs). The library staff wants to apply a special offer to books whose IDs are prime numbers. You need to create a Python program that finds all prime numbers between a given range of book IDs.

In a school management system that stores the marks of students for each subject. You are asked to compute the average marks of a student to evaluate their overall performance. Write a Python program that takes the marks of a student in different subjects and calculates the average.

A small inventory list where you need to search for a specific product ID. Since the list isn't sorted, you can use linear search, which checks each element sequentially until it finds the target. Write a python program to perform linear search.

A sorted list of product IDs and need to quickly find a specific product. Binary search is ideal for this scenario because it efficiently narrows down the search space by repeatedly dividing the list into two halves.

Module 4	**Lists, Tuples, Dictionaries**	Assignment		Programming	**12 Sessions**

Lab Sheet:

Demonstrate List, Tuple and Dictionary.

A supermarket wants to maintain a list of available products and update it when new products arrive or old products are sold out.

A library maintains book records using dictionaries, where the book title is the key and the quantity available is the value.

A school stores student grades in a list, and the teacher wants to see only the top 3 grades.

A restaurant receives online orders in a queue (list) and processes the first 3 orders at a time. Write a Python program to handle orders using list slicing.

A university has course details stored in tuples. The system should extract and display only the course codes. Write Python program to extract course code from tuples.

A fitness tracking app stores a user's daily step count for a week and extracts steps from Monday to Friday. WAP to extract weekly steps using slicing.

A school stores student marks in a list. Write a program to:

- **Find the highest and lowest marks.**
- **Calculate the average marks.**
- **Count how many students scored above 75.**

A company maintains a list of employees' names. Write a program to:

- **Add a new employee to the list.**
- **Remove an employee from the list.**
- **Sort and display all employees in alphabetical order.**

A tuple stores flight details (Flight Number, Destination, Duration). Write a program to:

- **Display all flights.**
- **Find flights with a duration of more than 3 hours.**
- **Access the destination of a specific flight.**

A grocery store stores item details as tuples (Item Name, Price per kg). Write a program to:

- **Calculate the total bill for a customer.**
- **Find the cheapest item.**
- **Sort items by price in ascending order.**

Use Dictionaries: A library stores book records as {Book Title: Copies Available}. Write a program to:

- **Borrow a book (decrease count).**
- **Return a book (increase count).**
- **Display all available books.**

Use List Comprehension: A company stores employee ID numbers. Write a Python program to extract only the even employee IDs from a given list.

Module 5	**Files**	Assignment		Program ming	**12 Sessions**

Lab Sheet:

Write a Python program that asks for a voter's age. If the age is below 18, raise an exception "Invalid Age: Must be 18 or older".

Write a Python program that counts the total number of lines, words, and characters in a given text file.

Write a Python program that reads text file and finds the most repeated word.

Write a program that searches for a word in a file ".txt" and replaces it with another word.

Write a Python program that copies the content from "source.txt" to "destination.txt". If "destination.txt" does not exist, create it.

Write a Python program that takes two numbers as command-line arguments and prints their sum.

Write a Python program that asks for a user's name, age, and marks in three subjects, then formats and displays the result in a structured way. Generate report using string formatting.

	Create a module called "mymath.py" with functions add(a, b), subtract(a, b), and multiply(a, b). write a separate Python script that imports this module and uses these functions. **Write a Python program that tries to read a file ".txt". If the file is not found, catch the exception and display a message.**
	Project work/Assignment:
4.	**1.Assignment 1 on (Module 1 and Module 2)** **5. Assignment 2 on (Module 3 and Module 4 & 5)**
	Text Book 1Paul 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** W2. https://onlinecourses.nptel.ac.in/noc20_cs70/preview W3. https://onlinecourses.swayam2.ac.in/cec23_cs02/preview W4. https://www.coursera.org/learn/ai-python-for-beginners
	Topics relevant to development of "Employability": Data structures using python. **Topics relevant to "PROFESSIONAL ETHICS":** Naming and coding convention for simple programs using python.

Course Code: **CSE2000**	**Course Title: Software Design and Development** **Type of Course: School Core [Theory Only]**		**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 the fundamentals concepts of Software Engineering process and principles. The course covers software requirement engineering processes, system analysis, design, implementation and testing aspects of software system development. The course covers software quality, configuration management and maintenance.			
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Software Engineering and attain Skill Development through Participative Learning techniques.			
Course Out Comes	On successful completion of this course the students shall be able to: 1] Describe the Software Engineering principles, ethics and process models(Knowledge) 2] Identify the requirements, analysis and appropriate design models for a given application(Comprehension) 3] Understand the Agile Principles(Knowledge) 4] Apply an appropriate planning, scheduling, evaluation and maintenance principles involved in software(Application)			
Module 1	**Introduction to Software Engineering and Process Models** **(Knowledge level)**	Quiz		**10 Hours**

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.

Module 2	**Software Requirements, Analysis and Design** **(Comprehension level)**	Assignment	Development of SRS documents for a given scenario	**12 Hours**

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.

Module 3	**Agile Principles & Devops** **(Knowledge level)**	Quiz		**10 Hours**

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.

Module 4	**Software Testing and Maintenance** **(Application Level)**	Assignment	Apply the testing concepts using Programing	**13 Hours**

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.

Targeted Application & Tools that can be used: Selenium, GitHub, CASE Tools

Text Book

1] R1. Roger S. Pressman, “Software Engineering – A Practitioner’s Approach”, VII Edition, McGraw-Hill, 2017.

2] B2. Bob Hughes, Mike Cotterell, Rajib Mall, “Software Project Management”, VI Edition, McGraw-Hill, 2018.

References

1. Rajib Mall, “Fundamentals of Software Engineering”, VI Edition, PHI learning private limited, 2015.
2. Ian Sommerville, “Software Engineering”, IX Edition, Pearson Education Asia, 2011.
3. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002

Topics Relevant to “Skill Development: Balck box Testing, White box Testing, Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout

Course Code: **CSE2000**	**Course Title: Software Design and Development** **Type of Course: School Core [Theory Only]**	**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 the fundamentals concepts of Software Engineering process and principles. The course covers software requirement engineering processes, system analysis, design, implementation and testing aspects of software system development. The course covers software quality, configuration management and maintenance.		
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Software Engineering and attain Skill Development through Participative Learning techniques.		
Course Out Comes	On successful completion of this course the students shall be able to: 1] Describe the Software Engineering principles, ethics and process models(Knowledge) 2] Identify the requirements, analysis and appropriate design models for a given application(Comprehension)		

	3] Understand the Agile Principles(Knowledge) 4] Apply an appropriate planning, scheduling, evaluation and maintenance principles involved in software(Application)			
Module 1	**Introduction to Software Engineering and Process Models** **(Knowledge level)**	Quiz		**10 Hours**
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.				
Module 2	**Software Requirements, Analysis and Design** **(Comprehension level)**	Assignment	Development of SRS documents for a given scenario	**12 Hours**
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.				
Module 3	**Agile Principles & Devops** **(Knowledge level)**	Quiz		**10 Hours**
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.				
Module 4	**Software Testing and Maintenance**	Assignment	Apply the testing concepts using Programing	**13 Hours**

	(Application Level)			

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.

Targeted Application & Tools that can be used: Selenium, GitHub, CASE Tools

Text Book

1] R1. Roger S. Pressman, “Software Engineering – A Practitioner’s Approach”, VII Edition, McGraw-Hill, 2017.

2] B2. Bob Hughes, Mike Cotterell, Rajib Mall, “Software Project Management”, VI Edition, McGraw-Hill, 2018.

References

4. Rajib Mall, “Fundamentals of Software Engineering”, VI Edition, PHI learning private limited, 2015.
5. Ian Sommerville, “Software Engineering”, IX Edition, Pearson Education Asia, 2011.
6. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002

Topics Relevant to “Skill Development: Balck box Testing, White box Testing, Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout

Course Code: CSE1509	**Course Title:** Data Structures Lab **Type of Course:Lab**	**-T-P-C**	0	0	4	2
Version No.	1.0					
Course Pre-requisites						

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	The objective of the course is SKILL DEVELOPMENT of student by using EXPERIENTIAL LEARNING techniques
Course Out Comes	**On successful completion of the course the students shall be able to:** **CO1** : Describe the concept of basic data structure, stacks, queues, and arrays and their operations. **[Understand]** **CO2:** Utilize linked lists for real-time scenarios. **[Apply]** **CO3:** Apply an appropriate non-linear data structure for a given scenario. **[Apply]** **CO4:** Demonstrate different searching and sorting techniques. **[Apply]**
Course Content:	

Module 1	**Introduction to Data Structure and Linear Data Structure – Stacks and Queues**	Assignment	Program activity	**9 Hours**

Introduction –Introduction to Data Structures, Types and concept of Arrays .

Stack –Concepts and representation, Stack operations, stack implementation using array and Applications of Stack .

Queues –Representation of queue, Queue Operations, Queue implementation using array, Types of Queue and Applications of Queue .

Module 2	**Linear Data Structure – Linked List**	Assignment	Program activity	**12 Hours**
Topics: Linked List - Singly Linked List, Operation on linear list using singly linked storage structures, Circular List, Applications of Linked list. **Recursion** - Recursive Definition and Processes.				
Module 3	**Non–linear Data Structures –Trees**	Assignment	Program activity	**12 Hours**
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 – AVL Trees ,Binary Serach Tree , .Heaps , Expression Tree ,Red Black Tree				
Module 4	**Non–linear Data Structures – Graphs and Hashing**	Assignment	Program activity	**6 Hours**
Topics: Graphs: Basic Concept of Graph Theory and its Properties, Representation of Graphs . ADT, Elementary graph operations, Minimum Cost spanning trees, Shortest path and Transitive closure. **Hashing**: Introduction, Static Hashing, Dynamic Hashing				
Module 5	**Searching & Sorting**	Assignment	Program activity	**6 Hours**
Topic: Sorting & Searching - Sequential and Binary Search, Sorting – Selection and Insertion sort, Quick sort, Merge Sort, Bubble sort.				
List of Laboratory Tasks: **Lab sheet -1**				

Level 1: Prompt the user, read input and print messages.Programs using class, methods and objects

Level 2: Programming Exercises on fundamental Data structure - Arrays based on Scenario.

Lab sheet -2

Level 1: Programming Exercises on Stack and its operations

Level 2: Programming Exercises on Stack and its operations with condition

Lab sheet -3

Level 1: Programming on Stack application infix to postfix Conversion

Level 2: -

Lab sheet -4

Level 1: Programming on Stack application – Evaluation of postfix

Lab sheet -5

Level 1: Programming Exercises on Queues and its operations with conditions

Level 2: -

Lab sheet -6

Level 1: Programming Exercises on Linked list and its operations.

Level 2: Programming Exercises on Linked list and its operations with various positions

Lab sheet -7

Level 1: Programming Exercises on Circular Linked list and its operations.

Level 2: Programming Exercises on Circular Linked list and its operations with various positions

Lab sheet -8

Level 1: Programming Exercises on factorial of a number

Level 2: Programming the tower of Hanoi using recursion

Lab sheet -9

Level 1: -

Level 2: Programming the tower of Hanoi using recursion

Lab sheet -10

Level 1: Programming Exercise on Doubly linked list and its operations

Level 2: -

Lab sheet -11

Level 1: Program to Construct Binary Search Tree and Graph

Level 2: Program to traverse the Binary Search Tree in three ways) in–order, pre–order and post–order (and implement BFS and DFS

Lab sheet -12

Level 1: Program to Implement the Linear Search & Binary Search

Level 2: Program to Estimate the Time complexity of Linear Search

Lab sheet -13

Level 1: Program to Implement and Estimate the Time complexity of Selection Sort

Level 2: Program to Implement and Estimate the Time complexity of Insertion Sort

Lab sheet -14 (Beyond syllabus activity)
Level 1: Program to Construct AVL Tree
Level 2:
Lab sheet -15 (Beyond syllabus activity)
Level 1: Program to Construct RED BLACK Tree

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: **MAT2602**	**Course Title: Numerical Computations** **Type of Course:1] School Core**	**L-T-P- C**	**3**	**0**	**0**	**3**
Version No.		1.0				
Course Pre-requisites		Calculus, Linear Algebra, Differential Equations				
Anti-requisites		**NIL**				
Course Description		The course explores mathematical techniques used to approximate solutions to complex problems that are difficult to solve analytically, often utilizing computers to perform calculations, including methods for root finding, interpolation, numerical differentiation and integration, solving systems of linear equations, and approximating solutions to differential equations, with applications across various scientific and engineering fields. It focuses on understanding the theoretical basis behind these methods, their implementation in programming languages, and analyzing their accuracy and stability.				
Course Objective		The objective of the course is to equip students with understanding and ability to apply various numerical techniques to approximate solutions to complex mathematical problems that are difficult or impossible to solve analytically, particularly focusing on areas like solving systems of equations, finding roots of functions, interpolation, numerical differentiation, and integration, often utilizing computational tools to implement these methods.				

Course Out Comes		On successful completion of the course the students shall be able to: CO1 - Calculate errors induced in the values by truncation of a series expansion. CO2 - Demonstrate the applications of numerical methods to find the roots of polynomial equations and eigen values of real symmetric matrices. CO3 - Apply the knowledge of numerical methods in modelling of various physical and engineering phenomena. CO4 - Apply various numerical methods for solving linear Ordinary & Partial differential equations arising in engineering field.	
Course Content:			
Module 1	**Solution of Linear Systems of Equation**		**(12 Classes)**
Numerical Computation: Motivation and Objectives, Number Representation, Machine Precision, Round-of Error, Truncation Error, Random Number Generation. Solution of algebraic and transcendental equations: Various types of errors - Bisection method, Regula-Falsi method, Newton-Raphson method, Graffe's method - Bairstow's method - Newton's method for solving f(x,y) = 0 and g(x,y) = 0, secant method, Fixed point iteration method, Solution of linear system of equations, Gauss elimination method, Pivoting, Gauss Jordan method, Iterative methods of Gauss Jacobi and Gauss Seidel, Sufficient conditions for convergence - LU decomposition method, Eigenvalues of a matrix by Power method and Jacobi's method for symmetric matrices.			
Module 2	**Interpolation and Approximation**	**Assignment**	**(8 Classes)**
Interpolation with equal intervals, Newton's forward and backward difference formulae, Interpolation with unequal intervals, Lagrange's interpolation, Newton's divided difference interpolation, Cubic Splines, Difference operators and relations.			
Module 3	**Numerical Differentiation and Integration**		**(10 Classes)**
Numerical differentiation, Approximation of derivatives using interpolation polynomials, Numerical integration using Trapezoidal rule, Simpson's one-third rule, Simpson's three-eighth rule, Weddle's rule, Romberg's Method, Two point and three point Gaussian quadrature formulae, Evaluation of double integrals by Trapezoidal rule and Simpson's one-third rule			
Module 4	**Initial & Boundary Value Problems for Ordinary & Partial Differential Equations**	**Assignment**	**(15 Classes)**
Single step methods — Taylor's series method, Modified Euler's method, Fourth order Runge-Kutta method for solving first order equations, Multi step methods, Milne's and Adams, Bash forth predictor corrector methods for solving first order equations. Finite difference methods for solving second order, two-point linear boundary value problems, Finite difference techniques for the solution of two-dimensional Laplace's and Poisson's			

equations on rectangular domain, One-dimensional heat flow equation by explicit and implicit (Crank Nicholson) methods, One-dimensional wave equation by explicit method.

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: 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. C.F.Gerald and P.O.Wheatley", Applied Numerical Analysis", McGraw-Hill, 1981.
2. Cheneg and Kincaid, "Introduction to Numerical Computing", Tata McGraw-Hill, 1998.

References:

1. SRK Iyengar & RK Jain, Numerical Methods, New Age Internationals.
2. Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition
3. B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_135224
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_141727
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_217628
4. http://.ac.in/courses.php?disciplineID=111
5. http://www.class-central.com/subject/math(MOOCs)
6. http://academicearth.org/
7. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
8. 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: MAT2013	**Course Title: Discrete Mathematics** **Type of Course:1] School Core**	**L-T-P- C**	**4**	**0**	**0**	**4**
Version No.	1.0					
Course Pre-	Linear Algebra					

requisites		
Anti-requisites		**NIL**
Course Description		The course explores the study of mathematical structures that are fundamentally discrete (not continuous), focusing on concepts like set theory, logic, graph theory, combinatorics, and number theory, with applications primarily in computer science fields like algorithms, software development, and cryptography; it covers topics such as propositional logic, proof techniques, relations, functions, counting principles, and basic graph algorithms, providing a foundation for analyzing discrete problems and structures within computer science.
Course Objective		The main objective of the course is that students should learn a particular set of mathematical facts and how to apply them. It teaches students how to think logically and mathematically through five important themes: mathematical reasoning, combinatorial analysis, discrete structures, algorithmic thinking, and applications and modeling. A successful discrete mathematics course should carefully blend and balance all five themes.
Course Outcomes		On successful completion of the course the students shall be able to: CO1 - Explain logical sentences through predicates, quantifiers and logical connectives. CO2 - Deploy the counting techniques to tackle combinatorial problems CO3 - Comprehend the basic principles of set theory and different types of relations. CO4 - Apply different types of structures of trees for developing programming skills
Course Content:		

Module 1	**Fundamentals of Logic**		**(10 Classes)**
Basic Connectives and Truth Tables, Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.			
Module 2	**Principle of Counting**	**Assignment**	**(15 Classes)**
The Well Ordering Principle – Mathematical Induction The Basics of Counting, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Generating Permutations and Combinations			

Advanced Principle Counting: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials.

Module 3	**Relations and Functions**		**(10 Classes)**

Cartesian Products and Relations, Functions, One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions.

Relations, Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders, Lattice, Hasse Diagrams, Equivalence Relations and Partitions.

Module 4	**Recurrence Relations and Generating Functions**		**(10 Classes)**

Homogeneous and inhomogeneous recurrences and their solutions - solving recurrences using generating functions - Repertoire method - Perturbation method - Convolutions - simple manipulations and tricks.

Module 5	**Graph Theory & Algorithms on Networks**	**Assignment**	**(15 Classes)**

Definitions and basic results - Representation of a graph by a matrix and adjacency list - Trees - Cycles - Properties - Paths and connectedness - Sub graphs - Graph Isomorphism - Operations on graphs - Vertex and edge cuts - Vertex and edge connectivity, Euler and Hamilton Paths, Shortest-Paths.

Tree - Definitions, Properties, and Examples, Routed Trees, Binary search tree, Decision tree, spanning tree: BFS, DFS.

Algorithms on Networks - Shortest path algorithm- Dijikstra's algorithm, Minimal spanning tree- Kruskal algorithm and Prim's algorithm.

Targeted Application & Tools that can be used:

Discrete mathematics provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems.

Assignment:

2. **Assignment 1: Logic Equivalences and Predicate calculus.**
3. **Assignment 2: Equivalence Relations and Lattices**
4. **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:

1. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_54588 2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_375 3. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html 4. https://www.scu.edu.au/study-at-scu/units/math1005/2022/ 5.
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: CSE1512	**Course Title: Analysis of Algorithms** **Type of Course: THEORY Only**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	**CSE2001 - Data Structures and Algorithms.**					
Anti-requisites	Nil					
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of applications. This course discusses the classic approaches for algorithm design such as Divide and Conquer, Dynamic Programming, Greedy method. This course also describes other basic strategies searching solution space. The core concepts of analyzing algorithms and classifying them into various complexity classes is covered in the end.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Analysis of Algorithms** and attain **Skill Development** through **Problem Solving** Methodologies.					
Course Out Comes	On successful completion of the course the students shall be able to: 1. Compute efficiency of a given algorithm.[Applying] 2. Apply divide and conquer technique for searching and sorting Problems.[Applying] 3. Apply the Dynamic Programming technique for a given problem. [Applying] 4. Apply greedy technique for solving a Problem.[Applying]					

	5. Demonstrate Back tracking technique and limitations of Algorithms.[Applying]			
Course Content:				
Module 1	**Introduction**	Assignment	Simulation/Data Analysis	**10 Sessions**
Introduction, Asymptotic Notations and its properties, Best case, worst case and average case-Sequential search, Sorting; Mathematical analysis for Recursive and Non-recursive algorithms: Substitution method and Master's Theorem.				
Module 2	**Divide-and-conquer**	Assignment	Simulation/Data Analysis	**08 Sessions**
Introduction. Insertion Sort; Merge sort, Quick sort, Binary search.				
Module 3	**Dynamic programming**	Term paper/Assignment	Simulation/Data Analysis	**10 Sessions**
Introduction with examples, Principles of Memoization, 0-1 Knapsack Problem, Bellman-Ford algorithm, Floyd-Warshall's Algorithms. Chain Matrix Multiplication.				
Module 4	**Greedy technique**	Term paper/Assignment	Simulation/Data Analysis	**09 Sessions**
Introduction, Fractional Knapsack Problem, Minimal Spanning Tree: Prim's Algorithm and Kruskal's Algorithm, Single-source Shortest Path: Dijkstra's Algorithm				
Module 5	**Complexity Classes**	Term paper/Assignment	Simulation/Data Analysis	**08 Sessions**
Complexity Classes- P,NP- NP Hard and NP Complete - Boolean Satisfiability Problem (SAT). Branch and Bound: Knapsack problem; Backtracking, - N-Queens 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.

References

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

Web-Resources

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

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.

