



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 ARTIFICIAL INTELLIGENCE & ADVANCED COMPUTING

Program Regulations and Curriculum 2026-2030

BACHELOR OF TECHNOLOGY (B.Tech) DEVOPS (CDV)

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

Table of Contents

Clause No.	Contents	Page Number
PART A – PROGRAM REGULATIONS		
1.	Vision & Mission of the University and the School / Department	4
2.	Preamble to the Program Regulations and Curriculum	4
3.	Short Title and Applicability	5
4.	Definitions	5
5.	Program Description	7
6.	Minimum and Maximum Duration	8
7.	Programme Educational Objectives (PEO)	8
8.	Programme Outcomes (PO) and Programme Specific Outcomes (PSO)	9-10
9.	Admission Criteria (as per the concerned Statutory Body)	10
10.	Lateral Entry / Transfer Students requirements	11
11.	Change of Branch / Discipline / Specialization	13
12.	Specific Regulations regarding Assessment and Evaluation	14
13.	Additional clarifications - Rules and Guidelines for Transfer of Credits from MOOC, etc.	17
PART B: PROGRAM STRUCTURE		
14.	Structure / Component with Credit Requirements Course Baskets & Minimum Basket wise Credit Requirements	20
15.	Minimum Total Credit Requirements of Award of Degree	20
16.	Other Specific Requirements for Award of Degree, if any, as prescribed by the Statutory Bodies	21
PART C: CURRICULUM STRUCTURE		
17.	Curriculum Structure – Basket Wise Course List	21
18.	Practical / Skill based Courses – Internships / Thesis / Dissertation / Capstone Project Work / Portfolio / Mini project	24
19.	List of Elective Courses under various Specializations / Stream Basket	26

20.	List of Open Electives to be offered by the School / Department (Separately for ODD and EVEN Semesters).	31
21.	List of MOOC (NPTEL) Courses	36
22.	Recommended Semester Wise Course Structure / Flow including the Program / Discipline Elective Paths / Options	36
23.	Course Catalogue of all Courses Listed including the Courses Offered by other School / Department and Discipline / Program Electives	41

PART A – PROGRAM REGULATIONS

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

1.1 Vision of the University

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

1.2 Mission of the University

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

1.3 Vision of Presidency School of Computer Science and Engineering

To be a value-based, practice-driven 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 2025-2029.
- b. These Regulations are subject to, and pursuant to the Academic Regulations.
- c. These Regulations shall be applicable to the ongoing Bachelor of Technology Degree Programs of the 2025-2029 batch, and to all other Bachelor of Technology Degree Programs which may be introduced in future.
- d. These Regulations shall supersede all the earlier Bachelor of Technology Degree Program Regulations and Curriculum, along with all the amendments thereto.
- e. These Regulations shall come into force from the Academic Year 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, 2025-2029;*
- 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 2025-2029 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 2025-2029 offered by the Presidency School of Computer Science and Engineering (PSCS):

1. Bachelor of Technology in Computer Science and Engineering, abbreviated as CSE
2. Bachelor of Technology in Computer Science and Technology (Big Data), abbreviated as CBD
3. Bachelor of Technology in Computer Science and Engineering (Block Chain), abbreviated as CBC
4. Bachelor of Technology in Computer Science and Technology (Dev Ops), abbreviated as CDV
5. Bachelor of Technology in Computer Science and Engineering (Cyber Security), abbreviated as CCS
6. Bachelor of Technology in Computer Science and Engineering (Internet of Things), abbreviated as CIT
7. Bachelor of Technology in Computer Science and Engineering (Data Science), abbreviated as CSD
8. Bachelor of Technology in Computer Science and Technology, abbreviated as CSG
9. Bachelor of Technology in Information Science and Technology, abbreviated as IST
10. Bachelor of Technology in Computer Science and Information Technology, abbreviated as CSI
11. Bachelor of Technology in Computer Science and Engineering (Networks), abbreviated as CSN
12. Bachelor of Technology in Computer Engineering, abbreviated as COM
13. Bachelor of Technology in Information Science and Engineering, abbreviated as ISE and
14. Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) abbreviated as CAI

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:

P01: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization to develop to the solution of complex engineering problems.

P02: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.

P03: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.

P04: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions.

P05: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.

P06: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

P07: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

P08: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

P09: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

P010: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

P011: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.

8.2 Program Specific Outcomes (PSOs):

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

PSO1: Leverage a strong foundation in computer science fundamentals to explain key concepts, analyze computational problems with theoretical knowledge, and design reliable and optimized computing systems.

PSO2: Make effective use of coding skills, software development knowledge, and understanding of big data systems to design and implement real-world applications that prepare individuals for different roles in software career, further studies, research work, or launching startups.

PSO3: Demonstrate collaborative skills, apply agile DevOps workflows, software engineering and automation principles to implement continuous integration (CI), continuous delivery (CD), and version control using modern DevOps tools and practices.

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, 2025-2029, 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**

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	

*CSS3150-Front End Full stack development

CSS3151-Java Full Stack Development

CSS3152-.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 student must obtain a minimum of 30% of the total marks/weightage assigned to the End Term Examinations in the concerned Course.
- 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.

- 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 be 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 (2025-2029) 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) 2025-2029: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets		
Sl. No.	Baskets	Credit Contribution
1	Humanities and Social Sciences including Management Courses (HSMC)	10
2	Basic Science Courses (BSC)	24
3	Engineering Science Courses (ESC)	22
4	Professional Core Courses (PCC)	64
5	Professional Elective Courses (PEC)	18
6	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:
- Fulfilled the Minimum Credit Requirements and the Minimum Credits requirements under various baskets;
 - 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;
 - No dues to the University, Departments, Hostels, Library, and any other such Centers/ Departments of the University; and
 - No disciplinary action is pending against her/him.

17. Curriculum Structure – Basket Wise Course List (not Semester Wise)

List of Courses Tabled – aligned to the Program Structure

(Course Code, Course Name, Credit Structure (LTPC), Contact Hours, Course Basket, Type of Skills etc., as applicable).

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

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

Table 3.1 : List of Humanities and Social Sciences including Management Courses (HSMC)									
S.No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of skill	Pre Requisite
1	ENG1030	Fundamentals of Communication	1	0	2	2	3	S	
2	ECE1512	Design Thinking in Practice	2	0	2	3	4	F	
3	ENG2030	Communication in Practice	1	0	2	2	3	S	ENG1002
4	PPS3018	Preparedness for interview	0	0	2	1	2	S	

5	APT4005	Aptitude for Employability	0	0	2	1	2	S	
6	FIN1002	Essentials of Finance	3	0	0	3	3	S	
Total No. of Credits						12			

Table 3.2 : List of Basic Science Courses (BSC)

S.No	Course Code	Course Name	L	T	P	C	Conact Hours	Type of Skill	Pre-requisite
1	MAT1211	Linear Algebra, Differential Equations and Statistical Methods	3	1	0	4	4	F	-
2	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F	-
3	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F	-
4	MAT1212	Mathematical foundations for Machine Learning and Data Analysis	3	1	0	4	4	F	-
5	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S	-
6	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S	-
7	MAT2311	Vector Calculus, Numerical Methods and Sampling Distributions	3	1	0	4	4	EM	-
8	MAT2321	Discrete Mathematical Structures	3	1	0	4	4	F	-
Total No. of Credits						24			

Table 3.3 : List of Engineering Science Courses (ESC)

S.No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Pre-requisite
1	MEC1006	Engineering Graphics	2	0	0	2	2	S	-
2	CSS1500	Computational Thinking using Python	2	0	2	3	4	S	-

3	ECE2022	Digital Design	2	0	0	2	2	F/S	-
4	ECE2052	Digital Design Lab	0	0	2	1	2	F/S	-
5	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S	-
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S	-
7	EEE1251	Basics of Electrical and Electronics Engineering Laboratory	0	0	2	1	2	F/S	-
8	CSS2264	Essentials of AI	3	0	0	3	3	S/EM	-
9	CSS2265	Essentials of AI Lab	0	0	2	1	2	S/EM	-
10	CSS2274	Competitive Programming and Problem Solving	0	0	4	2	4	S/EM	-
Total No. of Credits						20			

Table 3.4 : List of Professional Core Courses (PCC)

S.No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Pre-requisite
1	CSS2500	Problem Solving using C	2	0	0	2	2	S	Nil
2	CSS2501	Problem Solving using C Lab	0	0	4	2	4	S	Nil
3	CSS2251	Data Communication and Computer Networks	3	0	0	3	3	S	Nil
4	CSS2252	Data Communication and Computer Networks Lab	0	0	2	1	2	S	Nil
5	CSS2253	Data Structures	3	0	0	3	3	S	Nil
6	CSS2254	Data Structures Lab	0	0	2	1	2	S	Nil
7	CSS2255	Object Oriented Programming Using Java	3	0	0	3	3	S/EM	Nil
8	CSS2256	Object Oriented Programming Using Java Lab	0	0	2	1	2	S/EM	Nil
9	CSS2257	Computer Organization and Architecture	3	0	0	3	3	S	Nil
10	CDV2000	DevOps Foundations	3	0	0	3	3	S/EM	Nil
11	CSS2271	Software Design and Development	3	0	0	3	3	S	Nil

12	CSS2258	Web Technologies	3	0	0	3	3	S/EM	Nil
13	CSS2259	Web Technologies Lab	0	0	2	1	2	S/EM	Nil
14	CSS2260	Database Management Systems	3	0	0	3	3	S	Nil
15	CSS2261	Database Management Systems Lab	0	0	2	1	2	S	Nil
16	CDV2501	DevOps Tools Internals	3	0	0	3	3	S	CDV2000
17	CDV2502	DevOps Tools Internals Lab	0	0	2	1	2	S	CDV2000
18	CSS2262	Analysis of Algorithms	3	1	0	4	4	S	Nil
19	CSS2263	Analysis of Algorithms Lab	0	0	2	1	2	S	Nil
20	CSS2266	Theory of Computation	3	0	0	3	3	S	Nil
21	CDV2001	Agile Structures and Frameworks	3	0	0	3	3	S	Nil
22	CSS2269	Operating Systems	3	0	0	3	3	S	Nil
23	CSS2270	Operating Systems Lab	0	0	2	1	2	S/EM	Nil
24	CDV2503	Software Testing	3	0	0	3	3	S/EM	CDV2000
25	CDV2504	Software Testing Lab	0	0	2	1	2	S/EM	CDV2000
26	CDV2508	Continuous Integration and Continuous Delivery (CI/CD) Pipelines	3	0	0	3	3	S/EM	CDV2000
27	CDV2512	Continuous Integration and Continuous Delivery (CI/CD) Pipelines Lab	0	0	4	2	4	S/EM	CDV2000
28	CDV2510	Edge Computing and DevOps	3	0	0	3	3	S/EM	CDV2000
Total No. of Credits						64			

Table 3.5 : List of course in Project Work basket (PRW)									
S.No	Course Code	Course Name	L	T	P	C	Contact HOURS	Type of Skill	Prerequisite
1	CSS7300	Capstone Project	0	0	0	10	-	S	-
2	CSS7000	Internship	0	0	0	2	-	S	-
3	CSS7100	Mini Project	0	0	0	4	-	S	-
Total No. of Credits						16			

Table 3.6 : List of Mandatory Courses (MAC)						
S.No	Course Code	Course Name	L	T	P	C
1	CHE7601	Environmental Studies	1	0	0	0
2	LAW7601	Indian Constitution	1	0	0	0
3	CIV7601	Universal Human Values and Ethics	1	0	0	0
4	PPS1025	Industry Readiness Program - I	0	0	2	0
5	PPS1026	Industry Readiness Program - II	0	0	2	0
6	APT4002	Introduction to Aptitude	0	0	2	0
7	APT4004	Aptitude Training - Intermediate	0	0	2	0
8	APT4006	Logical and Critical Thinking	0	0	2	0
Total No. of Credits						0

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

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

- 18.1.1** The Internship / In-plant Training / Skill-based Program / IM / RPL shall be conducted in accordance with the Internship Policy prescribed by the University from time to time.
- 18.1.2** Internship: student shall undergo internship, either in industry / company, academic / research organizations, government bodies, or international institutions. The objective is to provide practical exposure, industry insights, and real-world experience relevant to the student's field of study.
- 18.1.3** In-plant Training: student shall undergo training / industrial exposure program aimed at providing with practical insights into real-world working environments. The training may be conducted by industries / companies on-campus or through student visits to industries / companies, government bodies / institutions, or technical organizations.
- 18.1.4** Skill-based Program: student shall undergo a certified skill-based program of 30 hours / 04 weeks. Skill-based program should cater to Skill-Enhancement, Practical Focus, and Career orientation, Complementary to

Curriculum or Industry relevant.

- 18.1.5** International Immersion (IM): student shall undergo IM aimed at providing global exposure through collaborations with foreign universities, industries, or research institutions. The International Immersion may include industry visits, expert interactions, and cultural exchange activities, enhancing students' international outlook, communication skills, and professional readiness.
- 18.1.6** Recognition of Prior Learning (RPL): student who shall undergo any formally recognize relevant prior work experience, internships, or project-based learning that meet the internship learning outcomes, thereby allowing students to earn internship credit without repeating equivalent practical training.
- 18.1.7** The number of Internships available for the concerned Academic Term. Further, the available number of Internships / In-plant training / Skill-based Program / IM / RPL shall be awarded to the students by the University on the basis of merit using the CGPA secured by the student and as per the selection criteria. Provided further, the student fulfils the criteria, as applicable, specified by the industry / company, government bodies, academic / research or through certified courses.
- 18.1.8** A student may opt for Internship / In-plant Training / IM / RPL in an industry / company, government bodies / academic / research institution, international bodies of her / his choice, subject to the condition that the concerned student takes the responsibility to arrange the Internship / In-plant Training / IM / RPL on her / his own. Provided further, that the industry / company, government bodies, academic / research institution national or international offering such Internship / Training confirms to the University that the Internship shall be conducted in accordance with the Program Regulations and Internship Policy of the University / Rubrics.
- 18.1.9** A student undergoing RPL must submit an application with supporting documents such as experience letters, project reports, employer feedback, certifications, a self-reflection report etc. Application must be submitted before the commencement of the internship semester.
- 18.1.10** A student selected for an Internship / In-plant Training / Skill-based Program / IM / RPL in an industry / company, government bodies, academic / research institution shall adhere to all the rules and guidelines prescribed in the Internship Policy of the University.

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

18.2 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 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 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 3.7 : 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	Contact Hours	Type of Skill	Pre requisite

1	CDV3400	Fundamentals of Cloud Computing	3	0	0	3	3	S	--
2	CDV3401	Version Control with Git & GitHub/GitLab	3	0	0	3	3	S	--
3	CDV3402	Serverless Computing	3	0	0	3	3	S	--
4	CDV3403	Infrastructure as Code (IaC) with Terraform & Ansible	3	0	0	3	3	S	--
5	CDV3404	Cloud Automation & Orchestration	3	0	0	3	3	S	--
6	CDV3405	Cloud Infrastructure and Systems Software	3	0	0	3	3	S	--
Track 2 – DevOps with AI									
S.No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Pre requisite
1	CDV3406	AI-Powered Chatbots & Automation in DevOps	3	0	0	3	3	S	--
2	CDV3407	AIOps – AI-Driven DevOps & IT Operations	3	0	0	3	3	S	--
3	CDV3408	Predictive Analytics for System Monitoring & Performance Optimization	3	0	0	3	3	S	--
4	CDV3409	AI-Enabled Cloud Cost Optimization	3	0	0	3	3	S	--
5	CDV3410	AI-Powered Incident Management & Root Cause Analysis	3	0	0	3	3	S	--
6	CDV3411	Reinforcement Learning for Self-Healing DevOps Pipelines	3	0	0	3	3	S	--
Track 3 – DevOps with Fin Tech									
S.No	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill	Pre requisite
1	CDV3412	DevOps Fundamentals for Financial Services	3	0	0	3	3	S	--
2	CDV3413	Introduction to FinTech & Digital Banking	3	0	0	3	3	S	--
3	CDV3414	Microservices Architecture for Banking Applications	3	0	0	3	3	S	--

4	CDV3415	Automated Testing & Monitoring in FinTech	3	0	0	3	3	S	--
5	CDV3416	DevSecOps in Financial Services	3	0	0	3	3	S	--
6	CDV3417	Blockchain & Smart Contracts for Financial Transactions	3	0	0	3	3	S	--

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

Table 3.8 : Open Elective Courses Baskets: Minimum Credits to be earned from this Basket is 12									
Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hours	Type of Skill/ Focus	Prerequisites/ Corequisites
Chemistry Basket									
1	CHE1003	Fundamentals of Sensors	3	0	0	3	3	S	-
2	CHE1004	Smart materials for IOT	3	0	0	3	3	S	-
3	CHE1005	Computational Chemistry	2	0	0	2	2	S	-
4	CHE1006	Introduction to Nano technology	3	0	0	3	3	S	-
5	CHE1007	Biodegradable electronics	2	0	0	2	2	S	-
6	CHE1008	Energy and Sustainability	2	0	0	2	2	S	-
7	CHE1009	3D printing with Polymers	2	0	0	2	2	S	-
8	CHE1010	Bioinformatics and Healthcare IT	2	0	0	2	2	S	-
9	CHE1011	Chemical and Petrochemical catalysts	3	0	0	3	3	S	-
10	CHE1012	Introduction to Composite materials	2	0	0	2	2	S	-
11	CHE1013	Chemistry for Engineers	3	0	0	3	3	S	-
12	CHE1014	Surface and Coatings technology	3	0	0	3	3	S	-
13	CHE1015	Waste to Fuels	2	0	0	2	2	S	-
14	CHE1016	Forensic Science	3	0	0	3	3	S	-
Civil Engineering Basket									
1	CIV1001	Disaster mitigation and management	3	0	0	3	3	S	-
2	CIV1002	Environment Science and Disaster Management	3	0	0	3	3	FC	-
3	CIV2001	Sustainability Concepts in Engineering	3	0	0	3	3	S	-
4	CIV2002	Occupational Health and Safety	3	0	0	3	3	S	-
5	CIV2003	Sustainable Materials and Green Buildings	3	0	0	3	3	EM	-
6	CIV2004	Integrated Project Management	3	0	0	3	3	EN	-
7	CIV2005	Environmental Impact Assessment	3	0	0	3	3	EN	-
8	CIV2006	Infrastructure Systems for Smart Cities	3	0	0	3	3	EN	-
9	CIV2044	Geospatial Applications for Engineers	2	0	2	3	4	EM	-
10	CIV2045	Environmental Meteorology	3	0	0	3	3	S	-
11	CIV3046	Project Problem Based Learning	3	0	0	3	3	S	-
12	CIV3059	Sustainability for Professional Practice	3	0	0	3	3	EN	-
Commerce Basket									
1	COM2001	Introduction to Human Resource Management	2	0	0	2	2	F	-
2	COM2002	Finance for Non Finance	2	0	0	2	2	S	-

3	COM2003	Contemporary Management	2	0	0	2	2	F	-
4	COM2004	Introduction to Banking	2	0	0	2	2	F	-
5	COM2005	Introduction to Insurance	2	0	0	2	2	F	-
6	COM2006	Fundamentals of Management	2	0	0	2	2	F	-
7	COM2007	Basics of Accounting	3	0	0	3	3	F	-
Computer Science Basket									
1	CSS2002	Programming in Java	2	0	2	3	4	S/EM	-
2	CSS2003	Social Network Analytics	3	0	0	3	3	S	-
3	CSS2004	Python Application Programming	2	0	2	3	4	S/ EM	-
4	CSS2005	Web design fundamentals	2	0	2	3	4	S/ EM/EN	-
5	CSS3111	Artificial Intelligence : Search Methods For Problem Solving	3	0	0	3	3	S/ EM/EN	-
6	CSS3112	Privacy And Security In Online Social Media	3	0	0	3	3	S/ EM/EN	-
7	CSS3113	Computational Complexity	3	0	0	3	3	S/ EM/EN	-
8	CSS3114	Deep Learning for Computer Vision	3	0	0	3	3	S/ EM/EN	-
9	CSS3115	Learning Analytics Tools	3	0	0	3	3	S/ EM/EN	-
Design Basket									
1	DES1001	Sketching and Painting	0	0	2	1	2	S	-
2	DES1002	Innovation and Creativity	2	0	0	2	2	F	-
3	DES1121	Introduction to UX design	1	0	2	2	3	S	-
4	DES1122	Introduction to Jewellery Making	1	0	2	2	3	S	-
5	DES1124	Spatial Stories	1	0	2	2	3	S	-
6	DES1125	Polymer Clay	1	0	2	2	3	S	-
7	DES2001	Design Thinking	3	0	0	3	3	S	-
8	DES1003	Servicability of Fashion Products	1	0	2	2	3	F	-
9	DES1004	Choices in Virtual Fashion	1	0	2	2	3	F	-
10	DES1005	Fashion Lifestyle and Product Diversity	1	0	2	2	3	F	-
11	DES1006	Colour in Everyday Life	1	0	2	2	3	F	-
12	DES2080	Art of Design Language	3	0	0	3	3	S	-
13	DES2081	Brand Building in Design	3	0	0	3	3	S	-
14	DES2085	Web Design Techniques	3	0	0	3	3	S	-
15	DES2089	3D Modeling for Professionals	1	0	4	3	5	S	-
16	DES2090	Creative Thinking for Professionals	3	0	0	3	3	S	-
17	DES2091	Idea Formulation	3	0	0	3	3	S	-
Electrical and Electronics Basket									
1	EEE1002	IoT based Smart Building Technology	3	0	0	3	3	S	-
2	EEE1003	Basic Circuit Analysis	3	0	0	3	3	S	-
3	EEE1004	Fundamentals of Industrial Automation	3	0	0	3	3	S	-
4	EEE1005	Electric Vehicles & Battery Technology	3	0	0	3	3	S	-
5	EEE1006	Smart Sensors for Engineering Applications	3	0	0	3	3	S	-
Electronics and Communication Basket									
1	ECE1003	Fundamentals of Electronics	3	0	0	3	3	F	-
2	ECE1004	Microprocessor based systems	3	0	0	3	3	F	-
3	ECE3089	Artificial Neural Networks	3	0	0	3	3	S	-
4	ECE3097	Smart Electronics in Agriculture	3	0	0	3	3	F/EM	-
5	ECE3098	Environment Monitoring Systems	3	0	0	3	3	F/EM	-
6	ECE3102	Consumer Electronics	3	0	0	3	3	F/EM	-
7	ECE3103	Product Design of Electronic Equipment	3	0	0	3	3	S/F/ EM / EN	-
8	ECE3106	Introduction to Data Analytics	3	0	0	3	3	F/EM	-
9	ECE3107	Machine Vision for Robotics	3	0	0	3	3	F/EM	-
English Basket									
1	ENG1008	Indian Literature	2	0	0	2	2	-	-
2	ENG1009	Reading Advertisement	3	0	0	3	3	S	-
3	ENG1010	Verbal Aptitude for Placement	2	0	2	3	4	S	-
4	ENG1011	English for Career Development	3	0	0	3	3	S	-

5	ENG1012	Gender and Society in India	2	0	0	2	2	-	-
6	ENG1013	Indian English Drama	3	0	0	3	3	-	-
7	ENG1014	Logic and Art of Negotiation	2	0	2	3	4	-	-
8	ENG1015	Professional Communication Skills for Engineers	1	0	0	1	2	-	-
DSA Basket									
1	DSA2001	Spirituality for Health	2	0	0	2	2	F	-
2	DSA2002	Yoga for Health	2	0	0	2	2	S	-
3	DSA2003	Stress Management and Well Being	2	0	0	2	2	F	-
Kannada Basket									
1	KAN1001	Kali Kannada	1	0	0	1	1	S	-
2	KAN1003	Kannada Kaipidi	3	0	0	3	3	S	-
3	KAN2001	Thili Kannada	1	0	0	1	1	S	-
4	KAN2003	Pradharshana Kale	1	0	2	2	3	S	-
5	KAN2004	Sahithya Vimarshe	2	0	0	2	2	S	-
6	KAN2005	Anuvadha Kala Sahithya	3	0	0	3	3	S	-
7	KAN2006	Vichara Manthana	3	0	0	3	3	S	-
8	KAN2007	Katha Sahithya Sampada	3	0	0	3	3	S	-
9	KAN2008	Ranga Pradarshana Kala	3	0	0	3	3	S	-
Foreign Language Basket									
1	FRL1004	Introduction of French Language	2	0	0	2	2	S	-
2	FRL1005	Fundamentals of French	2	0	0	2	2	S	-
3	FRL1009	Mandarin Chinese for Beginners	3	0	0	3	3	S	-
Law Basket									
1	LAW1001	Introduction to Sociology	2	0	0	0	2	HP	-
2	LAW2001	Indian Heritage and Culture	2	0	0	0	2	HP/GS	-
3	LAW2002	Introdcution to Law of Succession	2	0	0	0	2	HP/GS	-
4	LAW2003	Introduction to Company Law	2	0	0	0	2	HP	-
5	LAW2004	Introduction to Contracts	2	0	0	2	2	F	-
6	LAW2005	Introduction to Copy Rights Law	2	0	0	2	2	F	-
7	LAW2006	Introduction to Criminal Law	2	0	0	2	2	F	-
8	LAW2007	Introduction to Insurance Law	2	0	0	2	2	F	-
9	LAW2008	Introduction to Labour Law	2	0	0	2	2	F	-
10	LAW2009	Introduction to Law of Marriages	2	0	0	2	2	F	-
11	LAW2010	Introduction to Patent Law	2	0	0	2	2	F	-
12	LAW2011	Introduction to Personal Income Tax	2	0	0	2	2	F	-
13	LAW2012	Introduction to Real Estate Law	2	0	0	2	2	F	-
14	LAW2013	Introduction to Trademark Law	2	0	0	2	2	F	-
15	LAW2014	Introduction to Competition Law	3	0	0	3	3	F	-
16	LAW2015	Cyber Law	3	0	0	3	2	F	-
17	LAW2016	Law on Sexual Harrassment	2	0	0	2	2	F	-
18	LAW2017	Media Laws and Ethics	2	0	0	2	2	F	-
Mathematics Basket									
1	MAT2008	Mathematical Reasoning	3	0	0	3	3	S	-
2	MAT2014	Advanced Business Mathematics	3	0	0	3	3	S	-
3	MAT2041	Functions of Complex Variables	3	0	0	3	3	S	-
4	MAT2042	Probability and Random Processes	3	0	0	3	3	S	-
5	MAT2043	Elements of Number Theory	3	0	0	3	3	S	-
6	MAT2044	Mathematical Modelling and Applications	3	0	0	3	3	S	-
Mechanical Basket (not to be offered for Mechanical Department students)									
1	MEC1001	Fundamentals of Automobile Engineering	3	0	0	3	3	F	-
2	MEC1002	Introduction to Matlab and Simulink	3	0	0	3	3	S/EM	-
3	MEC1003	Engineering Drawing	1	0	4	3	5	S	-
4	MEC2001	Renewable Energy Systems	3	0	0	3	3	F	-