Course Code: CSE1514	**Course Title:** CSEXXXX - Object Oriented Programming Using Java **Type of Course: Theory**	**L-T-P- C**	3	0	0	3
Version No.	2.0					
Course Pre-requisites	CSEXXXX – Problem Solving Using C					
Anti-requisites	Nil					
Course Description	This course introduces the core concepts of object-oriented programming. This course has theory and lab component which emphasizes on understanding the implementation and application of object-oriented programming paradigm. It helps the student to build real time secure applications by applying these concepts and also for effective problem solving. The students interpret and understand the need for object oriented programming to build applications.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Problem-Solving using JAVA and attain **SKILL DEVELOPMENT** through **EXPERIENTIAL LEARNING** techniques					
Course Out Comes	**On successful completion of the course the students shall be able to:** **CO1:** Describe the basic programming concepts. [Understand] **CO2:** Apply the concept of classes, objects and methods to solve problems. [Application] **CO3:** Apply the concept of arrays and strings. [Appy] **CO4:** Implement inheritance and polymorphism building secure applications. [Apply]					

	CO5: Apply the concepts of interface and error handling mechanism. [Apply]			
Course Content:				
Module 1	**Basic Concepts of Programming and Java**	Assignment	Problem Solving	**9 Sessions**
Topics: Introduction to Principles of Programming: Process of Problem Solving, Java program structure, Download Eclipse IDE to run Java programs, Sample program, Data types, Identifiers, Variables, Constants in java, Operators, Assignments and Expression, Basic Input/ Output functions, Control Statements: Branching and Looping.				
Module 2	**Classes, objects, methods and Constructors**	Assignment	Problem Solving	**10 Sessions**
Topics: Classes, Objects and Methods: Introduction to object Oriented Principles, defining a class, adding data members and methods to the class, access specifiers, instantiating objects, reference variable, accessing class members and methods. Static Polymorphism: Method overloading, constructors, constructor overloading, this keyword, static keyword, Nested classes, Accessing members in nested classes.				
Module 3	**Arrays, String and String buffer**	Assignment	Problem Solving	**8 Sessions**
Topics: Arrays: Defining an Array, Initializing & Accessing Array, Multi –Dimensional Array, Array of objects. String: Creation & Operation. String builder class, methods in String Buffer.				
Module 4	**Inheritance and Polymorphism**	Assignment	Problem Solving	**10 Sessions**
Topics: Inheritance: Defining a subclass, Types of Inheritance, super keyword. Dynamic Polymorphism: Method overriding. Final keyword: with data members, with member functions and with class. Abstract keyword: with data members, with member functions and with class, Exception handling.				
Module 5	**Input & Output Operation in Java**	Assignment	Problem Solving	**8 Sessions**
Input/output Operation in Java(java.io Package), Streams and the new I/O Capabilities, Understanding Streams, working with File Object, File I/O Basics, Reading and Writing to Files, Buffer and Buffer Management, Read/Write Operations with File Channel, Serializing Objects, Observer and Observable Interfaces.				

Text Book

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

References

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

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

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

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

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

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

Topics relevant to development of "Skill Development":

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

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

Course Code: CSE1515	**Course Title:** CSEXXXX - Object Oriented Programming Using Java Lab **Type of Course: Lab**	**L-T-P- C**	0	0	4	2
Version No.	2.0					
Course Pre-requisites	CSEXXXX – Problem Solving Using C					
Anti-requisites	Nil					
Course Description	This course introduces the core concepts of object-oriented programming. This course has theory and lab component which emphasizes on understanding the implementation and application of object-oriented programming paradigm. It helps the student to build real time secure applications by applying these concepts and also for effective problem solving. The students interpret and understand the need for object oriented programming to build applications.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Problem-Solving using JAVA and attain **SKILL DEVELOPMENT** through **EXPERIENTIAL LEARNING** techniques					
Course Out Comes	**On successful completion of the course the students shall be able to:** **CO1:** Demonstrate basic programming concepts. [Apply] **CO2:** Apply the concept of classes, objects and methods to solve problems. [Application] **CO3:** Apply the concept of arrays and strings. [Appy] **CO4:** Implement inheritance and polymorphism building secure applications. [Apply] **CO5:** Apply the concepts of interface and error handling mechanism. [Apply]					
Course Content:						
Module 1	**Basic Concepts of Programming and Java**	Assignment	Problem Solving	**12 Sessions**		

Download Eclipse IDE to run Java programs, Sample programs on Data types, Identifiers, Variables, Constants in java, Operators, Assignments and Expression, Basic Input/ Output functions, Control Statements: Branching and Looping.

Module 2	**Classes, objects, methods and Constructors**	Assignment	Problem Solving	**14 Sessions**

Problem solving using Classes, Objects and Methods: defining a class, adding data members and methods to the class, access specifiers, instantiating objects, reference variable, accessing class members and methods.
Use Static Polymorphism: Method overloading, constructors, constructor overloading, this keyword, static keyword, Nested classes, Accessing members in nested classes.

Module 3	**Arrays, String and String buffer**	Assignment	Problem Solving	**10 Sessions**

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

Module 4	**Inheritance and Polymorphism**	Assignment	Problem Solving	**12 Sessions**

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

Module 5	**Input & Output Operation in Java**	Assignment	Problem Solving	**12 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.

P1: Programming Exercises on Basic Concepts.

LEVEL 1: Discuss about datatypes and variables.

LEVEL 2: Demonstrate a simple java program

P2: Programming Exercises on Basic Concepts.

LEVEL 1: Discuss about datatypes and variables.

LEVEL 2: Demonstrate a simple java program

P3: Programming Exercises on operators, expressions based on a given scenario.

LEVEL 1: Explain operators, expressions.

LEVEL 2: Demonstrate operators

P4: Programming Exercises Command Line Arguments based on a given scenario.

LEVEL 1: Explain command line arguments

LEVEL 2: Demonstrate command line arguments

P5: Programming Exercises on basic Input/ Output functions and Control Statements: Branching

LEVEL 1: Explain Input/ Output functions

LEVEL 2:Demonstrate Control Statements: Branching

P6: Programming Exercises on Control Statements: Looping

LEVEL 1: Explain variour loops.

LEVEL 2:Demonstrate Control Statements: Looping

P7: Programming Exercises on Creating Objects, classes on a given scenario.

LEVEL 1: Illustrate class, object and methods.

LEVEL 2: Execute java program using class and objects

P8: Programming Exercises on Adding methods and Constructors to the class based on a given scenario.

LEVEL 1: Illustrate methods and constructors

LEVEL 2: Execute java program using methods and constructors

P9: Programming Exercises on methods based on a given scenario.

LEVEL 1: Illustrate method overloading

LEVEL 2: Apply method overloading for the given scenario.

P10: Programming Exercises on methods based on a given scenario.

LEVEL 1: Illustrate constructors overloading

LEVEL 2: Apply constructor overloading for the given scenario

P11: Programming Exercises on methods for static members bassed on a given scenario.

LEVEL 1: Benefits of usage static members

LEVEL 2: Usage of Static Members for the given scenario

P12: Programming Exercises on static methods based on a given scenario.

LEVEL 1: Benefits of usage static methods

LEVEL 2: Usage of Static Methods for the given scenario.

P13: Programming Exercises on nested Classes based on a given scenario.

LEVEL 1: Benefits of usage nested classes

LEVEL 2: Apply the concept of usage of nested classes for the given scenario

P14: Programming Exercises on Arrays and its built-in functions based on a given scenario.

LEVEL 1: Illustrate one dimensional arrays and its functions.

LEVEL 2: Demonstrate programs with single-dimensional arrays and operations.

P15: Programming Exercises on Arrays and its built-in functions based on a given scenario.

LEVEL 1: Illustrate multi dimensional arrays and its functions.

LEVEL 2: Demonstrate programs with multi-dimensional arrays and operations.

P16: Programming Exercises on String Class and its built-in functions based on a given scenario.

LEVEL 1: Explain about String class and String methods.

LEVEL 2: Execute simple java applications for String and StringBuffer operations

P17: Programming Exercises on String Buffer Class and its built-in functions based on a given scenario.

LEVEL 1: Explain about StringBuffer class and String methods.

LEVEL 2: Execute simple java applications for String and StringBuffer operations

P18: Programming Exercises on String Builders and its built-in functions based on a given scenario.

LEVEL 1: Explain about String Builders.

LEVEL 2: Execute java applications for String Builders

P19: Programming Exercises on single, multi level Inheritance and super keyword based on given scenario.

LEVEL 1: Explain single and multi level inheritance.

LEVEL 2: Demonstrate simple applications for the different types of inheritance

P20: Programming Exercises hierarchical Inheritance and super keyword based on given scenario.

LEVEL 1: Explain hierarchical inheritance.

LEVEL 2: Demonstrate simple applications for hierarchical inheritance

P21: Programming Exercises on Overriding.

LEVEL 1: Differentiate method overloading and method overriding.

LEVEL 2: Demonstrate simple program with dynamic method dispatch.

P22: Programming Exercises on Final based on given scenario.

LEVEL 1: Implement programs using concept of final.

LEVEL 2: Use final keyword for the given problem

P23: Programming Exercises on Abstract keyword based on given scenario.

LEVEL 1: Implement programs using concept of Abstract.

LEVEL 2: Use abstract keyword for the given problem

P24: Programming Exercises on Interface based on a given scenario.

LEVEL 1: Differentiate abstract class about interface

LEVEL 2: Implement interfaces in the given problem

P25: Programming Exercises on Exception Handling based on a given scenario.

LEVEL 1: Explain exception handling

LEVEL 2: Solve the given problem using exception handling mechanism.

P26: Programming Exercises on Character Stream Classes based on a given scenario.

LEVEL 1: Explain Character Stream Classes

LEVEL 2: Solve the given problem using Character Stream Class.

P27: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.

LEVEL 1: Explain Read/Write Operations with File Channel

LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.

P28: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.

LEVEL 1: Explain Read/Write Operations with File Channel

LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.

P29: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.

LEVEL 1: Explain Read/Write Operations with File Channel

LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.

P30: Programming Exercises on Read/Write Operations with File Channel based on a given scenario.

LEVEL 1: Explain Read/Write Operations with File Channel

LEVEL 2: Solve the given problem using Read/Write Operations with File Channel.

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

Text Book

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

References

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

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

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

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

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

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

Topics relevant to development of "Skill Development":

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

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

Course Code: CSE1504	**Course Title: Web Technology** **Type of Course: Program core** **Theory Only**	**L-T-P-C**	2-0-0-2
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	**The objective of the course is to familiarize the learners with the concepts of Web Technology and attain Skill Development through Experiential Learning techniques.**		

<table>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of this course the students shall be able to:

CO1: Implement web-based application using client-side scripting languages. (Application level)

CO2: Apply various constructs to enhance the appearance of a website. (Application level)

CO3: Illustrate java-script concepts to demonstration dynamic web site (Application level)

CO4: Apply server-side scripting languages to develop a web page linked to a database. (Application level)</td></tr>
<tr><td>Course Content:</td><td colspan="4"></td></tr>
<tr><td>Module 1</td><td>Introduction to XHTML</td><td>Quizzes and Assignments</td><td>Quizzes on various features of XHTML, simple applications</td><td>8 Sessions</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 2</td><td>Advanced CSS</td><td>Quizzes and assignments</td><td>Comprehension based Quizzes and assignments; Application of CSS in designing webpages</td><td>8 Sessions</td></tr>
<tr><td colspan="5">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</td></tr>
</table>

Module 3	**Fundamentals of JavaScript**	Quizzes and assignments	Application of JavaScript for dynamic web page designing	**7 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**	Quizzes and assignments	Application of PHP in web designing	**7 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): 1] Robert. W. Sebesta, "*Programming the World Wide Web"*, Pearson Education, 8th Edition, 2015. 2] *CSS Notes for Professionals*, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022) 3] Deitel, Deitel, Goldberg,"*Internet & World Wide Web How to Program*", Fifth Edition, Pearson Education, 2021.				
References 1] Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", Pearson Education India, 1st. Edition.2016. 2] Jeffrey C. Jackson, "Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition,2016.				
Topics related to development of "FOUNDATION": 1. Web, WWW, Web browsers, Web servers, Internet. 2. CSS, PHP.				

3. Designing for healthcare.

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

E-References

pu.informatics.global, https://sm-nitk.vlabs.ac.in/

Course Code: CSE1505	**Course Title: Web Technologies Lab** **Type of Course: Program core lab course**	**L-T-P-C**	0	0	2	1

Version No.		1.0
Course Pre-requisites		Database Management Systems-CSE3156
Anti-requisites		**NIL**
Course Description		This course highlights the comprehensive introduction to scripting languages that are used for creating web-based applications. The associated laboratory provides an opportunity to implement the concepts and enhance critical thinking and analytical skills.
Course Objective		**The objective of the course is to familiarize the learners with the concepts of Web Technology and attain Skill Development through Experiential Learning techniques.**
Course Outcomes		**On successful completion of this course the students shall be able to:** **CO1:** Implement web-based application using client-side scripting languages. **(Apply)** **CO2**: Apply various constructs to enhance the appearance of a website. **(Apply)** **CO3:** Apply server-side scripting languages to develop a web page linked to a database. **(Apply)**
Course Content:		

List of Laboratory Tasks:

Experiment No. 1: Demonstration of XHTML features

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

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

Experiment No. 2: Application of CSS in web designing

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

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

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

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

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

Experiment No. 4: Building a website.

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

Targeted Application & Tools that can be used: **Xampp web server to be used to demonstrate PHP.**

Project work/Assignment:

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

Textbook(s):

1. Robert. W. Sebesta, "*Programming the World Wide Web"*, Pearson Education, 9th Edition, 2016.

2]Paul Deitel, Harvey Deitel, Abbey Deital,"Internet & World Wide Web How to Program", Fifth Edition, Pearson Education, 2021.

3]CSS Notes for Professionals, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022)

4]Deitel, Deitel, Goldberg,"*Internet & World Wide Web How to Program*", Fifth Edition, Pearson

Education, 2021.

Reference Book(s):

R1. Randy Connolly, Ricardo Hoar,"Fundamentals of Web Development", Pearson Education India,

1st. Edition.2016.

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

Edition,2016.

Additional web-based resources

W1. W3schools.com

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

W3. docs.microsoft.com

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

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

Topics related to development of "FOUNDATION":

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

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

Course Code: CSE1511	**Course Title: Database Management Systems Laboratory** **Type of Course: 1) Laboratory**	**L-T-P-C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	Foundational understanding of data types, basic programming knowledge, operating systems and file management.					
Anti-requisites	NIL					
Course Description	The Database Management Systems (DBMS) Laboratory is designed to provide students with hands-on experience in database design, implementation, and management using SQL and database management tools such as MySQL. The lab complements theoretical concepts learned in database courses by allowing students to practice database creation, querying, and optimization techniques. The DBMS Lab enables students to develop industry-relevant skills in database management, preparing them for careers in **software development, data engineering, and database administration**.					

Course Objective	The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain **Employability** through **Problem Solving** Methodologies.
Course Out Comes	On successful completion of the course the students shall be able to: 1. Demonstrate the database concepts, practice, and SQL queries. [Apply] 2. Design and implement database schemas while applying normalization techniques to optimize structure. [Apply]] 3. Develop and implement stored procedures, triggers, and views for automation and efficiency. [Apply] 4. To Design and build database applications for real world problems. [Apply]

Course Content:

List of Laboratory Tasks:

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

Labsheet-1 [3 Practical Sessions]

Experiment No 1: [1 Session]

1. To study and implement the different language of Structured Query Language.

Level 1: Perform operations using Data Definition Language and Data Manipulation Language commands including different variants of SELECT on Student DB.

Level 2: Identify the given requirements; valid attributes and data types and Perform DDL and DML operations on a given scenario. [Banking Databases]

Experiment No. 2: [2 Sessions]

2. To study and implement the concept of integrity constraints in SQL.

Level 1: Create tables on Banking database using PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY and demonstrate the working of relational, logical, pattern matching, BETWEEN, IS NULL, IN and NOT IN Special Operators on Student Database.

Level 2: Enforce different types of data and referential integrity constraints. Then try queries with special operators based on the student database. [Banking Database].

Labsheet-2 [3 Practical Sessions]

Experiment No. 3: [1 Session]

3. Implement complex queries in SQL.

Level 1: Implement the conjugate of GROUP BY, ORDER BY and aggregate functions on Banking Database. Level 2: Implement MySQL DB queries on library database using appropriate clauses and aggregate functions. Also order the data either in ascending and descending order using corresponding clause. [Library databases].

Experiment No. 4: [2 Session]

4. To study and implement different types of Set and Join Operations [2 Slots]

Level 1: Demonstrate different types of Set Operations (UNION, UNION ALL, INTERSECT, MINUS) and Join Operations (INNER JOINs, OUTER JOINS, CROSS JOIN, NATURAL JOIN) on two or more tables of Airline Database. Level 2: Use Set and Join operations to retrieve the data from two or more relations(tables) as per the given
scenario. [Airline Database]

Labsheet-3 [2 Practical Sessions]

Experiment No. 5: [2 sessions]

5. To study and implement Views, and Procedures in MySQL DB.

Level 1: Implement MySQL Views, and Procedures in ORACLE DB on Employee database.

Level 2: Analyze the requirement and construct views, and Procedures on Mini Project Domain. [Banking Database]

Labsheet-4 [2 Practical Sessions]

Experiment No. 6: [2 Sessions]

6. To study and implement Functions, and Triggers in MySQL DB.

Level 1: Implement Oracle Functions and Triggers in Oracle on Employee database.

Level 2: Analyze the requirement and construct Functions and Triggers. [Supply chain Database]

Labsheet-5 [2 Practical Sessions]
Experiment No. 7: [2 Sessions]
7. To study and implement Functions, and Triggers in MySQL DB.

Level 1: Implement Oracle Functions and Triggers in Oracle on Employee database.

Level 2: Determine the requirement and construct Functions and Triggers. [Supply chain Database]

Labsheet-6 [4 Practical Sessions]
Experiment No. 8: [2 Sessions]
8. To implement the concept of forms and reports.

Level 1: Implement the concept of forms and reports.

Level 2: Examine the schema relationship.

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

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

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

Course Code: **CSE1513**	**Course Title:** **Analysis of Algorithms Laboratory** **Type of Course: Integrated**	**L-T-P-C**	0	0	2	1
Version No.	1					

Course Pre-requisites	**CSE2001 - Data Structures and Algorithms.**	
Anti-requisites	NIL	
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of applications. This course discusses the classic approaches for algorithm design such as Divide and Conquer, Dynamic Programming, Greedy method. This course also describes other basic strategies searching solution space. The core concepts of analyzing algorithms and classifying them into various complexity classes is covered in the end.	
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Analysis of Algorithms** and attain **Skill Development** through **Experiential Learning** Methodologies.	
Course Out Comes	**On successful completion of the course the students shall be able to:** 1. Compute efficiency of a given algorithm. [Applying] 2. Apply divide and conquer technique for searching and sorting Problems.[Applying] 3. Apply the Dynamic Programming technique for a given problem. [Applying] 4. Apply greedy technique for solving a Problem.[Applying] 5. Demonstrate Back tracking technique and limitations of Algorithms.[Applying]	
Course Content		
Module 1	**Introduction**	**3 Sessions**
Measuring running time of an algorithm, Compare running time of algorithms, Implement sorting algorithms such as bubble sort, selection sort		
Module 2	**Divide-and-conquer**	**3 Sessions**
Compare searching algorithms: Linear Search, Binary Search; Compare Sorting algorithms: Insertion Sort, Merge Sort, QuickSort.		
Module 3	**Dynamic programming**	**3 Sessions**
Introduction and memorization: Factorial; Coin Change Problem ; Floyd-Warshall's Algorithm.		
Module 4	**Greedy technique**	**3 Sessions**
Fractional Knapsack Problem; Minimal Spanning Tree Algorithms-Prim's Algorithm, Kruskal's algorithm		

Module 5	Complexity Classes	3 Sessions
Branch and Bound: Knapsack problem; Backtracking, - N-Queens problem.		

List of Laboratory Tasks:

1. Measuring running time of an algorithm

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

2. Compare running time of algorithms

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

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

Objective: To implement comparison based sorting strategies.

4. Compare searching algorithms

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

5. Compare Sorting algorithms

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

6. Quick Sort

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

7. Dynamic Programming

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

8. Coin Change Problem

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

9. Floyd-Warshall's Algorithm

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

10. Fractional Knapsack Problem

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

	11. Minimal Spanning Tree Algorithm **Objective: To implement greedy strategy to solve the Minimal Spanning Tree problem using Prim's Algorithm.** **12. Kruskal's Minimal Spanning Tree Algorithm** **Objective: To implement greedy strategies to solve the Minimal Spanning Tree problem using Kruskal's Algorithm.** **13. Knapsack Problem** **Objective: To implement Knapsack problem using branch and bound technique.** **14. N-Queen's Problem** **Objective: To demonstrate backtracking method with the help of N-Queen's problem.** **15. Case Study** **Objective: To demonstrate how various techniques can be used to solve the same problem with the help of Knapsack problem.**
	Targeted Application & Tools that can be used 1. PyTorch/Jupyter Notebook – For Python programming
	Text Book **T1** Anany Levitin, *"Introduction to the Design and Analysis of Algorithms*", 3rd edition, Pearson Education, 2018. **T2** Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, *"Introduction to Algorithms*", 4th edition, MIT Press, 2022.
	References R1. J. Kleinberg and E. Tardos, "*Algorithm Design*", Addison-Wesley, 2005. R2. Tim Roughgarden, *"Algorithms Illuminated"* (books 1 through 3), "Operating Systems Design and Implementation", Soundlikeyourself Publishing, 2017-2019. R3. AV Aho, J Hopcroft, JD Ullman, *"The Design and Analysis of Algorithms"*, Addison-Wesley, 1974. R4. Donald E. Knuth, "*The Art of Computer Programming",* Volumes 1and 3 Pearson. Web Based Resources and E-books: W1. NPTEL: https://onlinecourses.nptel.ac.in/noc19_cs47/preview

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

Course Code: CDV2501	**Course Title: Agile Structures and FrameworksType of Course: School Core**	**L- P- C**	**3**	**0**	**3**
Version No.	1.0				
Course Pre-requisites	Software Engineering				
Anti-requisites	NIL				
Course Description	This course imparts knowledge to students in the basic concepts of Agile Software Process, methodology and its development The objective of this course is to provide the fundamentals concepts of Agile and its Significance. This course covers the Agile and its methodologies. The objective of the course is to understand the Agility and Assurance.				
Course Objectives	The objective of the course is to familiarize the learners with the concepts of **Agile Structures and Frameworks** and attain **Skill Development** through **Participative Learning** techniques.				
Course Out Comes	On successful completion of this course the students shall be able to: **1]** Understand the basic concepts of Agile Software Process. **(Knowledge level)** **2]** Comprehend the various Agile Methodologies. **(Comprehension level)** **3]** Develop Agile Software Process. **(Knowledge level)** **4]** Apply principles of Agile Testing. **(Application level)**				
Module 1	**Introduction**	Assignment	Agile Estimation		**08 Sessions**

Introduction to Agile technology, Iterative and Evolutionary Methods, Agile – Agile Development. Agile Values, Agile Principles, Compare and Contrast the agile with traditional methods. Agile Benefits. Agile Estimation Techniques. Case Study				
Module 2	**Agile and Its Significance**	Assignment	Comparison of Agile technologieswith traditional methods	**09 Sessions**
Agile Story : Evolutionary delivery ,Scrum Demo, Planning game, Sprint back log, adaptive planning. Agile Motivation – Problems With The Waterfall - Research Evidence. Scrum : Method Overview ,Life cycle phases and Work productroles and practices.				
Module 3	**Agile methodology**		Case Study	**12 Sessions**
Extreme Programming: Method Overview ,Life cycle phases and Work product roles and practices. Unified process : Method Overview ,Life cycle phases and Work product roles and practices. EVO : Method Overview ,Life cycle phases and Work product roles and practices. Case Study.				
Module 4	**Agility and Quality Assurance**	Assignment	Apply the testing concepts usingPrograming	**09 Sessions**
Agile product development – Agile Metrics – Feature Driven Development (FDD). Agile approach to Quality Assurance. Test Driven Development – Agile approach in Global Software Development. Agile Technology Tools.				
Targeted Application & Tools that can be used: JIRA				
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course				
Agile Estimation Comparison of Agile technologies with traditional methods 3. Case Study: Student group must collaborate and report together along with assigned batch members. Collect the requirements from the client and adopt the suitable agile practice method for your project Installation and features of JIRA tool.				