5	MEC2002	Operations Research & Management	3	0	0	3	3	F	-
6	MEC2003	Supply Chain Management	3	0	0	3	3	S/ EM/ EN	-
7	MEC2004	Six Sigma for Professionals	3	0	0	3	3	S/EM	-
8	MEC2005	Fundamentals of Aerospace Engineering	3	0	0	3	3	F	-
9	MEC2006	Safety Engineering	3	0	0	3	3	S/EM	-
10	MEC2007	Additive Manufacturing	3	0	0	3	3	F/EM	-
11	MEC3069	Engineering Optimisation	3	0	0	3	3	S/EM	-
12	MEC3070	Electronics Waste Management	3	0	0	3	3	F/S	-
13	MEC3071	Hybrid Electric Vehicle Design	3	0	0	3	3	S/EM	-
14	MEC3072	Thermal Management of Electronic Appliances	3	0	0	3	3	S/EM	-
15	MEC3200	Sustainable Technologies and Practices	3	0	0	3	3	S/EM	-
16	MEC3201	Industry 4.0	3	0	0	3	3	S/EM	-
Petroleum Basket									
1	PET1011	Energy Industry Dynamics	3	0	0	3	3	FC	-
2	PET1012	Energy Sustainability Practices	3	0	0	3	3	FC	-
1	PHY1003	Mechanics and Physics of Materials	3	0	0	3	3	FC / SD	
2	PHY1004	Astronomy	3	0	0	3	3	FC	
3	PHY1005	Game Physics	2	0	2	3	4	FC / SD	
4	PHY1006	Statistical Mechanics	2	0	0	2	2	FC	
5	PHY1007	Physics of Nanomaterials	3	0	0	3	3	FC	
6	PHY1008	Adventures in nanoworld	2	0	0	2	2	FC	
7	PHY2001	Medical Physics	2	0	0	2	2	FC	
8	PHY2002	Sensor Physics	1	0	2	2	3	FC / SD	
9	PHY2003	Computational Physics	1	0	2	2	3	FC	
10	PHY2004	Laser Physics	3	0	0	3	3	FC	
11	PHY2005	Science and Technology of Energy	3	0	0	3	3	FC	
12	PHY2009	Essentials of Physics	2	0	0	2	2	FC	
1	MGT2007	Digital Entrepreneurship	3	0	0	3	3	S/EM/EN	-
2	MGT2015	Engineering Economics	3	0	0	3	3	S	-
3	MGT2023	People Management	3	0	0	3	3	S/EM/ EN	-
Management Basket- II									
1	MGT1001	Introduction to Psychology	3	0	0	3	3	F	-
2	MGT1002	Business Intelligence	3	0	0	3	3	EN	-
3	MGT1003	NGO Management	3	0	0	3	3	S	-
4	MGT1004	Essentials of Leadership	3	0	0	3	3	EM/ EN	-
5	MGT1005	Cross Cultural Communication	3	0	0	3	3	S/EM/ EN	-
6	MGT2001	Business Analytics	3	0	0	3	3	S/ EM/EN	-
7	MGT2002	Organizational Behaviour	3	0	0	3	3	F	-
8	MGT2003	Competitive Intelligence	3	0	0	3	3	S	-
9	MGT2004	Development of Enterprises	3	0	0	3	3	S/EM/EN	-
10	MGT2005	Economics and Cost Estimation	3	0	0	3	3	S/EM	-
11	MGT2006	Decision Making Under Uncertainty	3	0	0	3	3	S	-
12	MGT2008	Econometrics for Managers	3	0	0	3	3	S	-
13	MGT2009	Management Consulting	3	0	0	3	3	S/EM/EN	-
14	MGT2010	Managing People and Performance	3	0	0	3	3	S/EM/EN	-
15	MGT2011	Personal Finance	3	0	0	3	3	F	-
16	MGT2012	E Business for Management	3	0	0	3	3	S/EM	-
17	MGT2013	Project Management	3	0	0	3	3	EN / EM	-
18	MGT2014	Project Finance	3	0	0	3	3	EN / EM	-
19	MGT2016	Business of Entertainment	3	0	0	3	3	EM/ EN	-
20	MGT2017	Principles of Management	3	0	0	3	3	S/EM/ EN	-
21	MGT2018	Professional and Business Ethics	3	0	0	3	3	S/EM/ EN	-
22	MGT2019	Sales Techniques	3	0	0	3	3	S/EM/ EN	-
23	MGT2020	Marketing for Engineers	3	0	0	3	3	S/EM/ EN	-

24	MGT2021	Finance for Engineers	3	0	0	3	3	S/EM/ EN	-
25	MGT2022	Customer Relationship Management	3	0	0	3	3	S/EM/ EN	-
Media Studies Basket									
1	BAJ3050	Corporate Filmmaking and Film Business	0	0	4	2	4	EM	-
2	BAJ3051	Digital Photography	2	0	2	3	4	EM	-
3	BAJ3055	Introduction to News Anchoring and News Management	0	0	2	1	2	EM	-

21.List of MOOC Courses

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

- 21.1.1 The student needs to study and complete School Core and Program Core Courses in offline mode only.
- 21.1.2 Massive Open Online Course (MOOC) courses maybe given for Open Elective and Discipline Elective Courses. These courses need to be approved by the concerned BOS and Academic Council from time to time.
- 21.1.3 SWAYAM/NPTEL/ other approved MOOCs shall be approved by the concerned Board of Studies and placed in the concerned PRC.
- 21.1.4 Student shall register for these courses in the ERP of Presidency University.
- 21.1.5 For these MOOC courses faculty coordinators are identified. These faculty should have undergone similar MOOC courses and therefore should be familiar with the mode of class conduction, types of assessments and evaluation procedures.
- 21.1.6 Study materials shall be provided to the students as video lectures shared by the MOOCs Coordinator(s), or the students may access the approved MOOCs Portal directly. The mode of class conduction is determined by the MOOCs coordinator(s) as detailed in the Course Catalogue and Course Plan.
- 21.1.7 The question paper shall be prepared by the MOOCs coordinator(s).
- 21.1.8 Students write the exams in online mode. These exams are scheduled and conducted by the School.
- 21.1.9 Results are evaluated by School and given to the Office of the Controller of Examinations (CoE).
- 21.1.10 The details of the duration, credits and evaluation are given below:

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

2.	8 weeks	2	Mid Term-50 Marks End Term-100 Marks
3	4 weeks	1	End Term-100 Marks

21.2 MOOC - Discipline Elective Courses for B. Tech. Computer Science and Technology DevOps

Table 3.5 : MOOC Discipline Elective Courses				
Sl.No	Course Code	Course Name	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	CSE3196	Foundations of Cyber Physical Systems	3	3-0-0-3
16	CSE3197	Getting Started with Competitive Programming	3	3-0-0-3
17	CSE3198	GPU Architectures and Programming	3	3-0-0-3
18	CSE3199	Artificial Intelligence: Knowledge Representation and Reasoning	3	3-0-0-3
19	CSE3200	Programming in Modern C++	3	3-0-0-3
20	CSE3201	Circuit Complexity Theory	3	3-0-0-3
21	CSE3202	Basics of Computational Complexity	3	3-0-0-3

22	CSE3212	Introduction to Computer and Network Performance Analysis using Queuing	1	1-0-0-1
23	CSE3213	C Programming and Assembly Language	1	1-0-0-1
24	CSE3214	Python for Data Science	1	1-0-0-1
25	CSE3215	Software Conceptual Design	1	1-0-0-1
26	CSE3117	Industrial Digital Transformation	3	3-0-0-3
27	CSE3118	Blockchain for Decision Makers	3	3-0-0-3
28	CSE3349	Technology for Lawyers	3	3-0-0-3
29	CSE3430	Deep Learning for Natural Language Processing	3	3-0-0-3
30	CSE3431	Machine Learning for Engineering and Science Applications	3	3-0-0-3
31	CSE3432	Algorithms in Computational Biology and Sequence Analysis	3	3-0-0-3
32	CSE3433	Introduction to Large Language Models (LLMs)	3	3-0-0-3
33	CSE3434	Quantum Algorithms and Cryptography	3	3-0-0-3
34	CAI3430	Responsible & Safe AI Systems	3	3-0-0-3
35	CCS3416	Practical Cyber Security for Cyber Security Practitioners	3	3-0-0-3
36	IST3409	Design & Implementation of Human-Computer Interfaces	3	3-0-0-3

21.3 MOOC - Open Elective Courses for B. Tech. Computer Science and Technology DevOps

Table 3.6: MOOC Open Elective Courses				
Courses duration is 4 weeks (01 credit)/ 8 weeks (02 credits)/ 12 weeks (03 credits)				
Sl. No.	Course Code	Course Name	Total Credits	L-T-P-C
1	BBA2022	Supply Chain digitization	3	3-0-0-3
2	BBA2021	E Business	3	3-0-0-3
3	BBB2016	Business Analytics for Management Decisions	3	3-0-0-3
4	BBB2015	Artificial Intelligence for Investments	3	3-0-0-3
5	MEC3001	Design and Development of Product	1	1-0-0-1
6	ENG3004	Perspectives of Neurolinguistics	1	1-0-0-1
7	PPS4009	Working in Contemporary Teams	1	1-0-0-1
8	MGT3001	Data Analysis and Decision Making	3	3-0-0-3
9	MEC3001	Design and Development of Product	1	1-0-0-1
10	EEE3105	Microsensors and Nanosensors	3	3-0-0-3
11	CIV3065	Drone Systems and Control	3	3-0-0-3
12	ECE3183	Neural Networks for Signal Processing - I	3	3-0-0-3

13	CIVXXXX	Disaster Management	3	3-0-0-3
----	---------	---------------------	---	---------

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

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

First year - CYCLE 2

Sl. No.	Course Code	Course Name	L	T	P	C	Contact hours	Type of Skill/Focus	Course Caters to	Basket
Semester 1 - Physics Cycle			114	1	12	20	27			
1	MAT1212	Mathematical foundations for Machine Learning and Data Analysis	3	1	0	4	4	F		BSC
2	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S		ESC
3	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S		BSC
4	CSS2200	Problem Solving using C	2	0	0	2	2	S		PCC
5	ENG2030	Communication in Practice	1	0	2	2	3	S		HSMC
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S		ESC
7	CSS2201	Problem Solving using C Lab	0	0	4	2	4	S		PCC
8	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S		BSC
9	PPS1026	Industry Readiness Program – II	0	0	2	0	2	S		MAC
10	EEE1251	Basics of Electrical and Electronics Engineering Laboratory	0	0	2	1	2	F/S		ESC
Semester 1 – Chemistry Cycle			15		1	12	21	28		
1	MAT1211	Linear Algebra, Differential Equations and Statistical Methods			1	0	4	4	F	BSC
2	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F		BSC
3	MEC1006	Engineering Graphics	2	0	0	2	2	S		ESC
4	ENG1030	Fundamentals of Communication	1	0	2	2	2	S		HSMC

5	CSS1500	Computational Thinking using Python	2	0	2	3	4	S		ESC
6	ECE2022	Digital Design	2	0	0	2	2	F/S		ESC
7	ECE1512	Design Thinking in Practice	2	0	2	3	4	F		HSMC
8	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F		BSC
9	PPS1025	Industry Readiness Program – I	0	0	2	0	2	S		MAC
10	ECE2052	Digital Design Lab	0	0	2	1	2	F/S		ESC

First year - CYCLE 2

Sl. No.	Course Code	Course Name	L	T	P	C	Contact hours	Type of Skill/Focus	Course Caters to	Basket
Semester 2 - Chemistry Cycle			14	1	12	20	27			
1	MAT1212	Mathematical foundations for Machine Learning and Data Analysis	3	1	0	4	4	F		BSC
2	CIV1200	Foundations of Integrated Engineering	2	0	0	2	2	S		ESC
3	CHE2501	Chemistry of Smart Materials	3	0	0	3	3	S		BSC
4	CSS2200	Problem Solving using C	2	0	0	2	2	S		PCC
5	ENG2030	Communication in Practice	1	0	2	2	3	S		HSMC
6	EEE1200	Basics of Electrical and Electronics Engineering	3	0	0	3	3	F/S		ESC
7	CSS2201	Problem Solving using C Lab	0	0	4	2	4	S		PCC
8	CHE2502	Chemistry of Smart Materials Lab	0	0	2	1	2	S		BSC
9	PPS1026	Industry Readiness Program – II	0	0	2	0	2	S		MAC
10	EEE1251	Basics of Electrical and Electronics Engineering Laboratory	0	0	2	1	2	F/S		ESC

Semester 2 – Physics Cycle			15	1	12	21	28			
1	MAT1211	Linear Algebra, Differential Equations and Statistical Methods	3	1	0	4	4	F		BSC
2	PHY2501	Optoelectronics and Quantum Physics	3	0	0	3	3	F		BSC

3	MEC1006	Engineering Graphics	2	0	0	2	2	S		ESC
4	ENG1030	Fundamentals of Communication	1	0	2	2	3	S		HSMC
5	CSS1500	Computational Thinking using Python	2	0	2	3	4	S		ESC
6	ECE2022	Digital Design	2	0	0	2	2	F/S		ESC
7	ECE1512	Design Thinking in Practice	2	0	2	3	4	F		HSMC
8	PHY2504	Optoelectronics and Quantum Physics Lab	0	0	2	1	2	F		BSC
9	PPS1025	Industry Readiness Program – I	0	0	2	0	2	S		MAC
10	ECE2052	Digital Design Lab	0	0	2	1	2	F/S		ESC

Semester 3			22	1	8	25	31			
1	CIV7601	Universal Human Values and Ethics	1	0	0	0	1		S	MAC
2	MAT2311	Vector Calculus, Numerical Methods and Sampling Distributions	3	1	0	4	4		EM	BSC
3	CSS2251	Data Communication and Computer Networks	3	0	0	3	3		S	PCC
4	CSS2252	Data Communication and Computer Networks Lab	0	0	2	1	2		S	PCC
5	CSS2253	Data Structures	3	0	0	3	3		S	PCC
6	CSS2254	Data Structures Lab	0	0	2	1	2		S	PCC
7	CSS2255	Object Oriented Programming Using Java	3	0	0	3	3		S/EM	PCC
8	CSS2256	Object Oriented Programming Using Java Lab	0	0	2	1	2		S/EM	PCC
9	CSS2257	Computer Organization and Architecture	3	0	0	3	3		S	PCC
10	CDV2000	DevOps Foundations	3	0	0	3	3		S/EM	PCC
11	CSS2271	Software Design and Development	3	0	0	3	3		S	PCC
12	APT4002	Introduction to Aptitude	0	0	2	0	2		S	MAC

Semester 4			22	2	12	28	36			
1	CHE7601	Environmental Studies	1	0	0	0	1		F	MAC
2	MAT2321	Discrete Mathematical Structures	3	1	0	4	4		F	BSC
3	CSS2258	Web Technologies	3	0	0	3	3		S/EM	PCC

4	CSS2259	Web Technologies Lab	0	0	2	1	2		S/EM	PCC
5	CSS2260	Database Management Systems	3	0	0	3	3		S	PCC
6	CSS2261	Database Management Systems Lab	0	0	2	1	2		S	PCC
7	CDV2501	Devops Tools Internals	3	0	0	3	3		S	PCC
8	CDV2502	Devops Tools Internals Lab	0	0	2	1	2		S	PCC
9	CSS2262	Analysis of Algorithms	3	1	0	4	4		S	PCC
10	CSS2263	Analysis of Algorithms Lab	0	0	2	1	2		S	PCC
11	CSS2264	Essentials of AI	3	0	0	3	3		S/EM	ESC
12	CSS2265	Essentials of AI Lab	0	0	2	1	2		S/EM	ESC
13	FIN1002	Essentials of Finance	3	0	0	3	3		S	HSMC
14	APT4004	Aptitude Training - Intermediate	0	0	2	0	2		S	MAC

Semester 5			15	0	6	19	21			
1	CSS2266	Theory of Computation	3	0	0	3	3		S	PCC
2	CDV2001	Agile Structures and Frameworks	3	0	0	3	3		S	PCC
3	CSS2269	Operating Systems	3	0	0	3	3		S	PCC
4	CSS2270	Operating Systems Lab	0	0	2	1	2		S/EM	PCC
5	CDV2503	Software Testing	3	0	0	3	3		S/EM	PCC
6	CDV2504	Software Testing Lab	0	0	2	1	2		S/EM	PCC
7	CSSXXXX	Professional Elective – I	3	0	0	3	3		S/EM	PEC
8	CSS7000	Internship	-	-	-	2	0		S/EM	PRW
9	APT4006	Logical and Critical Thinking	0	0	2	0	2		S	MAC
Semester 6			16	0	10	20	26			
1	LAW7601	Indian Constitution	1	0	0	0	1		S	MAC
2	CDV2508	Continuous Integration and Continuous Delivery (CI/CD) Pipelines	3	0	0	3	3		S/EM	PCC
3	CDV2512	Continuous Integration and Continuous Delivery (CI/CD) Pipelines Lab	0	0	4	2	4		S/EM	PCC
4	CDV2510	Edge Computing and DevOps	3	0	0	3	3		S/EM	PCC
5	CSS2274	Competitive Programming and Problem Solving	0	0	4	2	4		S/EM	ESC

6	CSSXXXX	Professional Elective – II	3	0	0	3	3		S	PEC
7	CSSXXXX	Professional Elective – III	3	0	0	3	3		S	PEC
8	XXXXXXX	Open Elective – I	3	0	0	3	3		S	OEC
9	APT4005	Aptitude for Employability	0	0	2	1	2		S	HSMC

Semester 7			12	0	2	17	14			
1	CSSXXXX	Professional Elective – IV	3	0	0	3	3	S	PEC	
2	CSSXXXX	Professional Elective – V	3	0	0	3	3	S	PEC	
3	CSSXXXX	Professional Elective – VI	3	0	0	3	3	S	PEC	
4	XXXXXXX	Open Elective – II	3	0	0	3	3	S	OEC	
5	CSS7100	Mini Project	-	-	-	4	0	S	PRW	
6	PPS3018	Preparedness for Interview	0	0	2	1	2	S	HSMC	

Semester 8			0	0	0	10	0			
1	CSS7300	Capstone Project	-	-	-	10	0	S/EM	PRW	
			117	4	62	160	183			

22. 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: MAT1212	Course Title: Mathematical foundations for Machine Learning and Data Analysis Type of Course:1] School Core	L-T- P- C	3-1-0-4
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course provides a comprehensive foundation in advanced mathematical concepts essential for engineering and scientific applications. It covers integral calculus and special functions, linear transformations and matrix decompositions, vector spaces and inner product spaces, and probability theory with random variables and standard probability distributions. The course emphasizes mathematical modeling, analytical problem-solving, data analysis techniques, and engineering applications, enabling students to apply mathematical tools to solve real-world engineering problems.		
Course Objective	The course aims to develop a strong conceptual understanding of vector spaces, linear algebra, and vector calculus. It introduces mathematical techniques used in engineering analysis and modelling, enabling students to apply these concepts effectively in solving engineering problems. The course also enables students to apply probability distributions and statistical methods in practical situations involving uncertainty and data analysis. Further, it strengthens analytical and problem-solving skills through applications of differential equations and statistical inference. In addition, the course familiarizes students with computational tools for solving engineering mathematics problems, thereby enhancing their ability to address real-world engineering challenges.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Apply integral calculus and special functions to solve engineering problems. (L3) CO2: Solve problems involving linear transformations and matrix techniques. (L3) CO3: Apply concepts of vector spaces, basis, dimension, and orthogonality in mathematical applications. (L3) CO4: Apply probability concepts and random variable distributions to solve problems involving uncertainty and stochastic phenomena (L3) CO5: Examine discrete and continuous random variables using probability distributions such as Binomial, Poisson, Uniform, Exponential, and Normal distributions. (L3) CO6: Utilize computational tools such as MATLAB, Python, or similar software for mathematical computation, data analysis, and engineering applications. (L3)		
Course Content:			
Module 1	Integral Calculus and Special Functions	(8L+3T=11 Sessions)	
Definite and indefinite integrals, Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Surface areas, volumes, Sterlin approximation theorem.			
Module 2	Vector Spaces and Inner Product Spaces	(12L+4T=16 Sessions)	

Vector Space, Examples, Properties, Subspaces, Linear Combinations, Spanning Set of a Vector Space, Linear Dependence and Independence, Basis and Dimension, Change of Basis; Normed Linear Spaces, Inner Products Spaces, Orthogonality, Orthogonal and Orthonormal Bases, Gram-Schmidt Orthogonalization Process, Orthogonal Complements, Orthogonal Projections. Applications to Difference Equations, Markov Chains.		
Module 3	Linear Transformations and Matrix Decompositions	(10L+3T=13 Sessions)
Linear Transformations, Algebra of Transformations, Singular and Nonsingular Linear Mappings, Isomorphisms, Matrix Representation of Linear Transformations, The Range and Null Space, Rank-Nullity Theorem. Least Squares Approximation, Singular Value Decomposition: Singular Values, Computing Singular Value Decomposition, Data Compression Using Singular Value Decomposition and Introduction to Principal Component Analysis.		
Module 4	Probability and Random Variables for machine learning	(15L+5T=20 Sessions)
Introduction to Probability, Conditional Probability, Bayes' Theorem and Problems. Random Variables (discrete and continuous), Probability Mass/Density Functions, Mathematical Expectations, discrete probability distributions - Binomial distribution, Poisson distribution, Continuous uniform distribution - Exponential distribution, Normal distribution. Machine learning: large language models		
Targeted Applications & Tools that can be used: - Evaluation of definite and improper integrals for computing areas, volumes, surface areas, engineering design calculations, and scientific modeling using Beta and Gamma functions. - Representation and analysis of high-dimensional vector spaces, feature extraction, orthogonal projections, signal processing, computer graphics, pattern recognition, and machine learning. - Least-squares regression, image compression, recommendation systems, dimensionality reduction, Principal Component Analysis (PCA), data mining, and machine learning applications. - Probabilistic modeling, Bayesian inference, uncertainty quantification, predictive analytics, classification algorithms, statistical learning, and foundations of large language models (LLMs). Software: MATLAB, Mathematica, Maple, Python, GeoGebra, Octave, R, Scilab.		
Text Book E. Kreyszig, <i>Advanced Engineering Mathematics</i>, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015. B. S. Grewal, <i>Higher Engineering Mathematics</i>, 45th ed. New Delhi, India: Khanna Publishers, 2017. sley Publishing Company.		
Assignment:		
Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor.		
References: G. Strang, <i>Introduction to Linear Algebra</i>, 6th ed. Wellesley, MA, USA: Wellesley-Cambridge Press, 2023. D. C. Lay, S. R. Lay, and J. J. McDonald, <i>Linear Algebra and Its Applications</i>, 6th ed. Harlow, U.K.: Pearson, 2021. E. Kreyszig, <i>Advanced Engineering Mathematics</i>, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015. B. S. Grewal, <i>Higher Engineering Mathematics</i>, 45th ed. New Delhi, India: Khanna Publishers, 2017. C. D. Meyer, <i>Matrix Analysis and Applied Linear Algebra</i>, 2nd ed. Philadelphia, PA, USA: SIAM, 2023. E-resources/ Web links: https://open.umn.edu/opentextbooks/textbooks/24 https://open.umn.edu/opentextbooks/textbooks/introduction-to-probability https://open.umn.edu/opentextbooks/textbooks/linear-transformations-on-vector-spaces		
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.		
Catalogue prepared by	Dr. Pavithra Sivaprakasam	
Recommended by the Board of Studies on		
Date of Approval by the Academic Council		

Course Code: CIV1200	Course Title: Foundations of Integrated Engineering 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 interdisciplinary course introduces first-year engineering students to foundational principles and practices across key engineering domains, emphasizing real-world problem-solving, sustainability, and ethical innovation. Students explore how civil, mechanical, electrical, and IT systems intersect with emerging technologies like IoT, AI, and geomatics to address global challenges. Through case studies, learners gain deeper understanding of smart infrastructure, prototyping mechanical/electronic systems, and securing IT solutions. Topics include bioinformatics for environmental monitoring, GIS-enabled urban planning, renewable energy integration, and cybersecurity fundamentals. The course cultivates a holistic understanding of engineering's role in sustainable development, safety, and ethical decision-making, preparing students to contribute meaningfully to multidisciplinary projects in a technology-driven world.		
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] Recall key principles of Agile, DevOps, and bioinformatics used in interdisciplinary engineering contexts. 2] Explain the role of GIS, LiDAR, and sustainable materials in designing smart infrastructure and disaster management systems. 3] Describe core components of mechanical systems and their real-world applications. 4] Describe the functionality of IoT-enabled wearable devices, embedded systems, and renewable energy integration in smart grids. 5] List foundational IT concepts such as cloud computing architectures, cybersecurity threats, and blockchain applications.		
Course Content:			
Module 1	Foundations of Engineering Practice	Assignment	Case studies 6 Sessions
Real-world problem-solving using data logic and practical applications, Collaboration and Innovation through multi-domain project, Engineering Ethics & Environmental Impact Emerging Fields: Automation, and Introduction to bioinformatics and its application Sustainability & Safety: Circular economy principles, carbon footprint analysis.			
Module 2	Civil Engineering & Geomatics	Assignment	Article Review 6 Sessions