Text Book 1]Craig Larman, “Agile and Iterative Development – A Manager’s Guide”, Pearson Education – 2006 2] Edward Scatter “Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban, 2015
References 1] Chetankumar Patel, Muthu Ramachandran, Story Card Maturity Model (SMM): A Process Improvementework for Agile Requirements Engineering Practices, Journal of Software, Academy Publishers, Vol 4, No 5 (2009), 435, Jul 2009. 2] Hazza& Dubinsky, Agile Software Engineering, Series: Undergraduate Topics in Computer Science, nger 2009 3] Kevin C. Desouza, Agile information systems: conceptualization, construction, and management, erworth-Heinemann, 2007. **Web resources:** https://presiuniv.knimbus.com/user#/home
Topics relevant to "SKILL DEVELOPMENT”: Agile Estimation techniques for **skill development** through **Participative Learning techniques.** This is attained through the assessment component mentioned in the course handout.

Course Code: CDV2502	**Course Title: DevOps Tools And InternalsType of Course: Theory**	**L-T-P-C**	**3-0-0-3**
Version No.	1.2		
Course Pre-requisites	**Fundamentals of Devops**		
Anti-requisites	**NIL**		
Course Description	This course is designed to offer profound perceptions and knowledge in various tools like Git, Ansible, Selenium and Jekins. With the proficient learning of DevOps course, a student will be able to work in all the above tools and become a trained practitioner in the integration and monitoring of software. **DevOps Tool** is an application that helps the software development process to industrialize. It mainly focuses on communication and collaboration between product management, software development, and operations professionals. The objective of this course is to discuss and implementthe various tools usage and internals practically.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of **DevOps ToolsAnd Internals** and attain **Skill Development** through **Experiential Learning** techniques.		
Course OutComes	On successful completion of this course the students shall be able to: **1]Apply** the features and common Git workflow. [Application] **2] Practice** the filters and plugins to populate, manipulate, and manage data used by Ansible Playbooks. [Application] **3] Compute** the features of selenium IDE. [Application]4] **Interpret** the installation and features of Jenkins and build jobs. [Application]		
Course Content:			

Module 1	**Git**	Quiz	Quiz on Git commands	**5L +4P Classes**
Topics: Introduction to Git, Features of Git, Benefits, Workflow, Git vs GitHub, Installation of Git on Windows/Linux and Environment set up, All Git Commands-Working with local and remote repositories, Running first Git command, Fundamentals of Repository structure and file status life cycle, Working locally with staging, unstaging and commit.				
Module 2	Containerization Usin gDocker	Quiz	Quiz on Ansible tool usage	**5L +4P Classes**
Topics: Docker Life Cycle,Docker Installation, Docker Operations,Docker Concepts - Registry, Repository, Tag, Image and Containers, Create A Docker Hub Account, Docker Images and Containers, Pushing Docker To Container Hub, Docker File.				
Module 3	**Ansible**	Assignm ent	Assignments on Selenium tool usage and test case	**5L +4P Classes**
Topics: Ansible Workflow, Architecture, Installation in Linux/Windows, ad-hoc Commands, Playbooks, Tower, Roles ,Variables open link, Tags, Galaxy, Commands Cheat Sheets, Modules, Shell, Templates, YAML, Inventory, Debug, Apt, Lineinfile, Copy, Command, File, Vault, Windows, Yum, AWX, Unarchive, Ansible Pip				
Module 4	**Jenkins**	Assignment	Assignments on Jenkins tool usage and Build jobs	**5L +4P Classes**
Topics: Introduction To Continuous Integration, Jenkins Architecture, Managing Nodes On Jenkins, Jenkins Master Node Connection, Jenkins Integration With Devops Tools, Understanding CI/CD Pipelines, Creating A CI/CD Pipeline				

Text Book

1. Craig Berg, "*DevOps For Beginners: A Complete Guide to DevOps Best Practices (Including How You Can Create World-Class Agility, Reliability, And Security In Technology Organizations With DevOps) (Code tutorials)*", Paperback – June 12, 2020.

Ferdinando Santacroce, "*Git Essentials*", Packt Publishing, April 2015, ISBN: 9781785287909

John Ferguson Smart. "*Jenkins: The Definitive Guide*", O'Reilly Media, Inc., July 2011, ISBN: 9781449305352

References

1. Jeff Geerling, "*Ansible for DevOps: Server and configuration management for humans*", Leanpub, August 5, 2020

. Unmesh Gundecha, Carl Cocchiaro, "*Learn Selenium*", Packt Publishing, July 2019, ISBN: 9781838983048

3. Gaurav Agarwal, "*Modern DevOps Practices: Implement and secure DevOps in the public cloud with cutting-edge tools, tips, tricks, and techniques*", July 2021.

4. Mikael Krief, "*Learning DevOps: The complete guide to accelerate collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps*", October 2019

Weblinks:

. https://git-scm.com/book/en/v2

. https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner

. **https://www.javatpoint.com/selenium-tutorial**

. **https://www.javatpoint.com/ansible**

. https://www.tutorialspoint.com/jenkins/jenkins_managing_plugins.htm

. https://nptel.ac.in/courses/128106012

Topics relevant to "SKILL DEVELOPMENT": Git&Junit, Ansible, Selenium, Jenkins for Skill Development throughExperiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CDV2503	**Course Title: DevOps Tools And InternalsLab Type of Course: Laboratory**	**L-T-P-C**	**0-0-2-1**
Version No.	1.2		
Course Pre-requisites	**Fundamentals of Devops**		
Anti-requisites	**NIL**		
Course Description	This course is designed to offer profound perceptions and knowledge in various tools like Git, Ansible, Selenium and Jekins. With the proficient learning of DevOps course, a student will be able to work in all the above tools and become a trained practitioner in the integration and monitoring of software. **DevOps Tool** is an application that helps the software development process to industrialize. It mainly focuses on communication and collaboration between product management, software development, and operations professionals. The objective of this course is to discuss and implementthe various tools usage and internals practically.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of **DevOps ToolsAnd Internals** and attain **Skill Development** through **Experiential Learning** techniques.		
Course OutComes	On successful completion of this course the students shall be able to: 4]**Apply** the features and common Git workflow. [Application] **5] Practice** the filters and plugins to populate, manipulate, and manage data used by Ansible Playbooks. [Application] **6] Compute** the features of selenium IDE. [Application]4] **Interpret** the installation and features of Jenkins and build jobs. [Application]		
Course Content:			

Module 1	**Git**	Quiz	Quiz on Git commands	**5L +4P Classes**

Topics:
Introduction to Git, Features of Git, Benefits, Workflow, Git vs GitHub, Installation of Git on Windows/Linux and Environment set up, All Git Commands-Working with local and remote repositories, Running first Git command,
Fundamentals of Repository structure and file status

life cycle, Working locally with staging, unstaging and commit.				
Module 2	Containerization Usin gDocker	Quiz	Quiz on Ansible tool usage	**5L +4P Classes**
Topics: Docker Life Cycle,Docker Installation, Docker Operations,Docker Concepts - Registry, Repository, Tag, Image and Containers, Create A Docker Hub Account, Docker Images and Containers, Pushing Docker To Container Hub, Docker File.				
Module 3	**Ansible**	Assignm ent	Assignments on Selenium tool usage and test case	**5L +4P Classes**
Topics: Ansible Workflow, Architecture, Installation in Linux/Windows, ad-hoc Commands, Playbooks, Tower, Roles ,Variables open link, Tags, Galaxy, Commands Cheat Sheets, Modules, Shell, Templates, YAML, Inventory, Debug, Apt, Lineinfile, Copy, Command, File, Vault, Windows, Yum, AWX, Unarchive, Ansible Pip				
Module 4	**Jenkins**	Assignment	Assignments on Jenkins tool usage and Build jobs	**5L +4P Classes**

Topics:
Introduction To Continuous Integration, Jenkins Architecture, Managing Nodes On Jenkins, Jenkins Master Node Connection, Jenkins Integration With Devops Tools, Understanding CI/CD Pipelines, Creating A CI/CD Pipeline

List of Laboratory Tasks:

Git

1. Level 1: Installation of Git on windows Level 2: Git commands-Local repositories Level 2: Git commands-Remote repositories

2. How Git can handle automatically file modifications when they are not related to the same lines of text. Level 1: You are in a new repository located in C:\Repos\Exercises\Ch2-1.
   Level 1: You have a master branch with two previous commits: the first commit with a file1.txt file and thesecond commit with a file2.txt file.
   Level 2: After the second commit, you created a new branch called File2Split. You realized that file2.txt is too big,and you want to split its content by creating a new file2a.txt file. Do it, and then commit the modifications.

3. How to resolve conflicts when Git cannot merge files automatically.
   Level 1: You are in the same repository used earlier, C:\Repos\Exercises\Ch2-1. On the master branch, you add the file3.txt file and commit it.
   Level 2: Then, you realize that it is better to create a new branch to work on file3.txt, so you create the File3Work branch. You move in this branch, and you start to work on it, committing modifications.
   Level 2: The day after, you accidentally move to the master branch and make some modifications on the file3.txtfile, committing it. 5. Then, you try to merge it.

4. Level 1: Installation of Ansible
   Level 2: Create a basic inventory file
   Level 2: Running your first Ad-Hoc Ansible command.

Ansible

5. Ansible Archive
   Level 1: Compressing the Directory with TAR and tar and gz
   Level 1: Compress the file – Default File Compress format and Remove the Source files after archivingLevel 2: Create a ZIP file archive – File and Directory
   Level 2: Create a BZIP archive – File and Directory

6. A Quick Syntax of Ansible Shell module – ADHOC
   Level 1: A Quick Syntax of Ansible Shell module in a PlaybookLevel 1: Ansible Shell Examples
   Level 2: Execute a Single Command with Ansible Shell Level 2: Execute a Command with Pipe and Redirection

. Level 1: Run playbook
   Level 2: Create the file on the target machines or servers as mentioned in the inventory file and the webserver'sgroup, save the below code with .yml extension and run the playbook.
   Level 2: Create multiple directories. To create multiple directories with one single task you can use the loop **with_items** statement. So when you run the below playbook it is interpreted as 3 different tasks.

Selenium

. Level 1: Selenium IDE Download and Install
   Level 2: Selenium IDE - First Test Case, Login Test and command usage

. Level 1: Write a script to open google.co.in using chrome browser (ChromeDriver).
   Level 2: Write a script to open google.com and verify that title is Google and also verify that it is redirected togoogle.co.in.

10. Level 1: Write a script to open google.co.in using internet explorer (InternetExplorerDriver).Level 2: Write a script to create browser instance based on browser name.

11. Level 1: Write a script to close all the browsers without using quit() method.Level 2: Write a script to search for specified option in the listbox

Jenkins

12. Level 1:
Environment Setup
Level 2:
Jenkins downloading and installation

13. Level 1:
. Setup a Jenkins Job with Apache Ant Build Tool
2. Setup a Jenkins Job with Apache MavenLevel 2 :
1. Setup a Jenkins Job with Batch Script.

14. Level 1: Add a Linux Node (Also Check SSH Slaves plugin plugins)Level 1: Add a Windows Node
Level 2: Assign a Java Based Job to Linux and Build it Level 2: Assign a MSBuild Based to Windows and Build it

Targeted Application & Tools that can be used:

Tracking changes in the source code and source code managementAutomates web browsers
Configuration Management and IT automation. Integration of Individual Jobs and Effortless Auditing
Tools: Git, Ansible, Selenium and Jekins

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

Each batch of students (self-selected batch mates) will identify projects from searching on Google and implementwith the most suitable 2 or 3 antecedents.

Text Book

4. Craig Berg, "*DevOps For Beginners: A Complete Guide to DevOps Best Practices (Including How You Can Create World-Class Agility, Reliability, And Security In Technology Organizations With DevOps) (Code tutorials)*", Paperback – June 12, 2020.
Ferdinando Santacroce, "*Git Essentials*", Packt Publishing, April 2015, ISBN: 9781785287909
John Ferguson Smart. "*Jenkins: The Definitive Guide*", O'Reilly Media, Inc., July 2011, ISBN: 9781449305352

References

5. Jeff Geerling, "*Ansible for DevOps: Server and configuration management for humans*", Leanpub, August 5, 2020
. Unmesh Gundecha, Carl Cocchiaro, "*Learn Selenium*", Packt Publishing, July 2019, ISBN: 9781838983048
7. Gaurav Agarwal, "*Modern DevOps Practices: Implement and secure DevOps in the public cloud with cutting-edge tools, tips, tricks, and techniques*", July 2021.
8. Mikael Krief, "*Learning DevOps: The complete guide to accelerate collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps*", October 2019

Weblinks:

. https://git-scm.com/book/en/v2
. https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner
. https://www.javatpoint.com/selenium-tutorial
0. https://www.javatpoint.com/ansible
1. https://www.tutorialspoint.com/jenkins/jenkins_managing_plugins.htm
2. https://nptel.ac.in/courses/128106012

Topics relevant to "SKILL DEVELOPMENT": Git&Junit, Ansible, Selenium, Jenkins for Skill Development throughExperiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CDV2504	**Software Testing**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	CDV1700 DevOps Foundations					
Anti-requisites	NIL					
Course Description	This course introduces students to fundamental principles and practices of software testing. It emphasizes various testing techniques including black-box and white-box testing, test case design, test automation, debugging, and reliability assessment. The course provides insights into the role of testing in software quality assurance and equips students to apply systematic testing strategies in real-world software projects.					
Course Objectives	To understand the importance of software testing in the software development lifecycle. To learn various testing techniques and their applications in software quality assurance. To analyze testing outcomes and debugging strategies. To explore tools and methodologies for automated testing and test management.					
Course Out Comes	CO1: Understand various software testing levels, techniques, and models used in industry. (Understand Level) CO2: Analyze software testing strategies and defect tracking mechanisms. (Analyze Level) CO3: Apply white-box and black-box testing techniques to evaluate software functionality and performance. (Apply Level) CO4: Implement and manage automated test scripts using industry tools. (Apply Level)					
Module 1	**Fundamentals of Software Testing**		**Understand**		**No. of Sessions: 11**	
Software Development Life Cycle, Role of Testing in SDLC, Verification vs Validation, Static vs Dynamic Testing, Levels of Testing, Types of Testing – Unit, Integration, System, Acceptance, Regression.						
Module 2	**Test Case Design Techniques and Defect Management**		**Analyse**		**No. of Sessions: 11**	
Black Box Testing: Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables; White Box Testing: Statement, Branch, Path, and Condition Coverage;						

Defect Lifecycle, Bug Tracking Systems, Test Reporting.				
Module 3	**Test Management and Automation**		**Apply**	**No. of Sessions: 11**
Test Planning, Test Metrics, Traceability Matrix, Test Environment Setup; Automation Testing Concepts, Selenium, JUnit/NUnit, Script Writing, Continuous Testing in CI/CD.				
Module 4	**Advanced Testing and Tools**		**Apply**	**No. of Sessions: 12**
Performance Testing (JMeter), Load Testing, Security Testing (OWASP overview), Mutation Testing, Test Data Generation, Model-Based Testing, Introduction to AI-driven Testing Tools.				

Textbooks

T1: Paul Ammann, Jeff Offutt, Introduction to Software Testing, Cambridge University Press, 2016.

T2: Srinivasan Desikan, Gopalaswamy Ramesh, Software Testing: Principles and Practices, Pearson Education, 2006.

Reference Books

R1: Ron Patton, *Software Testing*, Pearson Education, 2nd Edition, 2005.

R2: Aditya P. Mathur, *Foundations of Software Testing*, Pearson Education, 2008.

R3: Ilene Burnstein, *Practical Software Testing*, Springer, 2003.

R4: Boris Beizer, *Software Testing Techniques*, Dreamtech Press, 2nd Edition, 2003.

Web Resources

Course Code: CDV2505	**Software Testing Lab**	**L-T-P-C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	CBD 1700 Introduction to Big Data					
Anti-requisites	NIL					
Course Description	This course provides practical exposure to various software testing techniques and tools used in industry. It enables students to design, develop, and execute test cases for functional and non-functional software systems. The lab sessions emphasize both manual and automated testing environments using open-source tools and frameworks.					
Course Objectives	To introduce the fundamental concepts of software testing life cycle and test management.					

	To provide hands-on training in writing effective test cases and executing them on various software systems. To expose students to automated testing tools like Selenium, JUnit, and bug-tracking tools. To develop skills in identifying, documenting, and reporting software defects and ensuring software quality.
Course Out Comes	Upon successful completion of this course, the students will be able to: CO1: Understand various software testing techniques, strategies, and levels of testing. CO2: Apply manual and automated testing tools to evaluate the functionality and performance of software applications. CO3: Analyze test outcomes and identify bugs using defect tracking systems. CO4: Design and implement comprehensive test plans and interpret results to improve software quality.

List of Tools:

1. Manual Testing Tools

- **TestLink** – Test management and documentation
- **Bugzilla** – Bug reporting and tracking
- **MantisBT** – Lightweight issue tracker

2. Unit Testing Tools

- **JUnit** – Unit testing for Java
- **TestNG** – Advanced unit testing for Java
- **PyTest** – Unit testing for Python
- **NUnit** – Unit testing for .NET applications

3. Automation Testing Tools

- **Selenium IDE** – Record-and-playback for browser testing
- **Selenium WebDriver** – Code-based automation for web apps
- **Katalon Recorder** – Selenium-based test recorder (Chrome plugin)

4. Mobile Application Testing

- **Appium** – Automation for Android and iOS apps

5. Integration / CI Tools

- **Jenkins** – Continuous integration, pipeline integration
- **GitHub Actions** – Cloud-based CI/CD automation

6. Performance and Load Testing Tools

- **Apache JMeter** – Load, stress, and performance testing for web applications

- **Gatling** – High-performance load testing

✅ 7. Code Coverage Tools

- **JaCoCo** – Java code coverage
- **Coverage.py** – Python code coverage
- **Clover** – Code coverage for Java/Groovy

🧹 8. Code Quality & Static Analysis

- **SonarQube** – Code quality and security analysis
- **PMD** – Java source code analyzer
- **FindBugs / SpotBugs** – Bug detection in Java

🌐 9. Cross-Browser Testing

- **Selenium Grid** – Parallel testing on different browsers/machines
- **Browsershots** – Web design testing in various browsers

List of Experiments

1. **Introduction to software testing, SDLC vs STLC, writing simple test cases**
2. **Manual testing – Functional and Non-functional test case design for a sample application**
3. **Boundary Value Analysis (BVA) and Equivalence Class Partitioning**
4. **Decision Table and State Transition testing techniques**
5. **Path coverage and control flow graph-based white-box testing**
6. **JUnit testing for Java programs – Unit Testing**
7. **Writing automation test scripts using Selenium IDE**
8. **Developing automated test cases using Selenium WebDriver**
9. **Cross-browser testing using Selenium Grid**
10. **Integration of Selenium with JUnit/TestNG for automation testing**
11. **Bug reporting and tracking using Bugzilla or Mantis**
12. **Test case execution and defect tracking on sample project**
13. **Performance testing using Apache JMeter**
14. **Preparing test plans, test suites, and test summary reports**
15. **Mini-project: Complete test life cycle on a web/mobile application including documentation**

Textbooks

T1: William Stallings, Cryptography and Network Security: Principles and Practice, Pearson, 7th Edition, 2017.
T2: Behrouz A. Forouzan, Cryptography and Network Security, McGraw-Hill Education, 2nd Edition, 2011.

Reference Books

R1: Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 2nd Edition, 1996.
R2: Douglas R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 4th Edition, 2018.
R3: Charlie Kaufman, Radia Perlman, and Mike Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall, 2nd Edition, 2002.
R4: Christof Paar, Jan Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer, 2010.

Web Resources

W1: https://cryptography.io/ – Python cryptographic library and resources.
W2: https://nvlpubs.nist.gov/ – NIST publications on cryptographic standards.
W3: https://www.owasp.org/ – Open Web Application Security Project for best practices.
W4: https://www.tutorialspoint.com/cryptography/index.htm – Cryptography basics and tutorials.
W5: https://www.coursera.org/learn/crypto – Stanford's free online cryptography course on Coursera.

Course Code: CSE1700	**Course Title:** Essentials of AI **Type of Course:** Theory	**L- T- P- C**	3	**0**	0	**3**
Version No.	2.0					
Course Pre-requisi Data**tes**	Basic knowledge of programming, mathematics, understanding of data handling					
Anti-requisites	**NIL**					
Course Description	This course is a comprehensive introductory course designed to equip learners with the fundamental Python programming skills necessary to work with artificial intelligence (AI) technologies. This course is aimed at individuals who are new to AI but have a basic understanding of programming concepts. It combines Python programming fundamentals with hands-on experience in implementing AI techniques such as machine learning, neural networks, and natural language processing.					
Course Objective	The objective of the course is to Understand Python Programming Fundamentals, Manipulate and Process Data with Python, Implement Machine Learning Algorithms and Build and Train Neural Networks for AI Applications.					
Course Outcomes	On successful completion of the course the students shall be able to: CO 1: Apply Python Programming to AI Projects CO 2: Build and Train Machine Learning Models CO 3: Develop Deep Learning Models with Neural Networks CO 4: Deploy AI Solutions and Understand Ethical Implications					

Course Content:				
Module 1	Introduction to Python Programming for AI	Assignment	Implementation	**10 Sessions**
Topics: Python Basics: Variables, Data Types, Operators, and Control Flow Functions, Loops, and Conditionals statements, Data Structures: Lists, Tuples, Dictionaries, Sets ,Introduction to Libraries: NumPy and Pandas for data manipulation, Basic Input/Output and File Handling Introduction to Python for AI: Libraries and Frameworks Overview				
Module 2	Data Processing, Visualization	Assignment	Implementation	10 Sessions
Topics: cleaning and preprocessing with Pandas,Handling missing data, outliers, and duplicates, Data transformation (Normalization, Encoding), Introduction to Matplotlib and Seaborn for Data Visualization, Exploratory Data Analysis (EDA), Visualizing datasets to understand patterns and relationships.				
Module 3	Introduction to Machine Learning	Mini - Project	Implementation	10 Sessions
Topics: What is Machine Learning? Types of ML algorithms Supervised Learning: Regression, Classification, Unsupervised Learning: Clustering, Key ML Algorithms: Linear Regression, Decision Trees, K-Means ,Introduction to Scikit-learn library Model evaluation (Accuracy, Precision, Recall, Confusion Matrix)				
Module 4	Neural Networks and Deep Learning	Quiz	Implementation	10 Sessions
Topics: Introduction to Neural Networks and Deep Learning, Perceptron Model and Backpropagation Deep Neural Networks and Activation Functions, Introduction to TensorFlow and Keras, Building and Training Neural Networks for Image and Text Classification, Overview of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)				

Targeted Application & Tools that can be used:
Applications:

1. **Data Preprocessing**: Clean and manipulate data from various sources such as CSV, Excel, SQL databases, and APIs.
2. **Exploratory Data Analysis (EDA)**: Gain insights into datasets by identifying trends, patterns, and outliers.
3. **Predictive Modeling**: Build models for classification (e.g., spam detection) and regression (e.g., house price prediction).
4. **Clustering**: Group data into clusters for unsupervised learning tasks (e.g., customer segmentation).
5. **Model Evaluation**: Assess model performance using appropriate metrics such as accuracy, precision, recall, and F1-score.