Smart Infrastructure & Geomatics: GIS mapping, LiDAR, drone surveys for urban planning, Geospatial data analysis for disaster management. Sustainable Construction: 3D-printed structures, self-healing concrete, Digital twins for infrastructure monitoring. Green Innovations: Net-zero energy buildings, rainwater harvesting systems.				
Module 3	Mechanical Engineering in Action	Assignment & Quiz	Data Collection	6 Sessions
Advanced Manufacturing: Collaborative robots (cobots), additive manufacturing and 3D printing, Reverse engineering and prototyping. Energy Systems: Solar/wind energy harvesting, piezoelectric applications. Biomechanics: Prosthetics design, ergonomic product lifecycle.				
Module 4	Electrical & Electronics Engineering	Assignment & Quiz	Data Collection and visualization	6 Sessions
Smart Devices & Systems: Embedded systems, Wearable technology, Edge computing and hardware platforms Energy Innovations: EV charging infrastructure, wireless power transfer, Smart grid integration with renewables.				
Module 5	Fundamentals of IT	Assignment & Quiz	Case studies	6 Sessions
Core IT Topics: Networking basics, Cloud computing Cybersecurity & Data: Encryption, phishing prevention, zero-trust models, Database management. Emerging Tech: Blockchain for supply chains, AI/ML basics, IoT integration with cloud platforms				
Targeted Application & Tools that can be used: Application Areas include Interdisciplinary problem-solving, Smart city planning, disaster management, Robotics prototyping, renewable energy systems, Wearable health tech, smart grids, Secure cloud systems. Tools: 3D Printers, Autocad, Tinkercad, ArcGIS / QGIS, Arduino/Raspberry Pi				
Text Book: T1: William Oakes & Les Leone, "Engineering Your Future: An Introduction to Engineering", Oxford University Press, 9th Edition, 2021 T2: Barry F. Kavanagh, "Introduction to Geomatics", Pearson, 5th Edition, 2021 T3: Ian Gibson, David Rosen, & Brent Stucker, "Additive Manufacturing Technologies", Springer, 3rd Edition, 2021 T4: Sudip Misra, "The Internet of Things: Enabling Technologies, Protocols, and Use Cases", Wiley, 2nd Edition, 2022 T5: James Kurose & Keith Ross, "Computer Networking: A Top-Down Approach", Pearson, 8th Edition, 2020				

References

- R1: Supratim Choudhuri, "Bioinformatics for Beginners: Genes, Genomes, and Molecular Evolution", Academic Press, 1st Edition, 2023,
- R2: Robert McGinn, "The Ethical Engineer: Contemporary Concepts and Cases", Princeton University Press, 1st Edition, 2020
- R3: Charles J. Kibert, "Sustainable Construction: Green Building Design and Delivery", Wiley, 5th Edition, 2022
- R4: Anthony M. Townsend, "Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia", W.W. Norton & Company, 1st Edition, 2020
- R5: David Buchla, "Renewable Energy Systems: A Smart Energy Systems Approach", Pearson, 2nd Edition, 2023
- R6: Charles Platt, "Make: Electronics: Learning Through Discovery", Make Community, 3rd Edition, 2021
- R7: Charles J. Brooks, Christopher Grow, & Philip Craig, "Cybersecurity Essentials", Wiley, 2nd Edition, 2021

Web-resources:

W1: Post-parametric Automation in Design and Construction

W2: <https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1155197&site=ehost-live>

Smart Cities : Introducing Digital Innovation to Cities

W3: <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

W4: <https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2323766&site=ehost-live>

Additive Manufacturing: Opportunities, Challenges, Implications

W5: <https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1134464&site=ehost-live>

Course Code: CHE2501	Course Name: Chemistry of Smart Materials Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	CHE2501		
Anti-requisites	CSEXXXX		
Course Description	The objective of the course is to introduce the students to concepts and applications of chemistry of smart materials. The course also aims to enhance the knowledge of smart materials associated with memory system, display devices, , sensors, energy devices and environment. It will also cultivate an ability to identify chemistry in each of smart engineered		

	materials and interpret solutions for the challenges connected to memory, display, energy, smart, green and sustainable technologies. It targets to strengthen the fundamental concepts behind chemistry of smart materials and then builds an interface with their industrial applications. This course is designed to cater to Environment and Sustainability			
Course Objective	The objective of the course is ‘SKILL DEVELOPMENT’ of the student by using Participative learning techniques.			
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Relate the knowledge of chemistry to computational approaches to identify materials functionalities and properties L1- Remember CO2:Recognize and interpret solutions for the challenges connected to memory, display, smart, green and sustainable technologies. L2- Understand CO3:Explain the quality parameters of engineering materials associated with environment and sensors. L3 – Apply CO4:Interpret the knowledge of sustainable chemistry for E- waste management. L4 – Analyze CO5:Analyse the importance of various electrochemical sources in energy systems. L5 – Evaluate			
Course Content:				
Module 1	Computational Chemistry	Assignment	Data Collection and analysis	09 Sessions
Topics: Fundamental particles of atom – their mass, charge and location – atomic number and mass number, Stabilizing interactions: Bonded and non-bonded interactions. Chemistry of weak interactions – van der Waals force and hydrogen bonding, Density functional theory. 3D co-ordinate generation for small molecules, geometry optimization by Molview. Chemical Databases: Chemoinformatics, MSDS Self- learning topics: Scope, cost and efficiency of computational modeling.				
Module 2	Materials for Memory and Display Systems	Assignment	Data Collection and analysis	09 Sessions
Topics: Memory Systems : Introduction, classification of electronic memory devices- Transistor, capacitor, charge -transfer and Resistor, types of materials - organic, polymeric and hybrid materials, and applications, manufacturing of semiconductor chips. Display Systems: photo and electroactive materials , materials for display -Principle, Properties and applications: Liquid crystals for LCD-Liquid crystals display, Basics of LED: OLED-organic light emitting diode and light emitting electrochemical cells. Self- learning topics: Green computing: Biocomposite based memory devices				
Module 3	Nanomaterials based Smart Sensors and Devices	Assignment	Data Collection and analysis	09 Sessions
Topics:				

Nanomaterials- Introduction, classification based on dimensionality, quantum confinement. Size dependent properties, Synthesis, Properties of CNT and Graphene and their application as Materials for data analysis and packaging -RFID and IONT. Sensors: Introduction, types, Principle and applications- electrochemical sensor: nanomaterials for sensing applications - Glucose, VOC sensing. Self-learning topics: Fullerene, biomolecules in sensing, Strain sensors				
Module 4	Sustainable Materials and Development	Quiz/Seminar	Data Collection and analysis	09 Sessions
Topics: E waste: Introduction, E waste Hazards, E- waste management, Recovery of precious metal- Cu by Hydrometallurgy. Green Chemistry: Fundamentals and 12 principles with examples, Carbon footprint and sequestration Sustainable Chemistry: -Introduction to Biomaterials- PLA , polymers in bio-compatible and bio-degradable materials - Poly(3-hydroxybutyrate-co-3-hydroxyvalerate) PHBV, synthesis and applications in drug delivery. <i>Self-learning topics:</i> circular economy- case studies.				
Module 5	Energy Science	Quiz/Seminar	Data Collection and analysis	09 Sessions
Topics: Battery technology: Fundamentals of electrochemistry, Introduction to electrochemical storage devices: battery (Lithium-ion battery- LiMnO_2, LiCoO_2, metal air batteries- LiO_2) and supercapacitors- Introduction, Principle, Types - EDLC, pseudo and asymmetric capacitor. Photovoltaic: Solar cells - Construction and working principle; types- Inorganic, Organic and quantum dot sensitized (QDSSC's). <i>Self-learning topics:</i> Battery technology for e-mobility, Green hydrogen				
Targeted Application & Tools that can be used: Application areas are Data storage and analysis, logistics, Biomedicine, Energy, Environment and sustainability Tools: Molview, chemdraw, excel etc				
Project work/Assignment:				
Assessment Type • Midterm exam • Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screenshot accessing the digital resource.) • Quiz/Student Seminar • End Term Exam • Self-learning				
Text Book T1: Wiley, "Engineering Chemistry", Wiley. T2: G.A. Ozin and A.C. Arsenault, Nanochemistry: A chemical approach to nanomaterials, Royal Society 2009				
Reference Books R1: Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press 978-036-727-510-5.				

- R2: E-waste recycling and management: present scenarios and environmental issues, Khan, Anish, and Abdullah M. Asiri. 2019, Springer, Vol. 33. ISBN: 978-3-030-14186-8.
- R3: Essentials of computational chemistry: theories and models, Christopher J Cramer, 2013, John Wiley & Sons. ISBN: 978-0-470-09182-1.
- R4: Energy storage and conversion devices: Supercapacitors, batteries and hydroelectric cells, Anurag Gaur, L. Sharma, Anil Arya. 2021, CRC press, 1st edition, ISBN: 978-1-003-14176-1.
- R5: Fundamentals of analytical chemistry: An introduction, Douglas A. Skoog et al., 2004 Thomson Asia pte Ltd., 8th, ISBN: 978-0-495-55828-6
- R6: Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press, ISBN: 978-036-727-510-5.
- R7: Electrical and electronic devices, circuits and materials: Technological challenges and solutions. Tripathi, L., Alvi, P. A., & Subramaniam, U, 2021, John Wiley & Sons, ISBN: 978-0367564261.
- R8: F. Jensen, Introduction to Computational Chemistry, 3rd edition, Wiley, 2017.

E-Resources/Web Resources

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

Skill Sets

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

Course Code: CSS2200	Course Title: Program Solving Using C Type of Course: Theory Course	L- T-P-C	2-0-0-2
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		

Course Description	The course is designed to provide a strong foundation in the C programming language and problem-solving techniques. It enables students to develop logical thinking and programming skills required to design and implement applications using C. By understanding fundamental programming concepts such as algorithms, control structures, arrays, functions, pointers, and data handling, students will be prepared to learn and adapt to other programming languages and advanced computing technologies in the future.	
Course Objective	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: CO1: Explain problem-solving techniques using algorithms, pseudo code, and flowcharts for developing programs. [Understand] CO2: Apply fundamental C programming constructs such as data types, operators, decision-making, and looping statements to develop simple programs. [Apply] CO3: Develop programs using arrays and strings to solve computational problems effectively. [Apply] CO4: Implement modular and reusable programs using functions, recursion, and pointers. [Apply] CO5: Design applications using structures, unions, file handling concepts, and demonstrate the role of C programming in emerging technologies. [Apply]	
Course Content:		
Module 1	Introduction to C Language	6 Sessions
Topics: Introduction to Programming – Algorithms – Pseudo Code - Flow Chart – Compilation – Execution – Preprocessor Directives (#define, #include, #undef) - Overview of C – Constants, Variables and Data types – Operators and Expressions – Managing Input and Output Operations – Decision Making and Branching - Decision Making and Looping.		
Module 2	Introduction to Arrays and Strings	6 Sessions
Topics: Arrays: Introduction – One Dimensional Array – Initialization of One Dimensional Arrays – Example Programs – Sorting (Bubble Sort, Selection Sort) – Searching (Linear Search) - Two Dimensional Arrays – Initialization of Two Dimensional Arrays. Example Programs – Matrix operations. Strings: Introduction – Declaring and Initializing String Variables – Reading Strings from Terminal – Writing String to Screen – String Handling Functions.		
Module 3	Functions and Pointers	6 Sessions
Topics: Functions: Introduction – Need for User-defined functions – Elements of User-Defined Functions: declaration, definition and function call–Categories of Functions – Recursion. Pointers: Introduction – Declaring Pointer Variables – Initialization of Variables – Pointer Operators – Pointer Arithmetic – Arrays and Pointers – Parameter Passing: Pass by Value, Pass by Reference.		
Module 4	Structures and Union and File Handling	6 Sessions
Topics: Structures: Introduction – Defining a Structure – Declaring Structure Variable – Accessing Structure Members – Union: Introduction – Defining and Declaring Union – Difference Between Union and Structure.- File handling: Defining and Opening a File – Closing a File – Input / Output Operations on File.		
Module 5	Application of C Programming in Emerging CSE Specializations	6 Sessions
Topics: Applications of C Programming in modern technologies – Cyber Security: Simple password checking programs and basic security concepts – Blockchain: Secure data storage and transaction concepts – Artificial Intelligence and Machine Learning: Basic data processing programs.		
Text Book(s): 1. E. Balaguruswamy, “Programming in ANSI C”, 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.		

Reference Book(s):

1. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020.
2. Reema Thareja, "Programming in C", Oxford University Press, Second Edition, 2016.
3. Kernighan, B.W and Ritchie, D.M, "The C Programming language", Second Edition, Pearson Education, 2015
4. Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014.
5. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.

Web Links and Video Lectures:

1. [NPTEL – Programming in C](#)
2. [NPTEL – Problem Solving Through Programming in C](#)
3. [GeeksforGeeks – C Programming Language](#)

Course Code: ENG2030	Course Title: Communication in Practice Type of Course: Integrated	L-T-P-C	1-0-2-2
Version No.	1.0		
Course Pre-requisites	ENG XXXX – Fundamentals of Communication (Semester I)		
Anti-requisites	NIL		
Course Description	<i>Communication in Practice</i> is the second-semester course in sequence designed for first-year undergraduate students in engineering and computer science programmes. This course puts those foundations under professional pressure. The question is no longer <i>what</i> communication is, but <i>how</i> to deploy it when it counts: in a job interview, a pitch to a client, a negotiation with a colleague, a professional report directed at a real reader, or a leadership communication challenge that affects a team. The course is organised around four modules, each of which explicitly upgrades the work done in semester one. Assessment is 100% Continuous Assessment. There is no end-semester examination. The centrepiece is a capstone Professional Communication Portfolio that the student could realistically share with an employer.		
Course Objective	The course aims to articulate the foundational tenets of professional communicative identity, discourse ethics, and analytical text evaluation and apply them to high-stakes institutional scenarios and complex professional ecosystems. It also focuses on cultivating advanced oral communication strategies and persuasive vocal mechanics, including structured rhetorical delivery and spontaneous adaptive responses, for collaborative, hierarchical, and leadership contexts. Furthermore, it aims to refine technical authorship capabilities and artificial intelligence-mediated workflows to produce precise, highly structured, and audience-centric professional artifacts across formal engineering, computational, and cross-functional corporate domains.		
Course Outcomes	On successful completion of this course the students shall be able to: 1. Explain the principles governing professional communication identity, communicator ethics, and context-sensitive self-presentation across workplace, hierarchical, and digital contexts.		

--	---	--	--	--

- Negotiation and principled disagreement - Communicating in leadership positions: chairing, facilitating, and giving direction - Managing communication under pressure: hostile questions, silence, and contradiction				
Module 4	Adaptive Communication	CA	Communication Strategy Plan	12 Sessions
The topics will be explored through leadership simulations, AI-assisted scenario practice, ethical debate exercises, case analysis, peer feedback sessions, and reflective portfolio activities: - Communication Strategy: Audiences, Channels, Timing, and Feedback Loops - What Emotional Intelligence Actually Does in High-Stakes Conversations - Reading the Room: Adapting Style Across Personalities, Contexts, and Power Gaps - Giving and Receiving Developmental Feedback as a Professional Practice - Navigating Difficult Conversations: Disagreement, Criticism, and Bad News - Repairing Communication Breakdowns: Apology, Accountability, and Recovery - AI as a Strategic Communication Tool: Capabilities, Limits, and What Professionals Must Decide				
Targeted Application & Tools that can be used: Otter.ai, Microsoft Word/PowerPoint/Teams/Clipchamp/Whiteboard/OneNote/Forms/Copilot, Miro, Padlet, Canva, Mentimeter, AllSides Media Bias Ratings, Ad Fontes Media Bias Chart, Snopes/factcheck.rog, Zotero, NotebookLM, Canva, Adobe Express, CapCut, OBS Studio, Claude, ChatGPT, Gemini, DeepSeek, Google Docs				
Project Work / Assignment:				
1. Register Switching Activity (Individual): Students receive a single professional scenario and must communicate the same core message to three different audiences — a peer, a manager, and a non-specialist outsider — switching register, vocabulary, and tone appropriately for each. Delivered live in class. Graded on accuracy of register shift, vocabulary choices, and adaptability under in-the-moment feedback from the assessor. 2. Word in the Wild Log (Individual): Over four weeks, students maintain a running log of 25 words or phrases encountered in professional contexts — podcasts, workplace conversations, news, interviews — that they did not fully understand or could not have used confidently themselves. For each entry, students record the source, the meaning in context, an alternative they might have used, and one original sentence demonstrating precise use. Submitted as a structured document at the end of Module 1. Graded on range, contextual accuracy, and quality of reflection. 3. Rapid Listening and Response (Individual): Students listen to an unedited workplace audio recording — a meeting excerpt, a briefing, or a professional conversation — played once, without transcripts. Under timed conditions, they produce a structured written response that captures key information, identifies the communication purpose, and drafts an appropriate follow-up action note. Graded on accuracy, organisation, and quality of written response. 4. Mock Professional Interview (Individual): Each student undergoes a 10-minute mock professional interview conducted by faculty or a peer panel. No prepared scripts permitted. Graded on composure, structure, vocal authority, and use of the STAR framework. Written self-evaluation of 200 words submitted within 48 hours. 5. Idea Pitch (Individual): Students deliver a 3-minute pitch of a solution to a real or simulated professional problem, to a panel that includes at least one hostile or sceptical questioner. Immediately after the pitch, the				

panel asks two minutes of unscripted questions. Graded on clarity of argument, persuasive structure, handling of pressure, and vocal presence.

6. Difficult Conversation Simulation (Pair): Pairs of students are assigned roles in a scripted conflict scenario — a performance issue, a missed deadline, a broken professional agreement. Each student must navigate the conversation toward a constructive resolution without a script. Immediately after, both students complete a structured individual debrief identifying what communication choices they made, what they would change, and what they observed in their partner. Graded individually on communication behaviour during the simulation and quality of the debrief.

7. Communication Strategy Plan — Presentation and Portfolio Submission (Group): Students select a real or realistic professional scenario — a team conflict, a stakeholder disagreement, a leadership announcement, or a crisis communication moment. They develop a communication strategy plan that maps the audience, chooses appropriate channels and timing, anticipates likely responses, and identifies where AI tools could assist and where they must not. The plan is delivered as a 5-minute structured presentation to a simulated panel, followed by two minutes of unscripted Q&A. The portfolio submission includes the written plan, a one-page reflection on their communication growth across all four modules, and one piece of evidence from each module that demonstrates that growth. Graded on strategic thinking, clarity of presentation, quality of reflection, and coherence of the portfolio as a whole.

Text Book(s):

- T1. Treasure, J. (2017). *How to be Heard: Secrets for Powerful Speaking and Listening*. Mango Publishing.
T2. Beebe, S. A., Beebe, S. J., & Ivy, D. K. (2019). *Communication: Principles for a Lifetime* (7th ed.). Pearson.

Reference Book(s):

- R1. Goleman, D. (1995). *Emotional Intelligence: Why It Can Matter More Than IQ*. Bantam Books.
- R2. Patterson, K., Grenny, J., McMillan, R., & Switzler, A. (2011). *Crucial Conversations: Tools for Talking When Stakes Are High* (2nd ed.). McGraw-Hill.
- R3. Cuddy, A. (2015). *Presence: Bringing Your Boldest Self to Your Biggest Challenges*. Little, Brown and Company.
- R4. Fisher, R., Ury, W., & Patton, B. (2011). *Getting to Yes: Negotiating Agreement Without Giving In* (3rd ed.). Penguin Books.
- R5. Stone, D., Patton, B., & Heen, S. (2010). *Difficult Conversations: How to Discuss What Matters Most* (2nd ed.). Penguin Books.

Web Sources:

- [ted.com/topics/communication](https://www.ted.com/topics/communication)
- [ted.com/topics/leadership](https://www.ted.com/topics/leadership)
- hbr.org (Harvard Business Review — communication and leadership articles)
- psychologytoday.com/us/basics/emotional-intelligence
- learnprompting.org
- unesco.org/en/artificial-intelligence/ethics
- yoodli.ai (AI speech coaching)
- themuse.com (professional communication and career resources)

- mindtools.com (communication skills and workplace tools)

Topics relevant to "SKILL DEVELOPMENT": Vocabulary building and register awareness; precise and concise language use; explaining technical terms to non-specialist audiences; active listening and information processing; converting spoken inputs into structured written responses; writing action notes and follow-up communications; professional presence and vocal authority; interview communication and the STAR framework; idea pitching and persuasive speaking; negotiation and principled disagreement; chairing and facilitating discussions; managing communication under pressure; communication strategy and audience mapping; emotional intelligence in professional contexts; adapting communication style across personalities and power gaps; giving and receiving developmental feedback; navigating difficult conversations; repairing communication breakdowns; AI as a strategic communication tool; ethics of AI in professional communication.

Catalogue prepared by	Dr Ajith K and Dr Shahanas Punnilath Shanavas
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: EEE1200	Course Title: Basics of Electrical and Electronics Engineering. Type of Course: Professional Core - Theory	L-T-P-C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This is a fundamental Course which is designed to know the use of basics of electrical and electronics engineering principles occurs in various fields of Engineering. The course emphasises on the characteristics and applications of electrical and electronic devices. The course also emphasizes on the working, analysis and design of electrical circuits using both active & passive components. Additionally, this course creates a foundation for the future courses such as Electrical machines, power system, power electronics Linear Integrated Circuits, Analog Communication and Digital Communication etc.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Basics of Electrical and Electronics Engineering and attain Skill Development through Participative Learning techniques.		
Course Outcomes	On successful completion of this course the students shall be able to:		

	Apply 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. Discuss the performance characteristics and applications of various electrical Machines.			
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, Nodal Analysis, Numerical examples. AC Circuits: Fundamentals of single phase circuits - Series RL, RC and R-L-C Circuits, Concept of active power, reactive power and Power factor, Numerical examples. Introduction to three phase system and relation between line and phase values in Star & Delta connection, Numerical examples.				
Module 2	Semiconductor and Diode applications	Assignment/ Quiz	Memory Recall based Quizzes	10 Sessions
Mass Action Law, Charge densities in a semiconductor, Types of SC, Junction diodes -Ideal and practical behaviour, Modelling the Diode Forward Characteristic, and Diode applications like rectifiers, Clipping and clamping circuits. Zener diode, characteristics and its applications like voltage regulator.				
Module 3	Transistors and its Applications	Assignment/ Quiz	Memory Recall-based Quizzes	10 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.				
Module 4	Fundamentals of Electrical Machines	Assignment/ Quiz	Numerical solving Task	10 Sessions
Electrical Machines: Single phase transformers: principle of operation and EMF equation, Numerical examples. DC Motor: principle of operation, Back EMF, torque equation, Numerical examples. AC Motor: Principle operation of Induction Motors and its Applications. Special Machines: Introduction to special electrical machines and its applications.				
Targeted Application & Tools that can be used:				

Targeted Applications: Application Area includes all electrical and electronic circuits (power supply unit, regulator unit, embedded devices, hardware electronics etc.). The students will be able to join a profession which involves basics to high level of electronic circuit design. Professionally Used Software: Multisim/ P Spice Besides these software tools hardware equipment such as Multimeters, Function Generators, Power Supplies, Oscilloscopes etc., can be used to perform component/circuit testing and analysis..
Project Work/ Assignment:
1. Article review: At the end, of course an article topic will be given to an individual or a group of students. They need to refer the library resources and write a report on their understanding about the assigned article in appropriate format. 2. Presentation: There will be a group presentation, where the students will be given a topic. They will have to explain/demonstrate the working and discuss the applications for the same. 3. Case Study: - At the end of the course students will be given a 'real-world' application based circuits like Power Amplifier, Signal/Function Generator etc. as a case study. Students will be submitting a report which will include Circuit Diagrams, Design, Working Mechanism and Results etc. in appropriate format
Text Book(s): T1: Kothari D. P. & Nagrath I. J., "Basic Electrical and Electronics Engineering", Tata McGraw-Hill Education T2: 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. T3: A.P.Malvino, Electronic Principles, 7th Edition, Tata McGraw Hill, 2007 T4: J. Millman, C. C. Halkias and C. D. Parikh, "Millman's Integrated Electronics", McGraw Hill Education, 2nd Edition. T5: Basics of Electrical & Electronics Laboratory Manual.
Reference Book (s): R1: John Hiley, Keith Brown and Ian McKenzie Smith, "HUGHES Electrical and Electronic Technology", 10th Edition (Indian Edition published by Dorling Kindersley), Pearson, 2011 R2: Samarajit Ghosh, "Fundamentals of Electrical and Electronics Engineering", 2nd Edition, Prentice Hall India, 2007. R3: K Uma Rao, A Jaya Lakshmi, "Basic Electrical engineering" IK International publishing house Pvt. Ltd R4: R. L. Boylestad and L. Nashelsky, "Electronic Devices and Circuit Theory", Pearson Education India 7th Edition. R5: A K. Maini, V. Agrawal, "Electronic Devices & Circuits", Wiley, 2nd Edition R6: A.S Sedra, K. C. Smith, "Microelectronic Circuits", Oxford University Press, 6th Edition Online Resources (e-books, notes, ppts, video lectures etc.): 1. https://presidencyuniversity.linways.com

2. <https://www.digimat.in/nptel/courses/video/108105112/L01> "Fundamentals of Electrical Engineering-Basic Concepts, Examples"
3. Seminar Topic: <https://nptel.ac.in/courses/108/105/108105153/> "Electrical Measurements"
4. Video lectures on "Electronic Devices" by Prof. Dr. A. N. Chandorkar, IIT Bombay
<http://www.satishkashyap.com/2013/03/video-lectures-on-electron-devices-by.html>
5. Video lectures on "Analog Electronics" by Prof. S.C. Dutta Roy, IIT Delhi
<https://nptel.ac.in/courses/108/102/108102095/>
6. Video lectures on "Diodes", by Prof. Chitralkha Mahanta, IIT Guwahati,
7. <https://nptel.ac.in/courses/117/103/117103063/>

E-content:

- W1: "Introduction to Electrical Machines <https://nptel.ac.in/courses/108/102/108102146/>"
- W2: M. -Y. Kao, H. Kam and C. Hu, "Deep-Learning-Assisted Physics-Driven MOSFET Current Voltage Modeling," in IEEE Electron Device Letters, vol. 43, no. 6, pp. 974-977, June 2022, doi: 10.1109/LED.2022.3168243
- W3: <https://ieeexplore-ieee-org-resiuniv.knimbus.com/document/9758727>
- W4: F. Bonet, O. Aviñó-Salvadó, M. Vellvehi, X. Jordà, P. Godignon and X. Perpiñà, "Carrier
- W5: Concentration Analysis in 1.2 kV SiC Schottky Diodes Under Current Crowding," in IEEE Electron Device Letters, vol. 43, no. 6, pp. 938-941, June 2022, doi: 10.1109/LED.2022.3171112.
<https://ieeexplore-ieeeorg-presiuniv.knimbus.com/document/9764749>
- W6: 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.
- W7: <https://ieeexplore.ieee.org/document/7018053>
- W8: R. Raut and O. Ghasemi, "A power efficient wide band trans-impedance amplifier in submicron
- W9: CMOS integrated circuit technology," 2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference, 2008, pp. 113-116, doi: 10.1109/NEWCAS.2008.4606334.
<https://ieeexplore.ieee.org/document/4606334>

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

Course Code: CSS2201	Course Title: Program Solving Using C Lab Type of Course: Laboratory Course	L- T-P-C	0-0-4-2
Version No.	1.0		

Course Pre-requisites	NIL
Anti-requisites	NIL
Course Description	The course is designed to provide a strong foundation in the C programming language and problem-solving techniques. It enables students to develop logical thinking and programming skills required to design and implement applications using C. By understanding fundamental programming concepts such as algorithms, control structures, arrays, functions, pointers, and data handling, students will be prepared to learn and adapt to other programming languages and advanced computing technologies in the future.
Course Objective	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: CO1: Explain algorithms, flowcharts, and basic programming concepts used in C programming. [Understand] CO2: Implement C programs using decision-making, looping statements, arrays, and strings to solve computational problems. [Apply] CO3: Construct modular programs using functions, recursion, and pointers for effective problem solving. [Apply] CO4: Develop applications using structures, unions, and file handling concepts in C programming. [Apply] CO5: Design and implement simple real-world applications in areas such as Cyber Security and Artificial Intelligence using C programming concepts. [Apply]
Course Content:	
List of Practicals: 60 Sessions Lab Sheet 1: 10 Sessions Program 1: Sum of Two Numbers Program 2: Find the Greatest of Three Numbers Program 3: Check Even or Odd using Conditional Operator Program 4: Print Multiplication Table using Loop Program 5: Count Digits in a Number using While Loop Program 6: Demonstration of Preprocessor Directives Program 7: Simple Calculator using Switch Case Lab Sheet 2: 10 Sessions Program 1: Check Whether a Number is Positive, Negative or Zero Program 2: Find the Sum of First N Natural Numbers Program 3: Check Whether a Number is Prime or Not Program 4: Find Factorial of a Number Program 5: Reverse a Number Program 6: Simple Number Guessing Game Lab Sheet 3: 10 Sessions Program 1: Linear Search in a One-Dimensional Array Program 2: Bubble Sort on an Integer Array Program 3: Matrix Addition (2D Arrays) Program 4: Count Vowels in a String Program 6: Selection Sort on an Array Lab Sheet 4: 10 Sessions	

Program 1: Sum of Two Numbers Using User-Defined Function Program 2: Factorial Using Recursion Program 3: Swap Two Numbers Using Call by Value (No Swap) Program 4: Swap Two Numbers Using Call by Reference (With Swap) Program 5: Pointer Basics - Access and Modify Variable via Pointer Program 6: Accessing Array Elements Using Pointers Lab Sheet 5: 10 Sessions Program 1: Basic Structure Usage Program 2: Structure with Nested Structures (Date of Birth) Program 3: Union Example and Member Access Program 4: Write to a File (Text Mode) Program 5: Read from a File (Text Mode) Program 6: Append Data to a File Lab Sheet 6: 10 Sessions Program 1: Write a C program to check whether a password is strong or weak based on its length. If the password contains more than 8 characters, display "Strong Password"; otherwise display "Weak Password". Program 2: Develop a simple login system using C. Store a predefined username and password in the program and verify the user input. Display "Login Successful" or "Invalid Credentials". Program 3: Write a C program to input marks of 5 students and calculate the average marks. Display whether the class performance is "Good" or "Needs Improvement". Program 4: Develop a C program to find the largest number in an array. This simulates basic data analysis used in AI systems.			
Text Book(s): 1. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.			
Reference Book(s): 1. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020. 2. ReemaThareja, "Programming in C", Oxford University Press, Second Edition, 2016. 3. Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015 4. Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014. 5. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.			
Web Links and Video Lectures: 1. NPTEL – Programming in C 2. NPTEL – Problem Solving Through Programming in C 3. GeeksforGeeks – C Programming Language 4. Programiz – Learn C Programming			

Course Code: CHE2502	Course Name: Chemistry of Smart Materials Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		

Anti-requisites	NIL		
Course Description	The laboratory course aims to develop experimental skills and apply fundamental chemical principles to address chemistry-related problems in engineering. The experiments are carefully designed to complement the theoretical concepts covered in lectures, providing hands-on experience to deepen understanding and reinforce learning. This course is designed to cater to Environment and Sustainability.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Chemistry of Smart Materials Lab” and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques.		
Course Outcomes	On successful completion of the course, students shall be able to: CO1: Recognize the basic techniques and instrumentation [Level 3 - Apply] used in chemistry laboratories for quantitative analysis. CO2: Estimate the presence of acids and metal ions in [Level 3 – Apply] domestic and industrial waste using laboratory techniques. CO3: Review the experimental results and demonstrate [Level 3 – Apply] improved experimental skills through hands-on laboratory experience. CO4: Classify laboratory techniques such as experimental [Level 3 – Apply] setups for synthesis, purification, recovery and analysis.		
Course Content:			
Experiment 1	Experimental	Data Collection	Analysis and Interpretation
Determination of strength of strong acid in battery electrolyte using conductometric sensors.			
Experiment 2	Experimental	Data Collection	Analysis and Interpretation
Estimation of iron from e-waste using Electrochemical sensors.			
Experiment 3	Experimental	Data Collection	Analysis and Interpretation
Determination of pKa of organic acid of battery electrolyte using pH sensor.			
Experiment 4	Experimental	Data Collection	Analysis and Interpretation
Estimation of copper from PCBs by using colorimeter (Optical Sensor).			

Course Code: PPS1026	Course Title: Industry Readiness Program – II Type of Course: Practical Only Course	L- T - P- C	0-0-2-0
Version No.	1.0		
Course Pre- requisites	NIL		
Anti-requisites	NIL		
Course Description	This course is designed to enable students learn styles of communication, team building and use empathy in leadership. The course will benefit learners in preparing themselves effectively through various activities and learning methodologies.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Industry Readiness for Young Professionals” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.		
Course Out Comes	On successful completion of this course the students shall be able to: CO 1 Apply different communication skills for success in workplace CO 2 Practice team building skills for career success CO3 Demonstrate ethical leadership skills in workplace		
Course Content			
Module 1	Effective Communication	Classroom activities	10 Hours
Topics: Practice effective communication skills (Verbal, Non-verbal, Written and Visual)			
Activity: Use social media prompts to prepare self-introduction videos			
Module 2	Team Building	Group Activity	10 Hours
Topics: Skills of an effective team player			
Activity: Student group activity to build class networking			
Module 3	Leadership	Case study	10 Hours
Topics: Types of leadership, using empathy in leadership			
Activity: Individual presentation by students on corporate leaders.			

Faculty : L&D
Targeted Application & Tools that can be used: TED Talks You Tube Links Activities
Assignment proposed for this course Assignment 1: One minute reel Assignment 2: Team building assignment
Continuous Individual Assessment Module 1: L-S-R-W class assessment Module 2: Team Presentation Module 3: Individual Assessment
The topics related to skill development: Students acquire knowledge on effective communication skills, team building skills and how to prepare themselves to be leaders in workplace using empathy and implement various skill sets during the course of their time in the university.

Course Code: EEE1251	Course Title: Basics of Electrical and Electronics Engineering Laboratory Type of Course: ESC - Laboratory	L-T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	Nil		
Anti-requisites	Nil		
Course Description	This fundamental laboratory provides an opportunity to validate the concepts taught in the basics of electrical and electronics engineering and enhances the ability to visualize real system performance, using both hardware and simulation tools.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Basics of Electrical and Electronics Engineering and attain Skill Development through Experiential Learning techniques.		

Basic skill sets required for the laboratory	The students shall be able to develop: 1) An attitude of enquiry. 2) Confidence and ability to tackle new problems. 3) Ability to interpret events and results. 4) Ability to work as a leader and as a member of team. 5) Assess errors and eliminate them. 6) Observe and measure physical phenomenon. 7) Write Reports. 8) Select suitable equipment, instruments and materials. 9) Locate faults in systems. 10) Skills for setting and handling equipment. 11) The ability to follow standard test procedures. 12) An awareness of the need to observe safety precautions. 13) To judge magnitudes without actual measurement.
Course Outcomes	On successful completion of the course, the students shall be able to: 1. Demonstrate the behavior of passive elements and fundamental electrical circuit laws 2. Determine the power factor, impedance, and phase relationship of R and RL loads under AC supply conditions. 3. Develop electrical circuits and three-phase supply systems using MATLAB/Simulink to study resistance, voltage, and current characteristics. 4. Analyze the performance characteristics of electrical machines, including single-phase transformers and three-phase induction motors, under different operating conditions. 5. Examine the output characteristics and performance parameters of half-wave and full-wave rectifiers. 6. Illustrate the performance characteristics of Zener diode and MOSFET under different operating conditions.
Course Content:	List of Laboratory Tasks: <u>Experiment No 1:</u> Familiarization of passive elements. Level 1: Identify and understand the function and ratings of basic passive elements. Level 2: Study the behavior of passive elements in simple electrical circuits by measuring voltage, current, and impedance. <u>Experiment No 2:</u> Experimental Verification of Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL). Level 1: Study and Verify KVL and KCL for the given electrical Circuit. <u>Experiment No 3:</u> Determination of Power Factor, and Impedance for R and RL Loads. Level 1: Determine the power factor and impedance of R and RL loads under steady-state AC supply conditions. Level 2: Analyze the effect of inductance on power factor, impedance, and phase angle for RL loads under varying load conditions. <u>Experiment No 4:</u> Modeling and Simulation of effective resistances in different circuit connections Using Simulink. Level 1: Model and simulate series and parallel resistor circuits in Simulink and determine their effective resistance. Level 2: Analyze the effect of different resistor combinations on total resistance, current distribution, and voltage drop.

	<u>Experiment No 5:</u> Simulation and analysis of line and phase voltages in Star-connected and Delta- connected supply systems. Level 1: Model and simulate star-connected and delta-connected three-phase supply systems and observe line and phase voltages. Level 2: Analyze and compare the relationship between line and phase voltages in star and delta connections. <u>Experiment 6:</u> Determination of transformation ratio of a Single-Phase Transformer Under different tapping conditions. Level 1: Determine the transformation ratio of a single-phase transformer under no-load condition for various tap settings. Level 2: Analyze the effect of different tapping conditions on secondary voltage, regulation, and transformer performance. <u>Experiment No. 7:</u> Study of the speed-torque characteristics of a Three-Phase Induction Motor. Level 1: Study and analyze the speed–torque characteristics of a three-phase induction motor. Level 2: Evaluate the effect of supply voltage and current on motor performance parameters such as slip, and torque. <u>Experiment 8:</u> Performance Analysis of Half-Wave and Full-Wave Rectifier Circuits Using Simulink. Level 1: Model and simulate half-wave and full-wave rectifier circuits in Simulink and analyze their output waveforms. Level 2: Evaluate and compare rectifier performance parameters such as ripple factor, efficiency, and output voltage. <u>Experiment 9:</u> Study of the characteristics of a Zener Diode. Level 1: Study the forward and reverse V–I characteristics of a Zener diode using DC analysis. Level 2: Analyze the Zener breakdown characteristics and determine the Zener voltage and regulation behavior. <u>Experiment 10:</u> Study of the Output and Transfer Characteristics of a MOSFET. Level 1: Study, analyze and implement the output characteristics of a MOSFET under different gate-source voltage conditions and observe their results. Level 2: Analyze the transfer characteristics of a MOSFET and evaluate key parameters such as threshold voltage and transconductance.
	Targeted Application & Tools that can be used: Targeted Applications: Application area includes all electrical and electronic circuits (power supply unit, regulator unit, embedded devices, hardware electronics etc.). The students will be able to join a profession which involves basics to high level of electronic circuit design. Professionally Used Software: MATLAB SIMULINK Besides these software tools hardware equipment such as Multimeters, Function Generators, Power Supplies, Oscilloscopes etc., can be used to perform component/circuit testing and analysis.
	Course Material 1. Basics of Electrical and Electronics Engineering Laboratory Manual, Presidency University, Bengaluru.
	Text Book: 1. D. P. Kothari and I. J. Nagrath, Basic Electrical Engineering, 4th Edition, McGraw Hill Education, New Delhi, 2019. 2. B. L. Theraja and A. K. Theraja, A Textbook of Electrical Technology, Volume-I: Basic Electrical Engineering in SI Units, 24th Edition, S. Chand Publishing, New Delhi, 2024.
	References 1. John Hiley, Keith Brown and Ian McKenzie-Smith, Hughes Electrical and Electronic Technology, 12th Edition, Pearson Education, 2016. 2. S. K. Bhattacharya and D. Devaraj, Basic Electrical and Electronics Engineering, 2nd Edition,