Tools:

- **Pandas**: For data manipulation and cleaning (e.g., handling missing values, merging datasets).
- **NumPy**: For numerical operations and working with arrays and matrices.
- **Matplotlib**: For creating static, animated, and interactive visualizations.
- **Seaborn**: For advanced data visualizations (e.g., heatmaps, pair plots).
- **Plotly**: For creating interactive visualizations, especially useful for large datasets.
- **Scikit-learn**: The go-to library for implementing machine learning algorithms (e.g., linear regression, decision trees, k-means clustering).
- **XGBoost**: For advanced gradient boosting models, particularly for large-scale machine learning tasks.
- **TensorFlow** (for deep learning in Module 4): A powerful open-source library for building machine learning and deep learning models.
- **Keras**: High-level neural network API, built on top of TensorFlow, to easily create deep learning models.

  NLTK: The Natural Language Toolkit for various text processing tasks like tokenization, stemming, and part-of-speech tagging.

  spaCy: A fast NLP library for advanced NLP tasks such as named entity recognition and dependency parsing.

  Transformers (by Hugging Face): A powerful library for using pre-trained Transformer-based models like BERT, GPT, and others for advanced NLP tasks.

Text Book(s):

T1: **Essentials of Python for Artificial Intelligence and Machine Learning** by Pramod Gupta and Anupam Bagchi

Reference(s):

- "Artificial Intelligence with Python" – Prateek Joshi
- "Python Machine Learning" – Sebastian Raschka & Vahid Mirjalili
- "Hands-On Artificial Intelligence with Python" – Teet Straus
- "Deep Learning for Coders with Fastai and PyTorch" – Jeremy Howard & Sylvain Gugger

Course Code: **CSE1701**	**Course Title:** Essentials of AI LAB **Type of Course:** Lab	**L- T- P- C**	0	0	4	2
Version No.	2.0					
Course Prerequisites	Basic Java Programming Knowledge, Mathematics: Linear Algebra and Probability, Basic Data Structures and Algorithms, Familiarity with Libraries and Tools, Understanding of Basic Machine Learning Concepts.					
Anti-requisites	**NIL**					
Course Description	This course introduces students to the essential concepts and techniques of Artificial Intelligence (AI) with a focus on practical implementation using Python. Students will explore core AI topics such as search algorithms, knowledge representation, machine learning, and neural networks, while gaining proficiency in using popular Python libraries like NumPy, pandas, scikit-learn, and TensorFlow. Through a series of lab exercises and projects, students will apply AI principles to solve real-world problems, develop intelligent applications, and understand how AI systems function at a foundational level.					

Course Objective	The primary objectives of the course are to Gain Proficiency in AI Concepts and Python Implementation, Develop and Implement Machine Learning Models, Understand and Build Neural Networks, Apply AI to Real-World Problems
Course Outcomes	On successful completion of the course the students shall be able to: 1. Proficiency in Implementing AI Algorithms Using Python 2. Ability to Build and Evaluate Machine Learning Models 3. Hands-on Experience with Neural Networks and Deep Learning 4. Practical Application of AI to Solve Real-World Problems
Course Content:	

Module 1	Introduction to AI and Python for AI	Assignment	Implementation	**8 Sessions** I

Lab Assignment 1: Setting Up the Python Environment

- **Objective:** Get familiar with setting up a Python environment for AI projects.
- **Tasks:**

1. Install Python, Anaconda, and Jupyter Notebook.
2. Set up a virtual environment for AI development.
3. Install essential Python libraries: numpy, pandas, matplotlib, and scikit-learn.
4. Write and execute simple Python code to verify installation (e.g., print a “Hello AI” message).

Lab Assignment 2: Basic Python Programming for AI

- **Objective:** Understand and practice the basic Python syntax and data structures used in AI.
- **Tasks:**

1. Write Python code to work with basic data types (integer, float, string, boolean).
2. Implement and manipulate Python lists, tuples, sets, and dictionaries.
3. Create basic control flow structures: if-else, for loops, while loops.
4. Use functions and lambda functions to solve small AI-related problems, such as calculating factorial or Fibonacci numbers.

Lab Assignment 3: Data Exploration and Preprocessing

- **Objective:** Learn how to work with data for AI models.
- **Tasks:**

1. Load a dataset (e.g., Titanic or Iris dataset) using pandas.
2. Clean the dataset by handling missing values, removing duplicates, and converting data types if needed.
3. Explore the dataset by visualizing it using matplotlib and seaborn.
4. Perform basic data preprocessing tasks such as feature scaling, encoding categorical variables, and splitting data into training and testing sets.

Module 2	Data Processing, Visualization	Assignment	Implementation	8 Sessions

Lab Assignment 1: Data Preprocessing with Pandas

Objective:
Learn the fundamentals of data preprocessing, including cleaning, handling missing values, and performing basic transformations using **Pandas**.

Tasks:

1. **Load and Inspect the Dataset:**
   - Load a dataset (e.g., **Iris**, **Titanic**, **Wine Quality** dataset) using pandas.read_csv() or pandas.read_excel().
   - Inspect the first few rows of the dataset using .head() and check basic information using .info().
2. **Handle Missing Values:**
   - Identify missing values in the dataset using .isnull() or .isna().
   - Handle missing data by imputing with mean, median, or mode using SimpleImputer from sklearn, or remove rows with missing data using .dropna().
3. **Data Transformation:**
   - Convert categorical variables to numerical values using one-hot encoding or label encoding.
   - Normalize/standardize numerical columns using StandardScaler or MinMaxScaler from sklearn.
4. **Subset and Filter Data:**
   - Create subsets based on certain conditions (e.g., select rows where a specific feature value is greater than a threshold).
   - Filter outliers from numerical data using interquartile range (IQR).

Lab Assignment 2: Data Aggregation and Grouping with Pandas

Objective:
Master aggregation and grouping techniques using **Pandas** for summarizing data.
Tasks:

1. **Group Data by Category:**
   - Group data by one or more categorical features (e.g., "class" in the Iris dataset or "embarked" in Titanic dataset).
   - Use .groupby() to calculate aggregate statistics such as mean, median, sum, and count.
2. **Pivot Tables:**
   - Create a pivot table to summarize data (e.g., aggregate the average age of passengers in the Titanic dataset by class and gender).
   - Use .pivot_table() to perform multi-dimensional aggregation.
3. **Data Aggregation and Custom Functions:**
   - Apply custom aggregation functions to the grouped data (e.g., calculate custom metrics or perform complex transformations within each group).
4. **Sorting and Ranking Data:**
   - Sort the dataset by multiple columns (e.g., sorting by "age" or "fare").
   - Rank data based on specific metrics (e.g., assign ranks to passengers by fare in the Titanic dataset).

Lab Assignment 3: Data Visualization with Matplotlib and Seaborn

Objective:
Learn to visualize datasets using **Matplotlib** and **Seaborn** for better understanding and insights.
Tasks:

1. **Basic Plotting with Matplotlib:**
   - Create simple plots like line plots, bar plots, and histograms using **Matplotlib**.
   - Customize the plots by setting titles, labels, and legends.
   - Create scatter plots to visualize relationships between two variables.

2. **Advanced Plotting with Seaborn:**
○ Use **Seaborn** to create advanced visualizations like pair plots, heatmaps, box plots, and violin plots.
○ Customize visualizations with color palettes, styling, and themes.
○ Create a correlation heatmap to visualize correlations between features in the dataset.
3. **Distribution Visualizations:**
○ Plot distributions of continuous variables using **Seaborn's** distplot() or kdeplot().
○ Create bar plots for categorical variables to understand their frequency distribution.
4. **Multi-Plot Grid Layouts:**
○ Use **Matplotlib's** subplots() function to create multiple plots in a grid layout for comparison (e.g., scatter plot and histogram in the same figure).

Lab Assignment 4: Visualizing Relationships and Feature Importance

Objective:
Understand how to visualize relationships between features and evaluate feature importance for predictive models.
Tasks:

1. **Scatter Plot Matrix:**
○ Use **Seaborn's** pairplot() to create a scatter plot matrix to visualize the relationships between multiple features.
○ Analyze the pairwise relationships between features and identify any patterns or correlations.
2. **Heatmap of Correlation Matrix:**
○ Use **Pandas** to calculate the correlation matrix of numeric features.
○ Visualize the correlation matrix using **Seaborn's** heatmap() to understand feature correlations and multicollinearity.
3. **Feature Importance from Models:**
○ Train a decision tree or random forest model using **scikit-learn** on a dataset (e.g., **Iris** or **Titanic**).
○ Visualize feature importance using a bar chart to understand which features have the most impact on the model.
4. **Visualizing Predictions vs. Actual Values:**
○ For regression tasks, visualize the predicted values against the actual values using a scatter plot.
○ For classification tasks, visualize the classification results with a confusion matrix.

Lab Assignment 5: Time Series Data Visualization and Processing

Objective:
Learn how to process and visualize time series data, which is common in AI applications like forecasting and trend analysis.
Tasks:

1. **Load and Preprocess Time Series Data:**
○ Load a time series dataset (e.g., stock market data, weather data).
○ Parse dates properly and set the date column as the index using pd.to_datetime() and .set_index().
2. **Plot Time Series Data:**
○ Plot a time series line chart using **Matplotlib** to visualize trends over time.
○ Create rolling averages (e.g., 7-day, 30-day) to smooth out short-term fluctuations in the time series data.

3. **Seasonal Decomposition of Time Series:**
○ Use **statsmodels** to decompose a time series into seasonal, trend, and residual components.
○ Visualize the decomposed components to understand seasonal variations.
4. **Forecasting with Simple Models:**
○ Use simple forecasting models (e.g., moving average, ARIMA) to predict future values.
○ Visualize the forecasted data along with actual historical data.

Module 3	Introduction to Machine Learning	Assignments	Implementation	8 Sessions

Lab Assignment 3: Implementing Linear Regression

- **Tasks:**

1. Load a real-world dataset (e.g., **Boston Housing Price** dataset).
2. Train a **Linear Regression** model using LinearRegression() from scikit-learn.
3. Evaluate the model using **Mean Squared Error (MSE)** and **R-squared Score**.
4. Visualize the regression line using Matplotlib.

Lab Assignment 4: Logistic Regression for Classification

- **Tasks:**

1. Load the **Iris** or **Breast Cancer** dataset.
2. Preprocess the dataset (handle missing values, encode categorical variables, scale data).
3. Train a **Logistic Regression** model using LogisticRegression().
4. Evaluate performance using **Accuracy, Precision, Recall, F1-score**.
5. Plot the **Confusion Matrix** and **ROC Curve**.

Lab Assignment 5: Implementing K-Nearest Neighbors (KNN)

- **Tasks:**

1. Load the **Iris dataset** and split it into training and testing sets.
2. Train a **KNN classifier** using KNeighborsClassifier().
3. Experiment with different values of **K** and evaluate performance.
4. Visualize decision boundaries using a **scatter plot**.

Lab Assignment 6: Decision Trees and Random Forests

- **Tasks:**

1. Train a **Decision Tree classifier** on the Titanic dataset.
2. Visualize the tree structure using plot_tree().
3. Train a **Random Forest classifier** and compare performance with the decision tree.
4. Determine the **feature importance** using feature_importances_.

Module 4	Neural Networks and Deep Learning	Quiz	Implementation	6 Sessions

Lab Assignment 7: Introduction to Perceptron and Activation Functions

Tasks:

1. Implement a **single-layer perceptron** using NumPy.
2. Train the perceptron to classify **AND, OR, XOR** gates.
3. Experiment with different **activation functions** (Sigmoid, ReLU, Tanh).
4. Visualize decision boundaries.

Lab Assignment 8: Building a Simple Neural Network with Keras

Tasks:

1. Load the **MNIST dataset** from keras.datasets.
2. Preprocess the data (normalize pixel values, reshape input).
3. Create a **fully connected neural network** using Sequential API.
4. Train and evaluate the model using **categorical cross-entropy loss** and **accuracy**.

Lab Assignment 9: Implementing CNN from Scratch

Tasks:

1. Load the **CIFAR-10 dataset**.
2. Build a CNN with **Conv2D, MaxPooling2D, Flatten, Dense, Dropout** layers.
3. Use **Adam optimizer** and **categorical cross-entropy loss**.
4. Train and visualize loss/accuracy curves.

Lab Assignment 10: Image Augmentation & Regularization

Tasks:

1. Apply **data augmentation** (rotation, zoom, flipping) using ImageDataGenerator.
2. Add **dropout and batch normalization** to prevent overfitting.
3. Compare model performance with and without augmentation.

Lab Assignment 11: Transfer Learning with Pre-trained Models

Tasks:

1. Use **VGG16 or ResNet50** pre-trained on ImageNet.
2. Replace the output layer to classify **new images**.
3. Freeze earlier layers and fine-tune deeper layers.
4. Evaluate the model on a custom **dataset (e.g., Cats vs. Dogs)**.

Lab Assignment 12: Implementing RNN for Text Classification

Tasks:

1. Load **IMDB movie reviews dataset** from keras.datasets.
2. Preprocess text (tokenization, padding sequences).

3. Build an **RNN** with **Embedding, SimpleRNN, Dense** layers.
4. Train and evaluate the model.

Lab Assignment 13: Building an LSTM for Time Series Prediction

Tasks:

1. Load a **time series dataset** (e.g., stock prices, temperature data).
2. Preprocess the data (normalize, reshape).
3. Build an **LSTM-based model**.
4. Predict future values and visualize trends.

Targeted Application & Tools that can be used:

Applications:

1. **Data Preprocessing**: Clean and manipulate data from various sources such as CSV, Excel, SQL databases, and APIs.
- **Exploratory Data Analysis (EDA)**: Gain insights into datasets by identifying trends, patterns, and outliers.
- **Predictive Modeling**: Build models for classification (e.g., spam detection) and regression (e.g., house price prediction).
- **Clustering**: Group data into clusters for unsupervised learning tasks (e.g., customer segmentation).
- **Model Evaluation**: Assess model performance using appropriate metrics such as accuracy, precision, recall, and F1-score.

Tools:

- **Pandas**: For data manipulation and cleaning (e.g., handling missing values, merging datasets).
- **NumPy**: For numerical operations and working with arrays and matrices.
- **Matplotlib**: For creating static, animated, and interactive visualizations.
- **Seaborn**: For advanced data visualizations (e.g., heatmaps, pair plots).
- **Plotly**: For creating interactive visualizations, especially useful for large datasets.
- **Scikit-learn**: The go-to library for implementing machine learning algorithms (e.g., linear regression, decision trees, k-means clustering).
- **XGBoost**: For advanced gradient boosting models, particularly for large-scale machine learning tasks.
- **TensorFlow** (for deep learning in Module 4): A powerful open-source library for building machine learning and deep learning models.
- **Keras**: High-level neural network API, built on top of TensorFlow, to easily create deep learning models.

NLTK: The Natural Language Toolkit for various text processing tasks like tokenization, stemming, and part-of-speech tagging.

spaCy: A fast NLP library for advanced NLP tasks such as named entity recognition and dependency parsing.

Transformers (by Hugging Face): A powerful library for using pre-trained Transformer-based models like BERT, GPT, and others for advanced NLP tasks.

Text Book(s):

T1: ***Essentials of Python for Artificial Intelligence and Machine Learning*** *by Pramod Gupta and Anupam Bagchi*
Reference(s): 1. "Artificial Intelligence with Python" – Prateek Joshi 2. "Python Machine Learning" – Sebastian Raschka & Vahid Mirjalili 3. "Hands-On Artificial Intelligence with Python" – Teet Straus 4. "Deep Learning for Coders with Fastai and PyTorch" – Jeremy Howard & Sylvain Gugger

Course Code: **CSE7000**	**Course Title:** **Internship** **Type of Course:**	**L- T-P-C**	-	-	-	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 Skills through Experiential Learning techniques.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Identify the engineering problems related to local, regional, national or global needs. (Understand) 2. Apply appropriate techniques or modern tools for solving the intended problem. (Apply) 3. Design the experiments as per the standards and specifications. (Analyze) 4. Interpret the events and results for meaningful conclusions. (Evaluate)					

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

CSE2508</td><td colspan="2">Course Title: Mobile Applications and Development

Type of Course: Theory</td><td>L- T-P-C</td><td>2</td><td>0</td><td>0</td><td>2</td></tr>
<tr><td>Version No.</td><td colspan="7">2.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="7">CSE3514 Object Oriented Programming Using Java</td></tr>
<tr><td>Anti-requisites</td><td colspan="7">NIL</td></tr>
<tr><td>Course Description</td><td colspan="7">The course deals with the basics of android platform and application life cycle. The goal of the course is to develop mobile applications with Android containing at least one of the following phone material components: GPS, accelerometer or phone camera, use simple GUI applications and work with database to store data locally or in a server. Topics include user interface design; user interface building; input methods; data handling; network techniques and URL loading; GPS and motion sensing. Android application framework and deployment. Power management, Screen resolution, Touch interface, Store data on the device.</td></tr>
<tr><td>Course Objective</td><td colspan="7">The objective of the course is to familiarize the learners with the concepts of Mobile Applications and Development as mentioned above and attain Employability Skills through Experiential Learning Techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="7">On successful completion of the course the students shall be able to:

1. Discuss the fundamentals of mobile application development and its architecture. (Comprehension)

2. Illustrate mobile applications with appropriate android view. (Application)

3. Demonstrate the use of services, broadcast receiver, Notifications and content provider.(Application)

4. Apply data persistence techniques, to perform CRUD operations. (Application)
5. Use advanced concepts for mobile application development. (Application)</td></tr>
<tr><td>Course Content:</td><td colspan="7"></td></tr>
<tr><td>Module 1</td><td colspan="2">Introduction and Architecture of Android</td><td>Assignment</td><td colspan="2">Simulation/Data Analysis</td><td>5 Sessions</td></tr>
<tr><td colspan="8">Topics:

Android: History and features, Architecture, Development Tools, Android Debug Bridge (ADB), and Life cycle.</td></tr>
</table>

Module 2	User Interfaces, Intent and Fragments	Term paper/Assignment	Simulation/Data Analysis	6 Sessions
Topics: Views, Layout, Menu, Intent and Fragments.				
Module 3	Components of Android	Term paper/Assignment	Simulation/Data Analysis	6 Sessions
Topics: Activities, Services, Broadcast receivers, Content providers, User Navigation				
Module 4	Notifications and Data Persistence	Term paper/Assignment	Simulation/Data Analysis	6 Sessions
Topics: Notification, Shared Preferences, SQLite database, Android Room with a View, Firebase.				
Module 5	Advance App Development	Term paper/Assignment	Simulation/Data Analysis	7 Sessions
Topics: Graphics and Animation, App Widgets, Sensors, Performance, Location, Places, Mapping, Custom Views, Canvas.				
Targeted Application & Tools that can be used: Applications: Native Android Applications Native iOS Applications Cross Platform mobile Apps Mobile web Applications				
Text Book(s): T1. Pradeep kothari "Android Application Development - Black Book", dreamtechpress T2. Barry Burd (Author), "Android Application Development" ALL – IN – ONE FOR Dummies T3. Jeff Mcherter (Author),Scott Gowell (Author), "Professional mobile Application Development" paperback, Wrox - Wiley India Private Limited T4. Wei-Meng Lee (Author) "Beginning Android Application Development" Wrox – Wiley				

India Private Limited

Reference(s):

1. Bill Phillips, Chris Stewart, and Kristin Marsicano (Author) “Android Programming” 3rd edition,

2017.The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 5. The Big Nerd Ranch Guide, by”

2. Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd,

2014.

3. Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O’Reilly SPD

Publishers, 2015.

4. J F DiMarzio, “Beginning Android Programming with Android Studio”, 4th Edition, Wiley India Pvt

Ltd, 2016. ISBN-13: 978-8126565580

5. Anubhav Pradhan, Anil V Deshpande, “ Composing Mobile Apps” using Android, Wiley 2014,

ISBN: 978-81-265-4660-2

6. Reto Meier “Professional Android Application Development”

E-Resources: https://puniversity.informaticsglobal.com/login Or http://182.72.188.193/

Course Code: **CSE2509**	**Course Title:** Mobile Applications and Development Lab **Type of Course:** Lab	**L- T-P-C**	0	**0**	4	**2**
Version No.	2.0					
Course Pre-requisites	EEE1007 Problem Solving using JAVA					
Anti-requisites	**NIL**					
Course Description	The course provides hands-on experience in designing, developing, and deploying mobile applications for Android and iOS platforms. Students will					

	work with native development frameworks such as Android Studio (Java/Kotlin) and Xcode (Swift), as well as explore cross-platform tools like Flutter or React Native.			
Course Objective	The objective of the course is to develop Native and Cross-Platform Mobile Applications, design Interactive and Responsive User Interfaces, integrate Backend Services and APIs, implement State Management and Performance Optimization, ensure Mobile App Security and Data Protection			
Course Outcomes	On successful completion of the course the students shall be able to: 1. Develop Functional Mobile Applications 2. Design and Implement Interactive UIs 3. Integrate Cloud Services and APIs 4. Integrate Backend Systems and Data Management 5. Deploy, Publish, and Maintain advanced Mobile Application			
Course Content:				
Module 1	Introduction and Architecture of Android	Assignment	Simulation/Data Analysis	8 Sessions
1.a. Design an app to read user inputs using edit text and display the result of arithmetic operations using toast message. 1.b. Create an android app to calculate the current age of yourself, select your DOB using date picker. 2. Design an app to input your personal information. Use an autocomplete text view to select your place of birth.				
Module 2	User Interfaces, Intent and Fragments	Term paper/Assignment	Simulation/Data Analysis	13 Sessions
3. a. Design an app to select elective course using spinner view and on click of the display button, toast your ID and selected elective course. 3. b. Design a restaurant menu app to print the total amount of orders.				
Module 3	Components of Android	Term paper/Assignment	Simulation/Data Analysis	13 Sessions

4. Develop an android app that uses intent to maintain the following scenario.

Check the eligibility criteria for voting. Input the Aadhar no., Name & age in the first activity. If the age is above 18, display the voter's detail in the second activity. Else, display, “You are not eligible to vote” in the second Activity.

5. Demonstrate the use of fragment with list of buttons representing various colors, and on click of these buttons, the appropriate color is filled in the next fragment. Create an Android application to input the vitals of a person (temperature, BP). If the vitals are abnormal, give proper notification to the user.

6. Create an android app to for movie ticket booking. Save the user name of the customer using shared preferences. After completion of booking, retrieve the username from the shared preferences and print the ticket details.

Module 4	Notifications and Data Persistence	Term paper/Assignment	Simulation/Data Analysis	13 Sessions

7. Create an android application to manage the details of students' database using SQLite.Use necessary UI components, which perform the operations such as insertion, modification, removal and

view.Presidency University needs an APP for Admission eligibility checking for students, for that you need to take the following information from the Student: registration ID, physics, chemistry and mathematics marks (PCM), fees is allotted as below criteria.

PCM (Total marks %) Fee concession

90 above 80 %

70 to 89 60 %

Below 69 % no concession

On click on the button “Registration” details should be stored in the database using SQLite. Create button DISPLAY ALL (full students list) on click on the button it should display the students list per the fee

concession.

8. A company need to design an app that plays soft music automatically in the background. Create an app to achieve this functionality.

9. Create an android application such that your view object in the Activity can be Animated with fade-in effect. Create an appropriate XML file named fade-in and write the application to perform the property animation.

Module 5	Advance App Development	Term paper/Assignment	Simulation/Data Analysis	13 Sessions

10. Demonstrate how to send SMS and email.

11. Create an android application to transfer a file using WiFi. Create an android application "Where am I" with an Activity that uses the GPS Location provider to find the device's last known location.

Targeted Application & Tools that can be used:

Applications:

1. Native Android Applications (Java/Kotlin)

- Android Mobile Apps built for Android smartphones and tablets using Java or Kotlin programming languages.
- Target audience: Android users.

2. Native iOS Applications (Swift)

- iOS Mobile Apps designed for iPhone and iPad using Swift.
- Target audience: iOS users (Apple ecosystem).

3. Cross-Platform Mobile Apps (Flutter, React Native)

- Cross-platform apps designed to run on both Android and iOS from a single codebase using frameworks like Flutter or React Native.
- Target audience: Users on both Android and iOS platforms.

4. Mobile Web Applications (Progressive Web Apps - PWA)

- Mobile-optimized web applications using HTML5, CSS3, and JavaScript that run in a browser with native-like functionality (offline support, push notifications).
- Target audience: Users accessing apps via mobile browsers.

Development Tools and Frameworks

1. Integrated Development Environments (IDEs)

- Android Studio (for Android): The official IDE for Android development, supporting Java, Kotlin, and Android SDK.
- Xcode (for iOS): The official IDE for iOS development with Swift and Objective-C, providing a comprehensive suite of development tools for iPhone/iPad applications.
- Visual Studio Code (VS Code): Lightweight IDE for working with Flutter, React Native, and web development projects.

2. Cross-Platform Development Frameworks

- Flutter: Open-source UI framework by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.

- React Native: Open-source framework developed by Facebook for building cross-platform apps with JavaScript and React.

3. Backend & Cloud Tools

- Firebase: Google's backend-as-a-service (BaaS) platform offering authentication, real-time databases, cloud storage, and push notifications for mobile apps.
- AWS Amplify: Cloud platform for backend services (API, storage, authentication) and mobile deployment.
- SQLite / Realm: Local storage solutions for mobile apps to manage data storage and retrieval on-device.

4. Mobile App Testing and Debugging Tools

- Android Emulator (for Android): A virtual device to run and test Android apps without needing physical devices.
- Xcode Simulator (for iOS): A tool to simulate different iOS devices and test apps during development.
- Appium: Open-source tool for automated testing across native, hybrid, and mobile web applications.

5. Version Control and Collaboration

- Git: Version control system for managing code changes and collaborating with teams.
- GitHub / GitLab / Bitbucket: Online platforms for hosting Git repositories, collaboration, and version control management.

6. Mobile App Deployment Tools

- Google Play Console: For managing Android app publishing, distribution, and monitoring.
- Apple App Store Connect: For managing iOS app submissions, reviews, and releases on the Apple App Store.

7. UI/UX Design Tools

- Figma / Adobe XD: Tools for UI/UX design and wireframing to create the visual elements of mobile applications before development.
- Sketch: Vector-based design tool for iOS UI design and prototyping

Text Book(s):

T1. Pradeep kothari “Android Application Development - Black Book”, dreamtechpress

T2. Barry Burd (Author), “Android Application Development” ALL – IN – ONE FOR Dummies

T3. Jeff Mcherter (Author),Scott Gowell (Author), “Professional mobile Application Development” paperback, Wrox - Wiley India Private Limited

T4. Wei-Meng Lee (Author) “Beginning Android Application Development” Wrox – Wiley India Private Limited

Reference(s):

1. Bill Phillips, Chris Stewart, and Kristin Marsicano (Author) “Android Programming” 3rd edition,

2017.The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 5. The Big Nerd Ranch Guide, by”

2. Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd, 2014.

3. Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O’Reilly SPD Publishers, 2015.

4. J F DiMarzio, “Beginning Android Programming with Android Studio”, 4th Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978-8126565580

5. Anubhav Pradhan, Anil V Deshpande, “ Composing Mobile Apps” using Android, Wiley 2014, ISBN: 978-81-265-4660-2

6. Reto Meier “Professional Android Application Development”

E-Resources: https://puniversity.informaticsglobal.com/login Or http://182.72.188.193/

Course Code: CSE2510	**Course Title:** Competitive Programming and Problem Solving **Type of Course: Program Core**	L-T-P-C	0	0	4	2
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	**NIL**					
Course Description	The **Competitive Programming and Problem Solving** course equips students with efficient problem-solving skills for coding competitions and real-world challenges. Starting with brute-force solutions, students learn to optimize time and space complexity					

	using advanced techniques like dynamic programming, greedy algorithms, and backtracking. Hands-on practice on platforms like CodeChef and Codeforces helps tackle problems involving number theory, data structures, and algorithmic paradigms. By understanding CP constraints and fostering a strategic mindset, students gain the confidence to excel in competitions, technical interviews, and practical applications.
Course Out Comes	On successful completion of the course the students shall be able to: CO1 : Understanding the issues of online platforms and Competitive Programming (CP) and developing brute force coding for commonly asked CP problems. CO2 : Analyzing the space and time complexity of brute force solutions and designing efficient solutions. CO3 : Evaluating the applicability of suitable algorithmic approaches to solve relevant CP problems. CO4: Creating efficient solutions of CP problems using the learnt algorithmic approaches.
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Competitive Programming and Problem Solving** and attain **Skill Development** through **Experiential Learning** techniques.

Module 1: Introduction to Competitive Programming

Overview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforces etc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL

Module 2: Number Theory for Problem-Solving

Use of Number Theory for problem-solving: reducing time/space complexity of brute force
coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient coding
for Permutation Combination; XORing based and pattern-based solutions.

Module 3: Optimizing Time & Space Using Sequential Storage

Coding for Optimizing time and Space using Sequential Storage: two pointer approach;
problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string
matching algorithms; Kadane's algo, stacks, priority-queues and hashing based efficient coding;
median based problems and alternate solutions.

Module 4: Non-Linear Data Structures

Applying Non-Linear Data Structures for real-life problems: design of efficient solutions for
problems such as finding loops in a linked list, memory efficient DLL, block reversal in

LL; problem
solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path
algos for CP problems with reduced time/space complexity.

Module 5: Problem Solving using Advanced Topics

CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and
applying them for CP problems using bottom-up dynamic programming.

List of Laboratory Tasks:

1. You are given the finishing times of 'N' runners in a marathon. Write a program to find the runner who finished in the third position. **Focus:** Basic data structures (arrays), sorting algorithms (e.g., insertion sort, selection sort), and basic input/output.
2. In the same marathon, you are given the finishing times of 'N' runners and their bib numbers. Write a program to efficiently find the top 10 runners and their corresponding bib numbers. **Focus:** Efficient sorting algorithms (e.g., merge sort, quick sort), data structures like priority queues, and optimizing for large datasets.
3. A library maintains a list of books with their unique IDs. Write a program to check if a given book ID is present in the library. Focus: Searching algorithms (linear search), basic data structures (arrays or lists).
4. The library wants to implement a system to quickly find books by their titles. Suggest an efficient data structure (e.g., a hash table or a trie) and explain how to implement it to achieve fast book lookups. Focus: Understanding the trade-offs between different data structures, choosing the most appropriate data structure for a specific problem, and implementing efficient search operations.
5. An online store sells products with different prices. Write a program to calculate the total cost of a given list of products. **Focus:** Basic arithmetic operations, working with arrays or lists to store product prices.
6. The online store offers discounts based on the total purchase amount. Design an algorithm to efficiently calculate the final cost of an order, considering different discount rules (e.g., percentage discounts, fixed amount discounts, tiered discounts). Focus: Algorithmic design, conditional statements, handling complex scenarios with multiple rules, and potentially using dynamic programming techniques for optimization.
7. You are given two integers, 'a' and 'm'. Calculate 'a' raised to the power 'm' modulo a large prime number 'p'. **Focus:** Basic modular arithmetic operations (modular exponentiation), understanding the modulo operator.
8. In a secure communication system, you need to efficiently compute the modular exponentiation for very large values of 'm'. Implement and analyze the efficiency of the binary exponentiation algorithm for this task. **Focus:** Efficient algorithms for modular exponentiation (binary exponentiation), time complexity analysis, and understanding the importance of efficient algorithms in cryptography.
9. You have a deck of 'N' cards. Calculate the total number of possible hands of size 'K' that can be drawn from the deck. **Focus:** Basic combinatorics (combinations), factorial calculations.
10. In a card game, you need to calculate the probability of drawing certain combinations of cards (e.g., a pair, a three-of-a-kind) from a shuffled deck. Design an efficient algorithm to calculate these probabilities. **Focus:** Advanced combinatorics (permutations and combinations with repetitions), probability calculations, and optimizing calculations to avoid overflows.

11. You are given a network of devices represented as a graph. Determine if there is a path between two given devices in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search).
12. In a secure network, you need to detect and isolate compromised devices. Design an algorithm that efficiently identifies devices that exhibit anomalous behavior (e.g., unusual traffic patterns) using XOR-based techniques for data comparison and pattern matching. **Focus:** Applying XOR operations for data comparison and pattern recognition, understanding the properties of XOR (e.g., commutative, associative), and designing algorithms for network anomaly detection.
13. You are given an array representing the speeds of cars on a highway. Find the minimum time required for all cars to pass a certain point. **Focus:** Basic array traversal, finding the minimum element in an array.
14. In a more realistic scenario, cars have different lengths. Implement a two-pointer approach to simulate the movement of cars and determine the minimum time for all cars to pass a given point. **Focus:** Two-pointer technique, simulating real-world scenarios with arrays, optimizing time complexity.
15. Given a string, find the number of occurrences of a specific substring within the string. **Focus:** Basic string manipulation, string matching (brute-force approach).
16. Implement the KMP (Knuth-Morris-Pratt) string matching algorithm to efficiently find all occurrences of a given pattern within a large text document. **Focus:** Advanced string matching algorithms, understanding the concept of the "next" array in KMP, optimizing for large input sizes.
17. An online auction platform receives bids for different items. Implement a data structure (e.g., a priority queue) to efficiently track the highest bid for each item. **Focus:** Priority queues, insertion and extraction operations on priority queues, basic implementation of a priority queue using an array or a suitable library.
18. The auction platform needs to handle a large number of bids concurrently. Design and implement a system that efficiently processes bids, updates the highest bid for each item, and handles potential race conditions. **Focus:** Concurrent data structures and algorithms, thread safety, handling race conditions, optimizing for high-throughput scenarios.
19. A social network can be represented as a graph where users are nodes, and connections between users are edges. Write an algorithm to find if two given users are connected in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search), basic graph representation (adjacency list or adjacency matrix).
20. In a large social network, efficiently finding the shortest path between two users is crucial. Implement Dijkstra's algorithm to find the shortest paths between users in the network, considering edge weights (e.g., representing the strength of connections). **Focus:** Shortest path algorithms (Dijkstra's algorithm), graph algorithms with weighted edges, optimizing for large graphs.
21. A file system can be modeled as a tree structure. Implement a function to traverse the file system and print the names of all files and directories. **Focus:** Tree traversal algorithms (depth-first search or breadth-first search), basic tree representation (using nodes and pointers).
22. Design and implement a file system that supports efficient operations like creating directories, deleting files, and finding files based on their names or paths. Consider using a combination of tree structures and hash tables for efficient indexing and searching. **Focus:** Designing and implementing file system structures, using multiple data structures together, optimizing for common file system operations.

23. An online shopping cart can be represented as a tree, where each node represents an item or a category of items. Write an algorithm to calculate the total price of all items in the shopping cart. **Focus:** Tree traversal, calculating sums within a tree structure.
24. Implement a system that allows customers to apply discounts and coupons to their shopping carts. Consider using a combination of trees and other data structures (e.g., hash tables) to efficiently apply discounts and calculate the final price. **Focus:** Applying discounts and promotions to tree-like structures, efficient implementation of discount rules, optimizing for complex pricing scenarios.
25. In a social network, users can form groups. Given a list of friendships, determine if all users in a specific group are connected (directly or indirectly) through friendships. **Focus:** Disjoint set union (DSU) data structure, basic connectivity checks.
26. Design an efficient algorithm to find the minimum number of new friendships needed to connect all users in the social network into a single, connected component. **Focus:** Applying DSU for finding connected components, greedy algorithms, optimization for minimizing connections.
27. A treasure hunt involves a series of clues leading to the final treasure. Given a list of possible paths and their associated costs, find the cheapest path to reach the treasure. **Focus:** Greedy algorithms (e.g., Dijkstra's algorithm for shortest paths), basic graph representation.
28. In a more complex treasure hunt, there are time constraints associated with each path. Design an algorithm to find the fastest path to the treasure while considering both path costs and time constraints.
    Focus: Combining greedy approaches with other techniques (e.g., priority queues), handling multiple constraints, optimizing for time-critical scenarios.
29. In a simplified chess game with only rooks, determine the minimum number of moves required for a rook to reach a specific target square on an empty board. **Focus:** Breadth-first search (BFS) on a graph (the chessboard), basic graph traversal.
30. In a more realistic chess game with multiple pieces and obstacles, implement a minimax algorithm with alpha-beta pruning to determine the best move for a player. **Focus:** Game tree search, minimax algorithm, optimization techniques like alpha-beta pruning, handling complex game states.

Targeted Application & Tools that can be used:

1. C or C++ Compiler (g++): The standard compiler for CP. Familiarize students with compilation flags (e.g., -O2 for optimization).
2. IDE (Integrated Development Environment): Code:: Blocks, Visual Studio, CLion, or similar IDEs. These provide debugging capabilities, code completion, and other helpful features.
3. Online Judges (CodeChef, Codeforces, LeetCode, HackerRank): Essential for practicing and submitting solutions.
4. Debugger (gdb): Crucial for understanding code execution and finding bugs. Origin, excel and Mat lab soft wares for programming and data analysis.
5. Number Theory Libraries: Some libraries provide pre-built functions for number theory operations (though often it's better to implement them yourself for learning).
6. Wolfram Alpha: A useful tool for verifying number theory calculations and exploring concepts.
7. **String Libraries:** Familiarize students with the string manipulation functions available in C++.

8. **Graph Visualization Tools:** Tools like Graphviz can be helpful for visualizing graphs and understanding graph algorithms.
9. **DP Debugging Techniques:** Practice debugging DP solutions, as they can be complex. Visualizing the DP table can be helpful.

Text Books:

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

2 **"Data Structures and Algorithms in Java: A Project-Based Approach" –** ***Dan S. Myers, Cambridge University Press***

Reference Books:

1. **Data Structures and Algorithmic Thinking with Python/C++/Java",** ***Narasimha Karumanchi, 5th Edition, Career Monk, 2017.***
2. **Introduction to Algorithms, Thomas H. Cormen (Author), Charles E. Leiserson (Author), Ronald L. Rivest , fourth edition April 2022**

Web Resources

1. **https://nptel.ac.in/courses/106106231**
2.

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

Course Code: CDV2507	Continuous Integration and Continuous Delivery Pipelines Lab	**L-T-P-C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	CDV1700 DevOps Foundations					
Anti-requisites	NIL					
Course Description	This lab course is designed to provide practical exposure to CI/CD practices in modern software development. It emphasizes hands-on experience in setting up automated build, test, and deployment pipelines using open-source tools. Students will work with source control, configuration management, testing frameworks, containerization, and monitoring solutions to understand the entire DevOps pipeline lifecycle.					

Course Objectives	1. To familiarize students with the CI/CD concepts through practical tools and workflows. 2. To enable implementation of automated build and test pipelines using version control systems. 3. To equip students with the skills to containerize applications and integrate deployment in a pipeline. 4. To develop proficiency in monitoring, error tracking, and performance management in pipelines.
Course Out Comes	**CO1**: Understand and set up basic CI/CD pipelines using tools like Jenkins and Git. **CO2**: Integrate automated testing and static analysis tools within pipelines. **CO3**: Implement container-based deployment using Docker and Kubernetes in CI/CD. **CO4**: Apply monitoring and logging tools to manage and optimize CI/CD workflows.

List of Open Source Tools Used

Jenkins – For building and automating CI/CD pipelines.

GitLab/GitHub Actions – For version control and pipeline automation.

Docker – For containerization of applications in the delivery pipeline.

SonarQube – For integrating code quality and static analysis into pipelines.

List of Experiments

1	**Introduction to DevOps: Install and configure Git and Jenkins**
2	**Create a simple pipeline in Jenkins using a sample Java/Python project**
3	**Integrate GitHub or GitLab repository with Jenkins for source control and automated build**
4	**Automate unit testing with JUnit or PyTest in a Jenkins pipeline**
5	**Perform static code analysis using SonarQube in Jenkins pipeline**
6	**Create a multi-stage pipeline with Jenkinsfile**
7	**Set up GitHub Actions CI workflow for Node.js/Python project**
8	**Build Docker containers for application and push to DockerHub**
9	**Integrate Docker with Jenkins: Build and deploy Docker containers**
10	**CI/CD with Kubernetes: Deploy containerized apps using Helm and Jenkins**

11	**Canary and Blue-Green deployment setup using Jenkins and Kubernetes**
12	**Infrastructure as Code: Use Terraform to provision infrastructure for CI/CD**
13	**Add monitoring tools like Prometheus and Grafana in a pipeline**
14	**CI/CD security: Use secrets management and Snyk for vulnerability scans**
15	**Final mini-project: Build full CI/CD pipeline for a microservice-based app**

Reference Books

R1: Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 2nd Edition, 1996.
R2: Douglas R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 4th Edition, 2018.
R3: Charlie Kaufman, Radia Perlman, and Mike Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall, 2nd Edition, 2002.
R4: Christof Paar, Jan Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer, 2010.

Web Resources

W1: https://cryptography.io/ – Python cryptographic library and resources.
W2: https://nvlpubs.nist.gov/ – NIST publications on cryptographic standards.
W3: https://www.owasp.org/ – Open Web Application Security Project for best practices.
W4: https://www.tutorialspoint.com/cryptography/index.htm – Cryptography basics and tutorials.
W5: https://www.coursera.org/learn/crypto – Stanford's free online cryptography course on Coursera.

Course Code: CDV2506	Continuous Integration and Continuous Delivery Pipelines	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	CDV1700 DevOps Foundations					
Anti-requisites	NIL					
Course Description	This course introduces the principles and practices of Continuous Integration and Continuous Delivery (CI/CD) in modern software development. Students will learn how to automate the software build, test, and deployment processes using industry-standard tools. Emphasis will be given to DevOps culture, pipeline as code, infrastructure automation, and integrating quality assurance into CI/CD workflows.					
Course Objectives	Understand the fundamental concepts of CI/CD and its role in DevOps.					

	Explore various CI/CD tools and pipelines for software build, testing, and deployment automation. Analyze pipeline performance, integration with testing tools, and deployment strategies. Apply CI/CD practices to real-world applications with automation and monitoring.
Course Out Comes	CO1 (Understand): Describe the role of CI/CD in modern software engineering and DevOps culture. CO2 (Analyze): Analyze different CI/CD tools, pipelines, and strategies for continuous delivery and deployment. CO3 (Apply): Develop automated CI/CD pipelines integrating version control, build, and test tools. CO4 (Apply): Implement secure and scalable deployment strategies using infrastructure automation tools.

Module 1	**Introduction to CI/CD and DevOps**		**Understand**	**No. of Sessions: 10**
CI/CD overview, Agile and DevOps synergy, DevOps lifecycle, Benefits of CI/CD, Software release cycle, CI vs CD, Key metrics in CI/CD.				
Module 2	**CI/CD Tools and Pipeline Architecture**		**Analyse**	**No. of Sessions: 12**
Jenkins, GitLab CI, GitHub Actions, CircleCI, Travis CI, Pipeline as code (Jenkinsfile), YAML scripting, Pipeline stages and best practices.				
Module 3	**Build Automation, Testing, and Security Integration**		**Apply**	**No. of Sessions: 12**
Build automation (Maven, Gradle), Unit testing (JUnit, PyTest), Static analysis (SonarQube), Integration with Docker, Secrets management, Security testing (OWASP, Snyk).				
Module 4	**Deployment, Monitoring, and Infrastructure Automation**		**Apply**	**No. of Sessions: 11**
Docker and Kubernetes in CI/CD, Blue-green & Canary deployment, Helm, Terraform, Infrastructure as Code (IaC), Monitoring with Prometheus and Grafana, Real-world CI/CD case studies.				

Textbooks

T1: Erich F. Poppendieck & Mary Poppendieck, *"Lean Software Development: An Agile Toolkit"*, Addison-Wesley, 2003.

T2: Gene Kim, Jez Humble, Patrick Debois, and John Willis, *"The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations"*, IT Revolution Press, 2016.

Reference Books

R1: Jez Humble and David Farley, *"Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation"*, Addison-Wesley, 2010.
R2: Mikael Krief, *"Learning DevOps: Continuously Deliver Better Software"*, Packt Publishing, 2021.
R3: Paul Swartout, *"Continuous Delivery and DevOps – A Quickstart Guide"*, Packt Publishing, 2012.
R4: Viktor Farcic, *"The DevOps Toolkit: Building the DevOps Culture"*, Leanpub, 2016.

Web Resources

W1: Jenkins Documentation – https://www.jenkins.io/doc
W2: GitHub Actions Docs – https://docs.github.com/actions
W3: DevOps.com (CI/CD Articles & Tutorials) – https://devops.com/category/cicd/
W4: SonarQube Docs – https://docs.sonarqube.org
W5: Kubernetes CI/CD Guide – https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Course Code: **CDV2508**	Edge Computing and DevOps	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	CDV1700 DevOps Foundations					
Anti-requisites	NIL					
Course Description	This course introduces the integration of Edge Computing and DevOps, focusing on how computational workloads are distributed across edge and cloud environments. It explores real-time data processing at the edge, latency reduction, and automation in deployment cycles using DevOps principles. Students will learn edge system architecture, containerization, orchestration, CI/CD pipeline automation, and performance monitoring in edge-centric applications.					
Course Objectives	To provide foundational understanding of Edge Computing concepts and architectures. To explore DevOps methodologies and tools in the context of edge-based deployments. To understand the role of containers, microservices, and orchestration in edge applications.					

	To enable automation, monitoring, and performance tuning of edge-cloud integrated systems
Course Out Comes	**CO1 (Understand)**: Explain the fundamentals and significance of edge computing and DevOps in distributed systems. **CO2 (Analyze)**: Evaluate architectural models and deployment strategies for edge computing using DevOps pipelines. **CO3 (Apply)**: Implement edge-based applications using containerization and orchestration tools. **CO4 (Apply)**: Automate CI/CD pipelines and monitor edge-cloud systems using modern DevOps practices.

Module 1	**Fundamentals of Edge Computing and DevOps**		**Understand**	**No. of Sessions: 10**
Edge computing concepts, edge vs cloud computing, use cases in IoT and real-time applications, DevOps principles, Agile, version control, CI/CD overview, benefits of DevOps in edge environments.				
Module 2	**Architecture of Edge Systems and DevOps Integration**		**Analyse**	**No. of Sessions: 12**
Edge nodes, edge-cloud continuum, data flow models, microservices architecture, edge orchestration, containerization (Docker), DevOps lifecycle for edge applications, Git workflows, DevOps tools for edge.				
Module 3	**CI/CD Pipeline Design for Edge Applications**		**Apply**	**No. of Sessions: 11**
Jenkins, GitLab CI, GitHub Actions for edge deployment, creating Jenkins pipelines, integrating testing, static analysis, building and pushing edge containers, Helm charts, Kubernetes on edge.				
Module 4	**Monitoring, Security and Case Studies**		**Apply**	**No. of Sessions: 12**
Security in edge environments, secrets management, edge-to-cloud monitoring with Prometheus and Grafana, anomaly detection, edge analytics, real-world applications in healthcare, smart cities, and autonomous vehicles.				

Textbooks

T1. Pethuru Raj, Anupama C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases”, CRC Press, 2017.

T2. Len Bass, Ingo Weber, Liming Zhu, “DevOps: A Software Architect's Perspective”, Addison-Wesley, 2015.

Reference Books

R1. Satya Shyam K Jayanty, "Edge Computing Systems with Kubernetes", Packt Publishing, 2021.
R2. Nebrass Lamouchi, “Hands-On DevOps with Linux”, Packt Publishing, 2020.
R3. Ernesto Garbarino, “DevOps for the Modern Enterprise”, IT Revolution Press, 2018.
R4. Amir H. Payberah, "Edge Intelligence in the Era of IoT", Springer, 2022.

Web Resources

W1. Edge Computing: What It Is and Why It Matters – IBM
W2. DevOps and Edge – Azure DevOps Documentation
W3. Kubernetes at the Edge – CNCF
W4. Docker for IoT and Edge
W5. EdgeX Foundry – Open Edge Computing Platform

Course Code: CDV2509	DevSecOps	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	CDV1700 DevOps Foundations					
Anti-requisites	NIL					
Course Description	DevSecOps integrates security practices within the DevOps process, aiming to build secure applications from the start of development to deployment. This course introduces the concepts of secure software development, automation of security testing, threat modeling, and secure CI/CD pipelines. It emphasizes tools and techniques for vulnerability detection, compliance, governance, and secure deployment strategies across cloud-native environments.					
Course Objectives	1. To understand the foundational principles and goals of DevSecOps and its importance in secure software delivery. 2. To learn the integration of security in the DevOps pipeline through automated tools and practices. 3. To explore secure software development life cycle (SSDLC), threat modeling, and code analysis techniques. 4. To implement compliance and governance in CI/CD pipelines and containerized environments.					
Course Out Comes	**CO1 (Understand):** Explain the concepts, importance, and practices of DevSecOps in modern software development. **CO2 (Analyze):** Identify security threats, vulnerabilities, and compliance issues in DevOps pipelines.					

	CO3 (Apply): Implement secure code analysis and automate security testing tools in CI/CD workflows. **CO4 (Apply):** Deploy and manage secure containerized applications with policy enforcement and governance.

Module 1	**Introduction to DevSecOps and Security Integration**		**Understand**	**No. of Sessions: 10**
DevOps vs DevSecOps, need for shift-left security, secure SDLC, security culture, security champions, automation in security integration, principles of secure by design.				
Module 2	**Threat Modeling and Secure Code Practices**		**Analyse**	**No. of Sessions: 12**
OWASP top 10, STRIDE and DREAD threat models, secure coding guidelines, static code analysis tools (SonarQube, Checkmarx), secure repositories, source code auditing.				
Module 3	**CI/CD Security Automation**		**Apply**	**No. of Sessions: 11**
Security in Jenkins, GitHub Actions, GitLab CI, container security with Trivy, automated security scans (SAST, DAST, SCA), dependency checking, DevSecOps pipelines.				
Module 4	**Container Security, Governance, and Compliance**		**Apply**	**No. of Sessions: 12**
Docker image hardening, Kubernetes security (RBAC, PSP), policy enforcement tools (OPA, Kyverno), compliance as code, audit logging, SIEM tools, case studies.				

Textbooks

T1. Jim Bird, *DevSecOps: A leader's guide to producing secure software without compromising flow, feedback and continuous improvement*, IT Revolution Press, 2020.
T2. Glenn Wilson, *DevSecOps: A Practical Guide to Secure Software Delivery*, Packt Publishing, 2020.

Reference Books

R1. Julien Vehent, *Securing DevOps: Security in the Cloud*, Manning Publications, 2018.
R2. Neil Madden, *API Security in Action*, Manning Publications, 2020.
R3. Josh Stella, *Cloud Security Handbook*, O'Reilly Media, 2023.
R4. Daniel Blander, *Security for DevOps*, Wiley, 2021.

Web Resources

W1. OWASP DevSecOps Guidelines
W2. DevSecOps by GitLab Docs
W3. SonarQube - Static Code Analysis
W4. CNCF DevSecOps Whitepapers
W5. DevSecOps Hub by Snyk

Course Code: CDV3400
Course Title: Fundamentals of Cloud Computing
L:T:P:C – 3:0:0:3
Prerequisite: CDV1700 – DevOps Foundations

Course Description
This course introduces the fundamental principles and practices of cloud computing. It covers cloud service models, deployment models, virtualization, containerization, and cloud architecture. The course also explores cloud platforms such as AWS, Azure, and Google Cloud, focusing on how to build, manage, and scale applications in cloud environments.

Course Objectives
— Understand core concepts and architecture of cloud computing
— Explore service models (IaaS, PaaS, SaaS) and deployment models
— Learn virtualization, containers, and cloud resource provisioning
— Apply cloud computing practices using leading cloud platforms

Course Outcomes
CO1 (Understand): Explain cloud computing principles, architecture, and service models
CO2 (Analyze): Compare deployment models and evaluate cost, scalability, and reliability
CO3 (Apply): Use virtualization and containerization tools for cloud deployment
CO4 (Apply): Work with cloud platforms to provision, monitor, and scale services

Course Content (45 Hours Total)

Module 1: Introduction to Cloud Computing – 11 Sessions (Understand)
Definition and evolution, Cloud characteristics, Service models: IaaS, PaaS, SaaS, Deployment models: Public, Private, Hybrid, Community, Benefits and limitations

Module 2: Cloud Architecture and Virtualization – 11 Sessions (Analyze)
Cloud architecture layers, Virtualization concepts: Hypervisors, VMs, Containers vs. VMs, Docker basics, Container orchestration overview, Cloud scalability, Elasticity, Fault tolerance

Module 3: Cloud Services and Resource Management – 11 Sessions (Apply)
Compute, Storage, and Network services in AWS/Azure/GCP, Identity and Access Management (IAM), Billing and metering, Cloud storage options, Auto-scaling and load balancing

Module 4: Cloud Platforms and Use Cases – 12 Sessions (Apply)
Hands-on with AWS Console, Azure Portal, Google Cloud Console, Cloud-native application development, Real-world use cases in e-commerce, banking, healthcare, education

Textbooks
T1: Rajkumar Buyya et al., *Mastering Cloud Computing: Foundations and Applications Programming*, McGraw-Hill Education, **2023**
T2: Lee Chao, *Cloud Computing: Principles and Paradigms*, CRC Press, **2022**

Reference Books

R1: Thomas Erl, *Cloud Computing: Concepts, Technology & Architecture*, Pearson, **2023**
R2: Arshdeep Bahga, Vijay Madisetti, *Cloud Computing: A Hands-On Approach*, Universities Press, **2022**
R3: Pethuru Raj, *Demystifying Cloud Computing*, Wiley, **2022**
R4: George Reese, *Cloud Application Architectures*, O'Reilly, **2021**

Web Resources
W1: https://aws.amazon.com/training
W2: https://learn.microsoft.com/en-us/training/azure
W3: https://cloud.google.com/training
W4: https://www.edx.org/course/introduction-to-cloud-computing
W5: https://cloudacademy.com

Course Code: CDV3401
Course Title: Version Control with Git & GitHub/GitLab
L:T:P:C – 3:0:0:3
Prerequisite: CDV1700 – DevOps Foundations

Course Description
This course focuses on modern version control practices using Git and collaborative platforms like GitHub and GitLab. It covers distributed version control concepts, branching strategies, pull requests, issue tracking, and CI/CD integration. Emphasis is placed on collaborative software development, change management, and automation in DevOps pipelines.

Course Objectives
— Understand distributed version control principles using Git
— Learn collaboration workflows with GitHub and GitLab
— Apply branching strategies, merge techniques, and conflict resolution
— Integrate version control with DevOps practices like CI/CD

Course Outcomes
CO1 (Understand): Describe the role of version control in modern software development
CO2 (Analyze): Compare and manage Git workflows, branching, and collaboration strategies
CO3 (Apply): Use Git and GitHub/GitLab to manage, track, and collaborate on code projects
CO4 (Apply): Integrate version control with CI/CD and DevOps automation tools

Course Content (45 Hours Total)

Module 1: Introduction to Git and Version Control – 11 Sessions (Understand)
Version control systems: centralized vs. distributed, Git basics: install, config, init, clone, status, Git objects and internal model, Commits, Logs, Aliases

Module 2: Branching, Merging, and Conflict Resolution – 11 Sessions (Analyze)
Creating and managing branches, Fast-forward and recursive merges, Merge conflicts and resolution, Rebase vs merge, Git stash, cherry-pick, reflog

Module 3: Collaborative Workflows with GitHub/GitLab – 11 Sessions (Apply)

Remote repositories, Forks, pull requests, code reviews, Issue tracking and labels, Branch protection rules, GitHub Actions/GitLab CI/CD basics

Module 4: Git in DevOps and Automation – 12 Sessions (Apply)
Git hooks and automation, Webhooks, CI/CD integration, Secrets and security, GitOps basics, Infrastructure as code versioning, Real-world project workflows (feature branching, trunk-based)

Textbooks
T1: Scott Chacon & Ben Straub, *Pro Git*, Apress, **2023**
T2: Brent Laster, *Professional Git*, Wiley, **2022**

Reference Books
R1: Kyle Banker, *Git Essentials*, Packt Publishing, **2023**
R2: Mariot Tsitoara, *Git Pocket Guide*, O'Reilly, **2022**
R3: Sarah Guthals, *GitHub For Dummies*, Wiley, **2023**
R4: GitLab Team, *GitLab Docs & DevOps Handbook*, GitLab Press, **2023**

Web Resources
W1: https://git-scm.com/doc
W2: https://docs.github.com
W3: https://docs.gitlab.com
W4: https://learngitbranching.js.org
W5: https://www.atlassian.com/git/tutorials

Course Code: CDV3402
Course Title: Serverless Computing
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course introduces the principles and practices of serverless computing as a cloud-native development model. It focuses on event-driven architectures, Function-as-a-Service (FaaS), API gateways, backend-as-a-service (BaaS), containerless deployments, and scalability. Students gain hands-on experience with platforms like AWS Lambda, Azure Functions, and Google Cloud Functions.

Course Objectives
— Understand the serverless paradigm and its advantages over traditional architectures
— Explore FaaS, BaaS, and event-driven workflows
— Develop and deploy serverless functions using major cloud platforms
— Apply serverless architectures to real-world application scenarios

Course Outcomes
CO1 (Understand): Explain the concepts and evolution of serverless computing
CO2 (Analyze): Compare traditional, microservices, and serverless architectures
CO3 (Apply): Build and deploy cloud-based serverless applications using FaaS platforms
CO4 (Apply): Implement event-driven services with monitoring, logging, and scaling

Course Content (45 Hours Total)

Module 1: Introduction to Serverless Computing – 11 Sessions (Understand)
Cloud computing evolution, Serverless vs. traditional vs. container-based models, Characteristics of serverless architecture, Benefits and challenges, Overview of FaaS and BaaS

Module 2: Serverless Architecture and Platforms – 11 Sessions (Analyze)
Event-driven architectures, FaaS internals, Serverless lifecycle, API Gateway, Function triggers (HTTP, storage, message queues), Stateless functions, Timeout and cold starts, Pricing models

Module 3: Developing Serverless Applications – 11 Sessions (Apply)
Deploying functions on AWS Lambda, Azure Functions, Google Cloud Functions, Using SDKs and CLI, Writing handler code, Environment variables, Deployment frameworks (Serverless Framework, SAM, Terraform)

Module 4: Use Cases, Integration & Monitoring – 12 Sessions (Apply)
Serverless use cases: data processing, real-time analytics, automation, ChatOps, CI/CD pipelines, Logging (CloudWatch, Stackdriver), Monitoring (X-Ray, OpenTelemetry), Security, Governance, Case studies

Textbooks
T1: Peter Sbarski, *Serverless Architectures on AWS*, Manning Publications, **2023**
T2: Rishabh Sharma, *Mastering Serverless Computing*, Packt Publishing, **2023**

Reference Books
R1: Gojko Adzic, *Serverless Handbook*, Leanpub, **2022**
R2: Slobodan Stojanovic, *Serverless Applications with Node.js*, Manning, **2022**
R3: John Chapin & Mike Roberts, *Serverless Framework: The Big Picture*, O'Reilly Media, **2022**
R4: Danilo Poccia, *AWS Lambda in Action*, Manning Publications, **2021**

Web Resources
W1: https://serverless.com
W2: https://docs.aws.amazon.com/lambda
W3: https://learn.microsoft.com/en-us/azure/azure-functions
W4: https://cloud.google.com/functions/docs
W5: https://faas-and-furious.io

Course Code: CDV3403
Course Title: Infrastructure as Code (IaC) with Terraform & Ansible
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course introduces the principles and practical implementation of Infrastructure as Code (IaC), focusing on automation tools such as Terraform and Ansible. Students will learn how to provision, configure, and manage cloud infrastructure using declarative and procedural approaches, integrating with DevOps workflows to ensure consistent, scalable, and reliable environments.

Course Objectives
— Understand the concepts and benefits of Infrastructure as Code
— Explore the features and syntax of Terraform and Ansible

— Develop and deploy infrastructure in public and private clouds
— Apply configuration management and provisioning to DevOps pipelines

Course Outcomes
CO1 (Understand): Describe the purpose, workflow, and tools of Infrastructure as Code
CO2 (Analyze): Compare declarative and imperative approaches for managing infrastructure
CO3 (Apply): Automate cloud provisioning using Terraform modules and workspaces
CO4 (Apply): Use Ansible for system configuration, orchestration, and remote execution

Course Content (45 Hours Total)

Module 1: Introduction to IaC and Cloud Infrastructure – 11 Sessions (Understand)
IaC definition and evolution, Benefits and challenges, Comparison of tools (Terraform, Ansible, Chef, Puppet), Overview of cloud service providers, Resource provisioning, Version control for IaC

Module 2: Terraform for Cloud Automation – 11 Sessions (Analyze)
Terraform basics: providers, resources, variables, state files, Modules, Workspaces, Provisioners, Dependency management, Remote backends, Best practices in writing Terraform scripts

Module 3: Configuration Management with Ansible – 11 Sessions (Apply)
Ansible architecture and YAML syntax, Inventory files, Playbooks and roles, Tasks and handlers, Modules (file, service, package, shell), Loops and conditionals, Idempotency, Ansible Galaxy

Module 4: IaC in DevOps Pipelines and Cloud Use Cases – 12 Sessions (Apply)
CI/CD with Terraform and Ansible, Secure secrets management (Vault, AWS Secrets Manager), Infrastructure testing (InSpec, Molecule), IaC in AWS/GCP/Azure, Real-world scenarios: auto-scaling, load balancing, HA architecture

Textbooks
T1: Yevgeniy Brikman, *Terraform: Up & Running*, O'Reilly Media, **2023**
T2: Lorin Hochstein, *Ansible: Up and Running*, O'Reilly Media, **2023**

Reference Books
R1: Russell Jones, *Learning Infrastructure as Code*, Packt Publishing, **2022**
R2: Steve Smith, *Infrastructure as Code Handbook*, Leanpub, **2023**
R3: James Turnbull, *The Terraform Book*, James Turnbull Publications, **2022**
R4: Gourav Shah, *Mastering Ansible*, Packt Publishing, **2022**

Web Resources
W1: https://www.terraform.io/docs
W2: https://docs.ansible.com
W3: https://learn.hashicorp.com
W4: https://www.redhat.com/en/technologies/management/ansible
W5: https://github.com/ansible/ansible-examples

Course Code: CDV3404
Course Title: Cloud Automation & Orchestration
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores the principles and tools used to automate and orchestrate cloud resources. It covers scripting for automation, orchestration tools, event-driven automation, and infrastructure lifecycle management. Students will gain practical exposure to tools like Terraform, Ansible, Kubernetes, and CI/CD systems that enable efficient, repeatable, and scalable cloud operations.

Course Objectives
— Understand automation and orchestration concepts in cloud environments
— Learn how to automate infrastructure and service deployment
— Explore orchestration tools like Kubernetes, Terraform, and Jenkins
— Apply automation and orchestration in DevOps and cloud-native workflows

Course Outcomes
CO1 (Understand): Explain the need and principles of cloud automation and orchestration
CO2 (Analyze): Evaluate automation workflows and orchestration strategies across tools
CO3 (Apply): Automate cloud infrastructure provisioning and application deployments
CO4 (Apply): Use orchestration tools to manage services in cloud-native environments

Course Content (45 Hours Total)

Module 1: Introduction to Cloud Automation – 11 Sessions (Understand)
Automation fundamentals, Infrastructure lifecycle, Scripting basics (Bash, Python), Cloud SDKs and CLIs, Infrastructure as Code (IaC) overview, Configuration management tools

Module 2: Orchestration Platforms and Strategies – 11 Sessions (Analyze)
Orchestration vs. automation, Kubernetes core components, Workload orchestration, Helm charts, Workflow engines (Argo Workflows, Jenkins Pipelines), Container orchestration use cases

Module 3: Infrastructure Automation Tools – 11 Sessions (Apply)
Terraform deep dive, Resource planning, Modules and state management, CloudFormation basics, Policy-as-Code, Ansible orchestration, CI/CD pipeline integration

Module 4: Cloud-native Automation and Monitoring – 12 Sessions (Apply)
Event-driven automation (CloudWatch Events, Azure Event Grid), Auto-scaling, Serverless automation, Secrets management (Vault), Monitoring and alerting (Prometheus, Grafana), Real-world case studies

Textbooks
T1: Steve Smith, *Infrastructure as Code: Dynamic Systems for the Cloud Age*, O'Reilly Media, **2023**
T2: Michael Hausenblas, *Cloud Native DevOps with Kubernetes*, O'Reilly Media, **2023**

Reference Books

R1: Yevgeniy Brikman, *Terraform: Up and Running*, O'Reilly, **2023**
R2: James Turnbull, *The Art of Monitoring*, James Turnbull Publications, **2022**
R3: Brent Laster, *Jenkins 2: Up and Running*, O'Reilly, **2022**
R4: Gourav Shah, *Mastering Ansible*, Packt Publishing, **2022**

Web Resources
W1: https://kubernetes.io/docs
W2: https://www.terraform.io
W3: https://docs.ansible.com
W4: https://argo-cd.readthedocs.io
W5: https://grafana.com/oss/prometheus

Course Code: CDV3405
Course Title: Cloud Infrastructure and Systems Software
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores the foundational components and systems software that support cloud computing infrastructure. Topics include virtualization, operating systems for cloud platforms, resource management, storage systems, networking, and containerization. It also introduces the role of hypervisors, system-level abstractions, and cloud resource orchestration.

Course Objectives
— Understand the architectural components of cloud infrastructure
— Learn how systems software enables virtualization and containerization
— Explore cloud storage, computing, and networking models
— Apply tools and techniques to manage virtualized environments

Course Outcomes
CO1 (Understand): Explain the architecture and components of cloud infrastructure
CO2 (Analyze): Evaluate the functionality of systems software in cloud environments
CO3 (Apply): Deploy and manage virtual machines and containers
CO4 (Apply): Configure system-level services for performance and scalability

Course Content (45 Hours Total)

Module 1: Cloud Infrastructure Overview – 11 Sessions (Understand)
Cloud architecture: compute, storage, network layers, Service and deployment models, Cloud datacenters, Resource pooling, SLA management, Multi-tenancy and elasticity

Module 2: Virtualization and Hypervisors – 11 Sessions (Analyze)
Hypervisor types (Type 1 and 2), VM lifecycle, Resource allocation, VirtualBox, KVM, VMware, Container vs VM, Container engines (Docker), Image layering

Module 3: Systems Software in Cloud – 11 Sessions (Apply)
Operating systems for virtualization, Kernel-level support, Filesystems and process management, Cloud-init, Remote management tools (SSH, Ansible), User and permission management

Module 4: Resource Management and Monitoring – 12 Sessions (Apply)
Cloud storage: block vs object vs file, Open-source storage (Ceph, GlusterFS), Cloud networking basics, IPAM, DNS, Load balancers, Monitoring tools (Nagios, Zabbix), Logs and metrics

Textbooks
T1: Pradeep Padala, *Cloud Infrastructure Technologies: A Hands-On Approach*, Wiley, **2023**
T2: James Bond, *Infrastructure as a Service Cloud Concepts*, BPB Publications, **2023**

Reference Books
R1: Rajkumar Buyya, *Cloud Computing Principles and Paradigms*, Wiley, **2022**
R2: Mark Carlson, *VMware vSphere 7: The Complete Guide*, Pearson, **2022**
R3: Kelsey Hightower et al., *Kubernetes: Up and Running*, O'Reilly, **2023**
R4: Tom Laszewski, *Cloud Native Infrastructure*, O'Reilly Media, **2022**

Web Resources
W1: https://docs.openstack.org
W2: https://www.vmware.com/technical-resources
W3: https://learn.microsoft.com/en-us/azure/architecture/
W4: https://www.redhat.com/en/technologies/linux-platforms/virtualization
W5: https://www.gluster.org

Course Code: CDV3406
Course Title: AI-Powered Chatbots & Automation in DevOps
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores the integration of AI-powered chatbots and automation tools in modern DevOps workflows. It covers the fundamentals of natural language understanding, chatbot frameworks, and AI-assisted automation for monitoring, deployment, and collaboration. Students will build chatbots and integrate them with tools like Slack, Jenkins, and incident response platforms to enhance DevOps efficiency.

Course Objectives
— Understand the role of conversational AI in DevOps and IT automation
— Learn chatbot architecture, NLP concepts, and development frameworks
— Apply automation for continuous integration, delivery, and operations
— Integrate AI assistants with DevOps tools for smart alerts and task execution

Course Outcomes
CO1 (Understand): Describe the role of chatbots and AI in DevOps environments
CO2 (Analyze): Evaluate chatbot design patterns and NLP techniques
CO3 (Apply): Develop conversational bots using open-source tools and APIs
CO4 (Apply): Automate DevOps tasks using AI assistants integrated with pipelines and monitoring tools

Course Content (45 Hours Total)

Module 1: Conversational AI Foundations – 11 Sessions (Understand)
Chatbot types (rule-based, AI-based), Natural Language Processing (NLP) basics,

Intent recognition, Entity extraction, Dialog management, Conversational UX, Overview of tools (Rasa, Dialogflow, Microsoft Bot Framework)

Module 2: Chatbot Architecture & Development – 11 Sessions (Analyze)
Bot architecture, Intent classification with ML, Slot filling, Fallback strategies, Integration with APIs and databases, Creating flows and actions, Context handling, Testing and analytics

Module 3: DevOps Integration with Chatbots – 11 Sessions (Apply)
Slack and Microsoft Teams bot integrations, Triggering Jenkins pipelines via chatbot, Monitoring tools integration (Prometheus, Grafana), Incident response automation, GitHub/GitLab bot interactions

Module 4: Intelligent Automation for DevOps – 12 Sessions (Apply)
AI-based alert triage, Anomaly detection and root cause analysis, ChatOps principles, Voicebot overview, Use of LLMs (ChatGPT) in DevOps workflows, Real-time use cases and mini project demos

Textbooks
T1: Navin Sabharwal, *Intelligent Automation with AI and Chatbots*, BPB Publications, **2023**
T2: Jason D. Brownlee, *Deep Learning for Natural Language Processing*, Machine Learning Mastery, **2023**

Reference Books
R1: Raj Ramesh, *AI and the Future of Automation*, Wiley, **2022**
R2: Munjal Desai, *Hands-On Chatbots and Conversational UI Development*, Packt, **2022**
R3: Dustin Coates, *Voice Applications for Alexa and Google Assistant*, Manning, **2021**
R4: Tom Taulli, *Artificial Intelligence Basics*, Apress, **2021**

Web Resources
W1: https://rasa.com
W2: https://cloud.google.com/dialogflow
W3: https://slack.dev/bolt-js/tutorial/getting-started
W4: https://learn.microsoft.com/en-us/azure/bot-service/
W5: https://chatops.readthedocs.io

Course Code: CDV3407
Course Title: AIOps – AI-Driven DevOps & IT Operations
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores **AIOps (Artificial Intelligence for IT Operations)**, combining big data and machine learning to automate and enhance IT operations. Students will learn how AIOps platforms process telemetry data, detect anomalies, predict outages, automate root cause analysis, and improve system performance across DevOps pipelines.

Course Objectives

— Understand the principles and architecture of AIOps platforms
— Explore how AI and ML are used to monitor, correlate, and automate IT operations
— Analyze telemetry data for alert prioritization, root cause detection, and predictions
— Apply AIOps tools for real-time automation in DevOps and enterprise environments

Course Outcomes
CO1 (Understand): Explain the fundamentals, components, and benefits of AIOps
CO2 (Analyze): Correlate events and alerts using ML models and behavior analysis
CO3 (Apply): Build anomaly detection and automation pipelines using AIOps tools
CO4 (Apply): Integrate AIOps with DevOps workflows for intelligent system management

Course Content (45 Hours Total)

Module 1: Introduction to AIOps – 11 Sessions (Understand)
Definition, evolution, and scope of AIOps, Role in DevOps and ITSM, AIOps architecture: data sources, analysis, and automation layers, Telemetry types (logs, metrics, traces), Observability vs Monitoring

Module 2: Data Collection and Event Correlation – 11 Sessions (Analyze)
Ingesting data from multiple sources (agents, APIs, cloud), Time-series analysis, Noise reduction, Pattern detection, Incident clustering, Correlation engines, Real-time event streams (Kafka, Fluentd)

Module 3: ML in AIOps – 11 Sessions (Apply)
Anomaly detection (statistical, supervised, unsupervised), Root cause analysis, Forecasting system health, Alert prioritization using NLP, AI models for log classification and enrichment

Module 4: AIOps in Practice – 12 Sessions (Apply)
AIOps tools (Dynatrace, Moogsoft, Splunk ITSI, Elastic, DataDog), Integration with CI/CD tools, Automated remediation and chatbot-based ops, Cloud-native AIOps, Case studies in fintech, healthcare, and SRE

Textbooks
T1: Pranay Ahlawat, *AIOps: Real-time Analytics for IT Operations*, Packt Publishing, **2023**
T2: Nilesh Barla, *Mastering AIOps*, BPB Publications, **2023**

Reference Books
R1: Mitch Ashley, *AIOps for Dummies*, Wiley, **2022**
R2: Ernest Lefner, *Site Reliability Engineering and AIOps*, Leanpub, **2022**
R3: Ankur Shah, *Enterprise AIOps with IBM Watson*, IBM Redbooks, **2021**
R4: Jason Bloomberg, *Data-Driven DevOps*, Apress, **2022**

Web Resources
W1: https://www.ibm.com/cloud/aiops
W2: https://www.dynatrace.com
W3: https://www.moogsoft.com
W4: https://www.elastic.co/observability
W5: https://azure.microsoft.com/en-us/solutions/aiops/

Course Code: CDV3408
Course Title: Predictive Analytics for System Monitoring & Performance Optimization
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course focuses on applying predictive analytics techniques to monitor, assess, and optimize system and application performance in IT environments. Students will explore the use of machine learning, statistical models, and real-time data to forecast system failures, reduce downtime, and ensure proactive operations.

Course Objectives
— Understand the fundamentals of predictive analytics in IT operations
— Learn to collect, preprocess, and analyze performance metrics
— Apply machine learning models for anomaly detection and forecasting
— Use visualization and automation tools for system performance optimization

Course Outcomes
CO1 (Understand): Explain the role of predictive analytics in system monitoring
CO2 (Analyze): Evaluate patterns and anomalies in performance metrics
CO3 (Apply): Implement predictive models to forecast system failures or overload
CO4 (Apply): Use monitoring tools and dashboards to support performance optimization

Course Content (45 Hours Total)

Module 1: Introduction to Predictive Analytics & Monitoring – 11 Sessions (Understand)
Concept of predictive analytics, Types of system metrics (CPU, memory, I/O, latency), Time-series analysis basics, Data sources and collection, Real-time vs batch monitoring, Monitoring frameworks overview (Prometheus, Grafana, Nagios)

Module 2: Data Preparation and Feature Engineering – 11 Sessions (Analyze)
Log and metric preprocessing, Feature extraction, Windowing and lag features, Outlier handling, Correlation analysis, Dimensionality reduction techniques (PCA, t-SNE)

Module 3: Predictive Modeling Techniques – 11 Sessions (Apply)
Supervised and unsupervised learning, Regression models (linear, ridge, lasso), Classification models (decision trees, random forests), Anomaly detection (Isolation Forest, DBSCAN), Forecasting (ARIMA, LSTM)

Module 4: System Optimization & Visualization – 12 Sessions (Apply)
Alert thresholds and automation, KPI dashboards with Grafana, Capacity planning and auto-scaling, Use of AI/ML in alert prioritization, Real-world case studies in cloud environments

Textbooks
T1: Usha Batra, *Predictive Analytics for IT Operations*, Wiley, **2023**
T2: Valliappa Lakshmanan, *Data Science on the Google Cloud Platform*, O'Reilly Media, **2023**

Reference Books

R1: Dean Abbott, *Applied Predictive Analytics*, Wiley, **2022**
R2: Michael Bowles, *Machine Learning in Action*, Manning, **2021**
R3: Soumendra Mohanty, *Big Data Imperatives*, Apress, **2022**
R4: Emily Freeman, *DevOps for Dummies*, Wiley, **2022**

Web Resources
W1: https://prometheus.io
W2: https://grafana.com
W3: https://scikit-learn.org
W4: https://www.tensorflow.org
W5: https://cloud.google.com/monitoring

Course Code: CDV3409
Course Title: AI-Enabled Cloud Cost Optimization
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course focuses on leveraging Artificial Intelligence (AI) and Machine Learning (ML) to optimize cloud infrastructure costs. It covers cloud pricing models, cost drivers, and resource management strategies using predictive analytics, anomaly detection, and intelligent scaling. Students will apply AI techniques to monitor, forecast, and automate cloud usage for efficiency and cost savings.

Course Objectives
— Understand cloud billing models, cost structures, and optimization challenges
— Learn to analyze and visualize usage patterns across cloud environments
— Apply AI/ML techniques for cost forecasting and anomaly detection
— Automate cost-saving strategies using cloud-native and third-party tools

Course Outcomes
CO1 (Understand): Explain cloud cost structures and factors affecting pricing
CO2 (Analyze): Examine usage data to identify inefficiencies and cost trends
CO3 (Apply): Use AI/ML models to forecast usage and automate optimization
CO4 (Apply): Implement tools and strategies for continuous cost governance

Course Content (45 Hours Total)

Module 1: Cloud Billing & Cost Management Basics – 11 Sessions (Understand)
Cloud pricing models (on-demand, reserved, spot), Service-level costing (compute, storage, bandwidth), Billing dashboards, Cloud provider cost calculators, Introduction to FinOps

Module 2: Cloud Usage Analytics & Visualization – 11 Sessions (Analyze)
Collecting and aggregating usage data, Data wrangling and transformation, Cost and usage reports, Budget alerting, Visualization tools (Grafana, AWS Cost Explorer, Azure Cost Management)

Module 3: Predictive Cost Optimization with AI – 11 Sessions (Apply)
ML for cost forecasting, Time-series models (ARIMA, Prophet), Workload prediction using regression/classification, Detecting outliers in billing, Auto-recommendations for rightsizing

Module 4: Automation and Governance – 12 Sessions (Apply)
Policy enforcement, Tag-based budgeting, Auto-scaling and scheduling, Cloud-native tools (AWS Compute Optimizer, Azure Advisor, GCP Recommender), Third-party platforms (CloudHealth, Spot.io), Real-world case studies

Textbooks
T1: J.R. Storment & Mike Fuller, *Cloud FinOps: Collaborative Cloud Financial Management*, O'Reilly Media, **2023**
T2: Aditya Garg, *Cloud Cost Optimization Handbook*, Packt Publishing, **2023**

Reference Books
R1: Ajay Dholakia, *Cloud Optimization for Dummies*, Wiley, **2022**
R2: Paul Zikopoulos, *AI for Cloud Operations*, IBM Redbooks, **2022**
R3: Rajdeep Dua, *Practical Google Cloud Platform*, Apress, **2022**
R4: Stephen Fleming, *The Art of Cost Optimization in AWS*, Independently Published, **2023**

Web Resources
W1: https://www.finops.org
W2: https://aws.amazon.com/aws-cost-management
W3: https://azure.microsoft.com/en-us/pricing
W4: https://cloud.google.com/billing/docs
W5: https://cloudhealth.vmware.com

Course Code: CDV3410
Course Title: AI-Powered Incident Management & Root Cause Analysis
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course introduces the integration of Artificial Intelligence in managing IT incidents and performing automated Root Cause Analysis (RCA). It explores predictive modeling, anomaly detection, log analysis, alert correlation, and intelligent remediation. The course emphasizes minimizing system downtime and improving operational efficiency through AI-driven solutions.

Course Objectives
— Understand the fundamentals of incident lifecycle and RCA in IT operations
— Explore AI techniques for event correlation and anomaly detection
— Apply machine learning models to automate incident identification and RCA
— Integrate AI with monitoring and alerting systems for proactive response

Course Outcomes
CO1 (Understand): Explain the incident management lifecycle and challenges in RCA
CO2 (Analyze): Detect anomalies and correlate events using AI models
CO3 (Apply): Use ML algorithms to automate RCA using logs and metrics
CO4 (Apply): Integrate AI-powered tools to improve incident response and remediation

Course Content (45 Hours Total)

Module 1: Incident Management Foundations – 11 Sessions (Understand)
Incident vs. problem vs. change management, Incident lifecycle, MTTR/MTBF/MTTI

metrics, Escalation models, SLA management, RCA methods (fishbone, 5 Whys, fault tree)

Module 2: Event Collection & Anomaly Detection – 11 Sessions (Analyze)
System telemetry: logs, metrics, traces, Feature extraction, Time-series anomaly detection, Unsupervised techniques (Isolation Forest, k-means), Event correlation with NLP, Noise reduction

Module 3: ML for Root Cause Analysis – 11 Sessions (Apply)
Log parsing and template mining, Feature encoding from logs, Classification/regression models for root cause prediction, Pattern recognition in incident sequences, RCA graphs, Alert deduplication

Module 4: Tools & Automation in Incident Resolution – 12 Sessions (Apply)
Integration with observability platforms (Prometheus, ELK Stack, Splunk), Incident automation platforms (PagerDuty, Moogsoft), ChatOps for incident response, Auto-remediation, Use cases and dashboards

Textbooks
T1: Nilesh Barla, *AI for IT Operations: Incident Detection to Root Cause Analysis*, Packt Publishing, **2023**
T2: Ramesh Srinivasan, *Machine Learning for IT Operations*, Apress, **2022**

Reference Books
R1: John Allspaw, *The Art of Capacity Planning and Incident Response*, O'Reilly, **2022**
R2: Gene Kim et al., *The Phoenix Project*, IT Revolution Press, **2023**
R3: Tarun Telang, *Effective Log Analysis with Machine Learning*, Manning, **2023**
R4: Arvind Chandaka, *Root Cause Analysis Handbook*, Elsevier, **2021**

Web Resources
W1: https://moogsoft.com/resources
W2: https://prometheus.io/docs
W3: https://www.elastic.co/guide
W4: https://www.pagerduty.com/resources
W5: https://docs.datadoghq.com

Course Code: CDV3411
Course Title: Reinforcement Learning for Self-Healing DevOps Pipelines
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course introduces the application of **Reinforcement Learning (RL)** techniques in building self-healing and autonomous DevOps pipelines. Students will learn foundational RL concepts and how to use them for intelligent automation, failure prediction, adaptive testing, and pipeline optimization. It emphasizes reliability, learning from feedback, and proactive system recovery.

Course Objectives
— Understand core concepts of Reinforcement Learning and its integration with DevOps
— Explore the design of reward systems and learning agents for automation
— Learn techniques for implementing self-healing actions in CI/CD environments
— Apply RL to build adaptive and resilient DevOps pipelines

Course Outcomes
CO1 (Understand): Explain the principles of Reinforcement Learning and its role in automation
CO2 (Analyze): Examine RL models for adaptive testing and pipeline optimization
CO3 (Apply): Implement RL agents for identifying and healing failures in CI/CD workflows
CO4 (Apply): Integrate RL with DevOps tools for real-time decision making and system recovery

Course Content (45 Hours Total)

Module 1: Reinforcement Learning Basics – 11 Sessions (Understand)
Agent-environment interaction, MDPs, Rewards and policies, Value and Q-functions, Exploration vs exploitation, Q-learning, SARSA, Deep Q-Networks (DQN)

Module 2: RL for DevOps Environments – 11 Sessions (Analyze)
DevOps lifecycle and failure points, Identifying reward signals in pipelines, State representation of CI/CD stages, Adaptive testing models, Log and metric feedback as input features

Module 3: Building Self-Healing Agents – 11 Sessions (Apply)
Designing and training RL agents, Environment simulation, Detecting build/test/deploy failures, Automated rollback and rerun policies, Integration with Jenkins, GitHub Actions, Kubernetes

Module 4: Real-World Use Cases and Optimization – 12 Sessions (Apply)
Pipeline performance tuning with RL, Scaling resource usage adaptively, Anomaly mitigation using RL, Real-time case studies (Netflix, Google SRE, GitLab), RL + AIOps hybrid architectures

Textbooks
T1: Richard S. Sutton & Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, **2nd Edition, 2023**
T2: Rahul Raj, *Reinforcement Learning for DevOps Automation*, Packt Publishing, **2023**

Reference Books
R1: Maxim Lapan, *Deep Reinforcement Learning Hands-On*, Packt Publishing, **2022**
R2: Daniel Whitenack, *Applied Machine Learning for DevOps Engineers*, O'Reilly Media, **2022**
R3: Ian Goodfellow et al., *Deep Learning*, MIT Press, **2022**
R4: Mark Smiley, *CI/CD for Machine Learning with Kubernetes*, O'Reilly, **2023**

Web Resources
W1: https://spinningup.openai.com
W2: https://keras.io/examples/rl/
W3: https://www.tensorflow.org/agents
W4: https://learn.microsoft.com/en-us/devops
W5: https://docs.github.com/en/actions

Course Code: CDV3412
Course Title: DevOps Fundamentals for Financial Services
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course provides an industry-oriented foundation in DevOps, tailored for the **financial services domain**. It covers key principles of DevOps, secure CI/CD pipelines, regulatory compliance, high-frequency deployment, and infrastructure automation in banking, insurance, and fintech environments. The course emphasizes scalability, resilience, and compliance-critical deployments.

Course Objectives
— Understand the DevOps lifecycle and its application in financial systems
— Explore automation, monitoring, and deployment practices tailored to finance
— Apply security and compliance practices in CI/CD pipelines
— Integrate DevOps with high-availability, auditability, and traceability requirements

Course Outcomes
CO1 (Understand): Describe the DevOps culture, principles, and practices for regulated industries
CO2 (Analyze): Evaluate DevOps toolchains, policies, and risk management in fintech
CO3 (Apply): Implement secure CI/CD pipelines with compliance enforcement
CO4 (Apply): Automate and monitor infrastructure and application delivery in financial services

Course Content (45 Hours Total)

Module 1: DevOps in Financial Services – 11 Sessions (Understand)
DevOps lifecycle and benefits, Financial sector IT requirements, Key challenges (latency, availability, auditability), Role of DevSecOps in finance, Compliance (PCI-DSS, SOC 2), Legacy system modernization

Module 2: CI/CD and Security Best Practices – 11 Sessions (Analyze)
Pipeline design in regulated environments, Secure builds, secrets management, Static and dynamic security testing, Approval workflows, Deployment gates, Artifact integrity verification

Module 3: Infrastructure Automation & Monitoring – 11 Sessions (Apply)
Infrastructure as Code (Terraform, Ansible), Cloud deployment (AWS, Azure for BFSI), Service monitoring (Prometheus, Grafana), Observability and alerting, Chaos engineering, Incident response

Module 4: Fintech Use Cases and Audit-Ready Pipelines – 12 Sessions (Apply)
DevOps in core banking and digital wallets, API testing and sandboxing, Audit trails and logging, Change management automation, FinOps alignment, Case studies from banking, insurance, and trading systems

Textbooks
T1: Emily Freeman, *DevOps for Dummies*, Wiley, **2023**
T2: Sricharan Vadapalli, *DevSecOps in Practice: A Guide to Secure CI/CD*, BPB Publications, **2023**

Reference Books
R1: Gene Kim et al., *The DevOps Handbook*, IT Revolution Press, **2022**
R2: Rob England, *Implementing DevOps in Financial Services*, ITSM Zone, **2021**
R3: Erdal Ozkaya, *Cybersecurity: The Beginner's Guide*, Packt Publishing, **2022**
R4: G. Venkatraman, *Mastering DevOps*, Packt Publishing, **2022**

Web Resources
W1: https://aws.amazon.com/financial-services/
W2: https://cloud.google.com/solutions/financial-services
W3: https://azure.microsoft.com/en-us/solutions/financial-services
W4: https://owasp.org/www-project-devsecops-guideline
W5: https://finops.org

Course Code: CDV3413
Course Title: Introduction to FinTech & Digital Banking
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course introduces the foundational concepts of **Financial Technology (FinTech)** and **Digital Banking**. It covers financial innovation, payment systems, blockchain in banking, digital lending, robo-advisory, regulatory frameworks, and emerging trends such as Open Banking and Embedded Finance. The course prepares learners for the rapidly evolving landscape of technology-driven financial services.

Course Objectives
— Understand the evolution of financial technology and digital transformation in banking
— Explore digital payment systems, lending platforms, and neobanking models
— Learn about key technologies driving FinTech (blockchain, AI, APIs)
— Examine regulations, cybersecurity, and ethics in FinTech ecosystems

Course Outcomes
CO1 (Understand): Describe the foundations of FinTech and key concepts in digital banking
CO2 (Analyze): Evaluate FinTech models, revenue strategies, and technology stacks
CO3 (Apply): Explore the application of APIs, mobile banking, and data-driven lending
CO4 (Apply): Examine real-world FinTech case studies, innovation, and regulatory compliance

Course Content (45 Hours Total)

Module 1: FinTech Fundamentals – 11 Sessions (Understand)
History of FinTech, FinTech vs traditional banking, Ecosystem and stakeholders, Types of FinTech (payments, lending, investments), Overview of neobanks and challenger banks

Module 2: Digital Payments & Lending – 11 Sessions (Analyze)
UPI, RTGS, SWIFT, wallets, BNPL models, Crowdfunding, P2P lending, Credit scoring using AI/ML, Embedded lending, Payment gateways, Fraud detection

Module 3: Technologies Behind FinTech – 11 Sessions (Apply)
Blockchain in banking and smart contracts, API-based banking and Open Banking (PSD2), Mobile-first development, Cloud adoption, Conversational banking with chatbots, Robo-advisory

Module 4: Regulations, Security & Future Trends – 12 Sessions (Apply)
Digital banking licenses, KYC/AML, GDPR and data privacy, Cybersecurity and threat intelligence, InsurTech and RegTech, FinTech in DeFi, ESG-driven FinTech innovation

Textbooks
T1: Susanne Chishti & Janos Barberis, *The FINTECH Book*, Wiley, **2023**
T2: Sanjay Phadke, *Fintech Future: The Digital DNA of Finance*, SAGE Publications, **2023**

Reference Books
R1: David Shrier, *Basic Blockchain: What It Is and How It Will Transform the Way We Work and Live*, Little Brown, **2022**
R2: Bernardo Nicoletti, *The Future of FinTech: Integrating Finance and Technology in Financial Services*, Palgrave Macmillan, **2022**
R3: Chris Skinner, *Digital Bank: Strategies to Launch or Become a Digital Bank*, Marshall Cavendish, **2021**
R4: Arvind Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press, **2022**

Course Code: CDV3414
Course Title: Microservices Architecture for Banking Applications
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course introduces the principles of **microservices architecture** and its application in building scalable, secure, and modular **banking systems**. It focuses on service decomposition, inter-service communication, security, API gateways, containerization, and deployment strategies specific to financial systems with high availability and regulatory compliance.

Course Objectives
— Understand microservices architecture and its relevance in banking systems
— Explore design patterns, communication protocols, and deployment strategies
— Learn containerization and orchestration in building cloud-native microservices
— Apply secure and scalable microservice practices in real-world fintech applications

Course Outcomes
CO1 (Understand): Describe the architecture and benefits of microservices in financial services
CO2 (Analyze): Evaluate service decomposition, inter-service communication, and database strategies
CO3 (Apply): Build and deploy microservices using containers and orchestration tools
CO4 (Apply): Implement security, observability, and fault tolerance for banking microservices

Course Content (45 Hours Total)

Module 1: Microservices Architecture Fundamentals – 11 Sessions (Understand)
Monolith vs microservices, Banking system case study, Domain-driven design

(DDD), Service decomposition strategies, RESTful services, Event-driven architecture basics

Module 2: Communication & Database Design – 11 Sessions (Analyze)
Synchronous vs asynchronous communication (REST, gRPC, messaging), Service discovery, API gateway patterns, Saga and CQRS patterns, Polyglot persistence, Data consistency challenges

Module 3: Containerization & Deployment – 11 Sessions (Apply)
Dockerizing banking microservices, Kubernetes for orchestration, Service mesh (Istio/Linkerd), CI/CD pipelines for microservices, Canary and blue-green deployments, Versioning and rollback

Module 4: Security, Compliance & Observability – 12 Sessions (Apply)
OAuth2, JWT, API security, Threat modeling in banking, Centralized logging and tracing (ELK, Jaeger), Health checks, Circuit breakers, SLA/SLO monitoring, Regulatory compliance (PCI DSS, RBI norms)

Textbooks
T1: Sourabh Sharma, *Mastering Microservices with Java*, Packt Publishing, **2023**
T2: Morgan Bruce & Paulo A. Pereira, *Microservices: Flexible Software Architecture*, O'Reilly Media, **2023**

Reference Books
R1: Sam Newman, *Building Microservices*, O'Reilly, **2022**
R2: Chris Richardson, *Microservices Patterns*, Manning Publications, **2022**
R3: Rajesh RV, *Spring Microservices in Action*, Manning, **2022**
R4: Susan Fowler, *Production-Ready Microservices*, O'Reilly, **2021**

Web Resources
W1: https://microservices.io
W2: https://spring.io/guides
W3: https://kubernetes.io
W4: https://istio.io/latest/docs
W5: https://12factor.net

Course Code: CDV3415
Course Title: Automated Testing & Monitoring in FinTech
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course focuses on automated testing and real-time monitoring practices in **FinTech environments**. It introduces testing strategies, continuous testing pipelines, observability principles, performance testing, and monitoring tools critical for ensuring reliability, compliance, and user trust in financial applications.

Course Objectives
— Understand the role of automated testing and monitoring in financial systems
— Learn about test frameworks, tools, and CI/CD integration in FinTech
— Explore system observability, metrics collection, and alerting techniques
— Apply test automation and monitoring strategies for compliance and audit readiness

Course Outcomes

CO1 (Understand): Describe testing types, observability concepts, and their role in FinTech platforms
CO2 (Analyze): Evaluate test automation frameworks and monitoring architectures
CO3 (Apply): Implement functional, security, and performance testing for digital banking systems
CO4 (Apply): Use monitoring tools and dashboards for real-time health checks and compliance reporting

Course Content (45 Hours Total)

Module 1: Testing in FinTech Applications – 11 Sessions (Understand)
Functional testing (unit, integration, system), Non-functional testing (security, performance), Compliance-driven testing, Regression testing, Test strategy for financial systems

Module 2: Automation Frameworks & CI/CD Integration – 11 Sessions (Analyze)
Selenium, TestNG, Postman, JMeter for APIs and load testing, BDD (Cucumber), CI tools (Jenkins, GitHub Actions), Test orchestration in cloud-native pipelines

Module 3: Observability & Monitoring – 11 Sessions (Apply)
Metrics, logs, traces overview, OpenTelemetry, Prometheus & Grafana, Distributed tracing (Jaeger), Alerting strategies, Uptime monitoring, SLA & SLO validation

Module 4: Secure, Scalable, and Auditable Test Environments – 12 Sessions (Apply)
Test data management, Masking and anonymization, Monitoring security events, Audit trails, Incident simulation, Sandbox environments, Case studies from FinTech APIs, wallets, and neobanks

Textbooks
T1: Rex Black et al., *Foundations of Software Testing: ISTQB Certification*, Cengage, **2023**
T2: Chris Riley, *DevOps Monitoring and Performance Testing*, O'Reilly, **2023**

Reference Books
R1: Lisa Crispin & Janet Gregory, *Agile Testing Condensed*, Addison-Wesley, **2022**
R2: Narayan Sashikanth, *Practical API Testing with Postman*, Packt Publishing, **2023**
R3: Neil Bartlett, *Continuous Testing for DevOps Professionals*, DZone Refcardz, **2022**
R4: Brendan Gregg, *Systems Performance: Monitoring and Tools*, Pearson, **2022**

Web Resources
W1: https://jmeter.apache.org
W2: https://www.selenium.dev
W3: https://opentelemetry.io
W4: https://grafana.com
W5: https://cucumber.io/docs/guides/10-minute-tutorial/

Course Code: CDV3416
Course Title: DevSecOps in Financial Services
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores the implementation of **DevSecOps** practices in the **highly regulated financial services sector**. It focuses on integrating security into DevOps pipelines, compliance automation, secure coding, threat modeling, and governance frameworks essential for building secure, compliant, and resilient financial applications.

Course Objectives
— Understand the role of DevSecOps in financial application development and deployment
— Learn to integrate security controls and compliance policies into CI/CD workflows
— Explore tools and techniques for vulnerability detection and mitigation
— Apply DevSecOps principles to meet audit, regulatory, and operational requirements

Course Outcomes
CO1 (Understand): Describe the DevSecOps lifecycle and security needs of financial applications
CO2 (Analyze): Evaluate security threats and compliance risks in DevOps pipelines
CO3 (Apply): Implement automated security testing and compliance checks
CO4 (Apply): Integrate policy-as-code and governance in financial DevOps systems

Course Content (45 Hours Total)

Module 1: DevSecOps Foundations in Finance – 11 Sessions (Understand)
Introduction to DevSecOps, Financial system threat landscape, Secure SDLC, Shared responsibility model, Security and compliance requirements (PCI DSS, SOX, GDPR, RBI guidelines)

Module 2: Secure CI/CD Pipelines – 11 Sessions (Analyze)
Secure coding practices, Static and dynamic analysis (SAST, DAST), Secret management (Vault, GitGuardian), Container security (image scanning), Supply chain security, SBOM (Software Bill of Materials)

Module 3: Automation and Policy-as-Code – 11 Sessions (Apply)
Infrastructure as Code security (Checkov, tfsec), Policy enforcement (OPA, Sentinel), IAM automation, Audit logging, SIEM integration, Compliance-as-code tools and frameworks

Module 4: DevSecOps Tools & Governance – 12 Sessions (Apply)
Security tools: SonarQube, Aqua, Snyk, Trivy, Real-time threat detection, DevSecOps maturity models, Governance frameworks, Case studies from banking and fintech (risk scoring, audit readiness)

Textbooks
T1: Jim Bird, *DevSecOps: A leader's guide to producing secure software without compromising flow*, O'Reilly, **2023**
T2: Glenn Wilson, *DevSecOps: A Practical Guide*, Apress, **2023**

Reference Books
R1: Mark Miller et al., *The DevSecOps Playbook*, IT Revolution, **2022**
R2: Tony Hsiang-Chih Hsu, *Security Automation with Ansible 2*, Packt, **2022**
R3: Erdal Ozkaya, *Cybersecurity: The Beginner's Guide*, BPB Publications, **2022**
R4: Rajesh Gupta, *Practical Security for Agile and DevOps*, Springer, **2021**

Web Resources
W1: https://owasp.org/www-project-devsecops-guideline
W2: https://snyk.io
W3: https://checkov.io
W4: https://www.hashicorp.com/blog/secure-infrastructure
W5: https://devsecops.org

Course Code: CDV3417
Course Title: Blockchain & Smart Contracts for Financial Transactions
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores the **application of blockchain technology and smart contracts** in modern **financial ecosystems**. It covers decentralized ledgers, consensus mechanisms, digital tokens, and the development of smart contracts using platforms like Ethereum and Hyperledger. The focus is on secure, transparent, and automated financial transactions.

Course Objectives
— Understand blockchain fundamentals and their relevance to the financial domain
— Learn about cryptocurrency, tokens, and smart contracts for finance
— Develop and test smart contracts using industry-standard platforms
— Analyze blockchain-based financial applications and their regulatory implications

Course Outcomes
CO1 (Understand): Explain blockchain architecture, transaction flow, and consensus in financial systems
CO2 (Analyze): Evaluate the benefits, risks, and use cases of smart contracts in financial services
CO3 (Apply): Develop and deploy smart contracts for secure and automated financial transactions
CO4 (Apply): Assess blockchain financial applications considering legal and compliance frameworks

Course Content (45 Hours Total)

Module 1: Blockchain Fundamentals & Financial Systems – 11 Sessions (Understand)
Distributed ledger concepts, Types of blockchain (public, private, consortium), Consensus algorithms (PoW, PoS, PBFT), Blockchain components, Cryptographic hashing, Blockchain in banking, insurance, and cross-border payments

Module 2: Smart Contracts for Finance – 11 Sessions (Analyze)
Smart contract principles and lifecycle, Languages (Solidity, Vyper), Ethereum Virtual Machine (EVM), Events and modifiers, Risks and attack vectors, Use cases: escrow, tokenized lending, KYC automation

Module 3: Development & Deployment – 11 Sessions (Apply)
Solidity programming basics, Smart contract structure, Remix IDE, Truffle and Ganache, Blockchain testnets, Transaction gas and optimization, Oracles and external data integration

Module 4: Applications & Regulation – 12 Sessions (Apply)
Decentralized Finance (DeFi), NFTs in finance, CBDCs, Hyperledger Fabric and Corda in enterprise banking, AML/KYC with blockchain, Smart contract auditing, Legal & compliance considerations (RBI, SEC, GDPR)

Textbooks
T1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **4th Edition, 2023**
T2: Andreas M. Antonopoulos & Gavin Wood, *Mastering Ethereum*, O'Reilly Media, **2022**

Reference Books
R1: Arvind Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press, **2022**
R2: Melanie Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly, **2021**
R3: Salman Baset, *Blockchain for Financial Institutions*, IBM Press, **2022**
R4: Xinxin Fan & Guang Gong, *Blockchain for Cybersecurity and Privacy*, CRC Press, **2022**

Web Resources
W1: https://ethereum.org/en/developers
W2: https://soliditylang.org
W3: https://remix.ethereum.org
W4: https://hyperledger.org
W5: https://trufflesuite.com

Course Code: CSE2514	**Course Title: Operating Systems** Lab **Type of Course: Lab Only**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	CSE2009- Computer Organization Students should have basic knowledge on computers, computer software & hardware, and Computer Organization. Prior programming experience in C is recommended.					
Anti-requisites	**NIL**					
Course Description	This laboratory course provides hands-on experience with the core concepts of operating systems through practical assignments, simulations, and case studies. It covers foundational aspects such as system calls, process and thread management, inter-process communication, synchronization, deadlocks, memory management, and file systems. Students will implement and simulate real-time OS components and scheduling algorithms, fostering deeper understanding of OS architecture and					

	design. The lab also introduces modern OS tools, programming interfaces, and the basics of open-source OS environments.			
Course Object	The objective of the course is to familiarize the learners with the concepts of Operating Systems and attain **Employability** through **Problem Solving** Methodologies.			
Course Out Comes	On successful completion of the course the students shall be able to: 1] Demonstrate system-level programming using system calls and OS structures. [**Apply**] 2] Simulate process scheduling and multithreading techniques. [**Apply**] 3] Apply various tools to handle synchronization problems using semaphores and shared memory. [**Apply**] 4] Demonstrate memory management and file system concepts using simulation or scripting. **[Apply]**			
Course Content:				
Module 1	Introduction to Operating System	Assignment	Programming	**9 Hours**
Topics: Introduction to OS , Operating-System Operations, Operating System Services, , System Calls and its types, Operating System Structure, System Program and its types, Linkers and Loaders, Overview of OS design and implementation, Open-source operating system				
Module 2	Process Management	Assignment/Case Study	Programming/Simulation	**11 Hours**
Topics: Process Concept, Operations on Processes, Inter Process Communication, Communication in client-server systems (sockets, RPC, Pipes), Introduction to threads - Multithreading Models, Thread Libraries, Threading Issues, Process Scheduling– Basic concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF, RR and Priority.				
Module 3	Process Synchronization and Deadlocks	Assignment	Programming	**11 Hours**
Topics: The Critical-Section Problem- Peterson's Solution, Synchronization hardware, Semaphores, Classic Problems of Synchronization with Semaphore Solution- Producer-Consumer Problem, Reader-Writer problems, Dining Philosopher's Problem, . Introduction to Deadlocks, Necessary conditions for deadlock, Resource allocation Graph, Methods for handling deadlock: Deadlock Prevention and				

Implementation, Deadlock Avoidance and Implementation, Deadlock detection & Recovery from Deadlock.

Module 4	Memory Management	Assignment	Programming/Simulation	**10 Hours**

Topics:

Introduction to Memory Management, Basic hardware-Base and Limit Registers, Memory Management Unit(MMU), Dynamic loading and linking, Swapping, Contiguous and Non-Contiguous Memory Allocation, Segmentation, Paging - Structure of the Page Table – Virtual Memory and Demand Paging – Page Faults and Page Replacement Algorithms, Copy-on-write, Allocation of Frames, Thrashing

Introduction to File system management: File System Interface (access methods, directory structures), File system implementation.

Targeted Application:

Application area is traffic management system, banking system, health care and many more systems where in there are resources and entities that use and manage the resources.

Software Tools:

Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]. Used to install and work on multiple guest Operating Systems on top of a host OS.

Intel Processor identification utility: This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.

List of Laboratory Tasks:

Lab sheet -1

L1: Write a program to demonstrate the use of fork() and exec() system calls in process creation.

L2: A system has limited memory and high-priority real-time processes. Design a scheduling algorithm that ensures responsiveness while preventing starvation.

Lab sheet -2

L1: Implement First-Come-First-Serve (FCFS) process scheduling using C or Python.

L2: You are designing a server that handles thousands of client connections. Compare multithreading and multiprocessing for this task and implement a basic server model.

Lab sheet -3

L1: Implement Round Robin Scheduling with a fixed time quantum.

L2: In a banking system, concurrent access to accounts leads to data corruption. Design a synchronization solution to avoid race conditions.

Lab sheet -4

L1: Write a program to create threads using Pthreads or Python's threading module.

L2: You're tasked with building a file access tracker in an OS. Implement a system to log file access patterns and identify frequent accesses.

Lab sheet -5

L1: Demonstrate inter-process communication (IPC) using pipes.

L2: A simulation tool needs to emulate process suspension and resumption. Design and implement such a mechanism using signals or condition variables.

Lab sheet -6

L1: Simulate the Producer-Consumer problem using semaphores.

L2: You're developing a system where sensor devices (producers) generate temperature readings, and data processors (consumers) store and process these readings. To prevent race conditions and ensure buffer safety, implement a synchronization mechanism using semaphores.

Lab sheet -7

L1: Implement Dining Philosophers Problem using threads and synchronization.

L2: In a multi-threaded cafeteria simulation, five philosophers sit around a circular table, each alternating between thinking and eating. To eat, a philosopher must hold two forks (represented by shared resources). Your task is to avoid deadlock and ensure no philosopher starves using thread synchronization techniques.

Lab sheet -8

L1: Write a program to simulate First Fit, Best Fit, and Worst Fit memory allocation strategies.

L2: A system with limited memory blocks needs to allocate memory to processes arriving with various size requests. Your task is to implement three classic memory allocation strategies—First Fit, Best Fit, and Worst Fit—to allocate memory to each process efficiently. Simulate and compare how memory gets allocated in each strateg

Lab sheet -9

L1: Demonstrate paging using a simple page table simulation.

L2: A program has a logical address space divided into pages. The system's memory is divided into equal-sized frames. When a program executes, its pages are loaded into available frames in main memory. Simulate the address translation process using a page table and demonstrate how a logical address is converted to a physical address.

Lab sheet -10

L1: Write a program to simulate page replacement algorithms like FIFO and LRU.

L2: In a virtual memory system, a process accesses pages in a specific order. The memory can only hold a limited number of pages (frames). When a page is needed and the memory is full, a page replacement algorithm is used to decide which page to evict. Simulate and compare FIFO and LRU algorithms for a given page reference string.

Lab sheet -11

L1: Simulate file directory structure (single level/two level).

L2: A university campus computer lab has limited memory space available for each student login session. When students open files or run programs, memory pages are loaded into available memory frames. Due to the limited number of frames, some pages must be replaced when new ones are needed. The lab system uses page replacement algorithms to decide which pages to evict when memory is full..

Lab sheet -12

L1: Write a shell script to demonstrate file handling commands in Linux.

L2: Design a command-line mini shell that can run background and foreground processes and handle basic built-in commands like cd, pwd, exit.

Project work/Assignment

Demonstrate process concepts in LINUX OS.

Simulation of CPU scheduling algorithms.

Develop program to demonstrate use of Semaphores in threads.

Develop program to demonstrate use of deadlock avoidance algorithms.

Develop program to demonstrate use of page replacement algorithms.

Simulation of memory allocation strategies [first fit, best fit and worst fit].

Text Book

Silberschatz A, Galvin P B and Gagne G , “Silberschatz's Operating System Concepts”, Paperback, Global Edition Wiley, 2019

References

Silberschatz A, Galvin P B and Gagne G, “Operating System Concepts”, 10th edition Wiley, 2018.

William Stallings, “Operating Systems”,Ninth Edition, By Pearson Paperback ,1 March 2018.

Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, “ Cracking the Operating System skills”, Dreamtech, paperback, 2020

Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , “Operating Systems: Three Easy Pieces, Amazon digital Services”, September 2018.

E-resources/Weblinks

https://www.os-book.com/OS9/

https://pages.cs.wisc.edu/~remzi/OSTEP/

https://codex.cs.yale.edu/avi/os-book/OS10/index.html

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

CSE1500</td><td colspan="3">Course Title: Computational Thinking Using Python

Type of Course: Integrated</td><td>L- T- P- C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course introduces students to the essential skills of computational thinking and their practical application through the Python programming language. By combining problem-solving strategies with coding, students will learn to decompose complex challenges, identify patterns, abstract general principles, and design algorithms to build functional programs</td></tr>
<tr><td>Course Objective</td><td colspan="8">The objective of the course is to familiarize the learners with the concepts of Computational Thinking and use the Computational Thinking Principles to solve the computational Problems using Python Language</td></tr>
<tr><td>Course Outcomes</td><td colspan="8">Upon successful completion of this course, students will be able to:
• Explain and apply the core principles of computational thinking:
o Decomposition
o Pattern Recognition
o Abstraction
o Algorithm Design
• Use Python to implement solutions to real-world problems.
• Write and debug Python code using functions, loops and conditions
• Design simple programs and algorithms to automate repetitive or complex tasks.
• Collaborate effectively and communicate problem-solving approaches using pseudocode and Python.</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1</td><td colspan="3">Pillars of Computational Thinking</td><td colspan="2">Comprehension</td><td colspan="2"></td><td>9 Sessions</td></tr>
<tr><td colspan="9">What is computational thinking? Why is it important? Pillars of computational thinking: decomposition; pattern recognition; data representation and abstraction; algorithms

Applying computational thinking to case studies</td></tr>
</table>

Module 2	**Algorithm Design & Problem-Solving Strategies**	Application		**9 Sessions**
Introduction to Algorithms, Introduction to Problem Solving techniques: Brute Force, Divide and conquer, Common algorithms: find-max, linear search, binary search and other simple Algorithms				
Module 3	**Applied Computational Thinking using Python**	Application		**12 Sessions**
Introduction to Python, Data representation: variables, lists, Conditionals, Loops and Iteration Basic Example programs to illustrate the programming constructs				
Targeted Application & Tools that can be used: **Google Colab, Python**				
Text Book 1. "Computational Thinking for the Modern Problem Solver" – David D. Riley & Kenny A. Hunt 2. "Mastering Python 3 Programming: Ultimate Guide to Learn Python Coding Fundamentals and Real-World Applications" Subburaj Ramaswamy, BPB publications				
References 1. • **Sweigart, Al.** *Automate the Boring Stuff with Python: Practical Programming for Total Beginners.* No Starch Press, 2015. https://automatetheboringstuff.com • **Severance, Charles.** *Python for Everybody: Exploring Data Using Python 3.* CreateSpace Independent Publishing, 2016. https://www.py4e.com • **Wing, Jeannette M.** "Computational Thinking." *Communications of the ACM*, vol. 49, no. 3, 2006, pp. 33–35. https://doi.org/10.1145/1118178.1118215 • **Downey, Allen B.** *Think Python: How to Think Like a Computer Scientist.*				

Green Tea Press, 2015.
http://greenteapress.com/wp/think-python-2e/

.

E-Resources

https://edu.google.com/resources/programs/exploring-computational-thinking

Topics relevant to "SKILL DEVELOPMENT": Decomposition, Abstraction, Pattern recognition, Data Representation ,Algorithms

Ittagalpura, Rajanukunte, Yelahanka, Bengaluru 560 119