Pearson Education India, 2017. 3. T. K. Nagsarkar and M. S. Sukhija, Principles of Basic Electrical Engineering, 1st Edition, Oxford University Press, 2018. 4. Robert L. Boylestad and Louis Nashelsky, Electronic Devices and Circuit Theory, 12th Edition, Pearson Education, 2019. 5. A. K. Maini and Varsha Agrawal, Electronic Devices and Circuits, 2nd Edition, Wiley India, 2019. 6. Adel S. Sedra and Kenneth C. Smith, Microelectronic Circuits, 8th Edition, Oxford University Press, 2019.	
Online Resources: 1. https://presidencyuniversity.linways.com 2. https://www.digimat.in/nptel/courses/video/108105112/L01 "Fundamentals of Electrical Engineering - Basic Concepts, Examples" 3. Video lectures on "Electronic Devices" by Prof. Dr. A. N. Chandorkar, IIT Bombay http://www.satishkashyap.com/2013/03/video-lectures-on-electron-devices-by.html 4. NPTEL lectures on "Rectifiers", by Prof. G. Bhuvaneswari, IIT Delhi https://nptel.ac.in/courses/108102145 5. NPTEL Lectures on "Electrical Machines II", by Prof. G. Sridhara Rao, IIT Madras. https://nptel.ac.in/courses/108106072	
Topics relevant to "EMPLOYABILITY SKILLS": All the experiments which are listed are for Skill Development through Experiential Learning Techniques. This is attained through the assessment component mentioned in course handout. Topics relevant to "HUMAN VALUES & PROFESSIONAL ETHICS": Standard test methods, safety procedure.	
Catalogue prepared by	
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: MAT1211	Course Title: Linear Algebra, Differential Equations and Statistical Methods Type of Course: 1] School Core	L-T- P- C	3-1-0-4
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces fundamental mathematical concepts from linear algebra, differential calculus, ordinary differential equations and statistics. The content includes matrices, systems of linear equations, eigenvalues and eigenvectors, functions of several variables, differential equations, correlation and regression analysis, and curve fitting techniques. The course combines theoretical developments with illustrative applications drawn from scientific and engineering settings. The course emphasizes the application of these mathematical tools in solving problems in science and engineering. It extends student's problem-solving abilities through analytical and computational approaches, with focus on both theoretical understanding and applications.		

Course Objective	The objective of this course is to provide students with a strong mathematical foundation required for engineering and scientific studies. The course introduces fundamental concepts from linear algebra, differential calculus, ordinary differential equations, and statistical methods that are widely used in modeling, analysis and problem solving. It aims to familiarize students with mathematical techniques for representing engineering systems, studying relationships among variables, describing dynamic processes and interpreting data. The course serves as a foundation to prepare students for advanced engineering subjects that rely on mathematical modeling, computation and analytical methods.	
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Use linear algebra concepts to evaluate properties of matrices and quadratic forms. (L3) CO2: Apply matrix-based methods to solve systems of linear equations. (L3) CO3: Use fundamental theorems and analytical techniques of differential calculus to single variable functions. (L3) CO4: Apply the concepts of partial derivatives to multivariable functions. (L3) CO5: Solve first order ordinary differential equations using analytical methods. (L3) CO6: Employ statistical techniques for data analysis and to interpret relationships between variables. (L3)	
Course Content:		
Module 1	Fundamentals of Linear Algebra	(15L+5T=20 Sessions)
Matrices, echelon forms, rank of matrix, Systems of linear equations: Consistency, solution to system of linear homogeneous and non-homogeneous equations- Gauss elimination method, Gauss-Jordan method (Self-study), LU decomposition method, Network analysis using linear systems, Design of traffic patterns, Electrical circuits. Eigenvalue and eigenvector, diagonalization of symmetric matrices, quadratic forms.		
Module 2	Differential Calculus	(15L+5T=20 Sessions)
Introduction, limit, continuity, Rolle's theorem, mean value theorems, Indeterminate forms and L'Hôpital's rule. Partial derivatives, Euler's theorem, total derivative, Jacobians, Taylor's and Maclaurin's theorems, maxima, minima and saddle points, method of Lagrange multipliers, Current Growth in an RL Circuit, Velocity Estimation, Fuel consumption optimization, Cost function optimization.		
Module 3	Ordinary Differential Equations	(6L+2T=8 Sessions)
Definition, types of differential equations-variable separable, homogeneous, exact, linear and Bernoulli's equations, Equations not of first degree: equations solvable for p , equations solvable for y , equations solvable for x and Clairaut's type, Newton's law of cooling, Current in RL circuit.		
Module 4	Statistical Methods and Curve Fitting	(9L+3T=12 Sessions)
Measures of central tendency, moments, skewness, and kurtosis. Correlation - Karl Pearson's coefficient of correlation, rank correlation, multiple correlation. Regression analysis - lines of regression, multiple regression. Principle of least square, curve fitting- straight line, parabola and exponential curves, Stress and strain curve fitting, Feature selection and Prediction models.		
Targeted Applications & Tools that can be used: - Linear Algebra is applied to solve systems of equations in circuits, traffic flow, and network analysis, with eigenvalues and quadratic forms aiding stability and optimization studies. - Differential Calculus supports process speed analysis, velocity estimation, and optimization techniques such as minimizing fuel consumption and cost functions. - Differential Equations model dynamic behaviours like current in RL circuits, heat transfer in materials, and growth or decay in environmental and nuclear engineering. - Statistical Methods and Curve Fitting enable effective data analysis through correlation, regression, and curve fitting, supporting predictive modeling and interpretation of experimental data such as stress-strain relationships.		
Software: MATLAB, Python, R Programming, Mathematica, Scilab, Octave		
Assignment:		

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

Text Book

1. B. S. Grewal, *Higher Engineering Mathematics*, 44th ed. New Delhi, India: Khanna Publishers, 2017.
2. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2018.

References:

1. P. V. O'Neil, *Advanced Engineering Mathematics*, 8th ed. Boston, MA, USA: Cengage Learning, 2018.
2. R. I. Miller and J. E. Freund, *Probability and Statistics for Engineers*, 9th ed. Noida, India: Pearson Education Ltd., 2017.
3. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
4. V. Henner, T. Belozero, and M. Khenner, *Ordinary and Partial Differential Equations*. Boca Raton, FL, USA: CRC Press, 2013.

E-resources/ Web links:

1. <https://nptel.ac.in/courses/111105121>
2. <https://nptel.ac.in/courses/111106146>
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_106839
4. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_61605
5. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_134719
6. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=TEXTBOOK_LIBRARY01_06082022_174

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

Catalogue prepared by	Dr. Kavita Permi, Dr. Rahul M. P. and Dr. Athmakoori Prashant
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: PHY2501	Course Title: Optoelectronics and Quantum Physics Type of Course: School Core	L-T-P-C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		

Course Description	The purpose of this course is to enable the students to understand the fundamentals, working and applications of optoelectronic devices and to develop the basic abilities to appreciate the applications of advanced microscopy and quantum computers. The course develops the critical thinking and analytical skills.		
Course Out Comes	On successful completion of the course the students shall be able to: CO1: To understand the concepts of electrical conducting properties of metal, semiconductor and superconductivity. CO2: To understand the principles of quantum mechanics. CO3: Discuss the quantum concepts used in quantum computers. CO4: Explain the applications of lasers and optical fibers in various technological fields.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Applied Physics for Computer Science Cluster “and to attain the basic knowledge related to quantum mechanics and computation.		
Course Content:			
Module 1	Electrical Conductivity Of Solids And Semiconducting Devices	Assignment	11 Sessions
Topics: Classification of materials based on bandgap, Fermi energy and Fermi level, Fermi level in semiconductors, Law of mass action, Electrical conductivity of a semiconductor, Hall effect, Superconductivity, p-n junctions, Zener diode, Solar cells, I-V characteristics, and LEDs			
Module 2	Quantum Mechanics	Assignment	11 Sessions
Topics: Introduction, de-Broglie hypothesis, Heisenberg’s uncertainty principle- statement and physical significance. Wave function-properties and physical significance. Schrodindger’s time independent wave equation, Probability density and normalization of wave function. Wave Function in Ket Notation: Matrix form of wave function, Identity operator, Determination of $ 0\rangle$ and $ 1\rangle$, Pauli Matrices and its operations on 0 and 1 states, Mention of Conjugate and Transpose, Unitary Matrix U, Examples: 2x2 Matrices and their multiplication (Inner Product), Probability, Orthogonality			
Module 3	Quantum Computing	Term paper	12 Sessions
Topics: Introduction to quantum computing, Moore’s law & its end, Differences between classical and quantum computing, Concept of Qubit and its properties, . Representation of qubit by Bloch sphere, Quantum Gates: Single Qubit Gates: Quantum Not Gate, Pauli Z Gate, Hadamard Gate, Phase Gate (or S Gate), T Gate. Multiple Qubit Gates: Controlled gate - CNOT Gate, (Discussion for 4 different input states). Representation of Swap gate, Controlled - Z gate, Toffoli gate. Problems.			
Module 4	Lasers And Optical Fibers	Term paper	11 Sessions
Topics: Interactions of radiations with matter, expression for energy density of a system under thermal equilibrium in terms of Einstein’s coefficients, conditions for LASER action using Einstein’s coefficients, Characteristics of laser, conditions and requisites of laser, Principle of optical fibers, Numerical aperture and acceptance angle (Qualitative), Attenuation, Applications: Point to point communication with block diagram, application of optical fibers in endoscopy.			
Targeted Application & Tools that can be used:			

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 T1: Engineering Physics by Avadhanalu, Revised edition, S. Chand Publications, 2024. T2: Quantum Computation and Quantum Information, Michael A. Nielsen & Isaac L. Chuang, Cambridge Universities Press, 2010 Edition
References: 1. Elementary Solid state Physics: Principles and Applications by M.A. Omar, 1st Edition, Pearson Publications, 2002. 2. Principles of Quantum Mechanics by R Shankar, 2nd edition, springer Publications, 2011 3. Optoelectronics: An Introduction by John Wilson and John Hawkes, 3rd edition, Pearson Publications, 2017. 4. Engineering Physics by Gaur and Gupta, Dhanpat Rai Publications, 2012. 5. Introduction to Quantum Mechanics, David J Griffiths, Cambridge University Press, 2019
E-Resources: 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: School Core & Theory Only	L-T-P-C	2-0-0-2	
Version No.	1.2			
Course Pre-Requisites	NIL			
Anti-requisites	NIL			
Course Description	The course is designed with the objective of giving an overview of engineering graphics. It is introductory in nature and acquaints the students with the techniques used to create engineering drawings. The course emphasizes on projection of points, lines, planes and solids and isometric projections.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Engineering Graphics” and attain SKILL DEVELOPMENT through Problem solving methodologies.			
Course Outcomes	On successful completion of this course the students shall be able to: CO1: Demonstrate competency in Engineering Graphics as per BIS conventions and standards. Level 3 – Apply CO2: Comprehend the theory of projection for drawing projections of points, lines, and planes under different conditions. Level 2 – Understand CO3: Prepare multiview orthographic projections of solids by visualizing them in different positions. Level 3 – Apply CO4: Prepare pictorial drawings using the principles of isometric projections to visualize objects in three dimensions. Level 3 – Apply			
Course Content:				
Module 1	Introduction to Drawing	Assignment	Standard technical drawing	7 Sessions
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	7 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	8 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: ENG1030	Course Title: Fundamentals of Communication Type of Course: Integrated	L-T-P-C	1-0-2-2
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	Fundamentals of Communication is a lab intensive, practice first integrated course designed to introduce first-year undergraduate engineering and computer science students to the foundational principles and advanced applications of effective communication across interpersonal, spoken, written, and digital contexts. Anchored in real world scenarios directly relevant to college life, cross-functional technical teams, and digital workplaces, the curriculum develops students' abilities to navigate cultural differences, speak with clarity, read texts critically, and write with precise audience awareness. The course explicitly relies on scenario-based learning, interactive case studies, and hands-on lab applications to ensure that theoretical communication frameworks are immediately actionable. By guiding students from reflective self-awareness to complex digital and AI-mediated content creation, the course builds an enduring communicative foundation optimized for performance-oriented professional practice.		
Course Objective	The course aims to explain the key principles of interpersonal, intercultural, and professional communication and apply them to real-world scenarios. It also focuses on developing spoken communication strategies, including structured delivery and impromptu responses, for academic and group contexts. Furthermore, it aims to develop effective writing skills to produce clear, coherent, and audience-appropriate texts across formal and professional contexts.		
Course Outcomes	On successful completion of this course the students shall be able to: 6. Understand the principles of interpersonal, intercultural, and professional communication in academic, social, and technical team contexts. 7. Demonstrate audience-centered spoken communication in individual presentations, group discussions, and simulated professional scenarios through vocal control, adaptive register, and collaborative engagement. 8. Interpret written, multimodal, and AI-generated texts to identify underlying assumptions, rhetorical strategies, logical errors, and credibility markers across diverse media and professional contexts. 9. Utilize plain-language principles and professional conventions to produce clear, purposeful, and appropriately formatted written and multimodal communication, including technical content adapted for non-specialist audiences. 10. Employ responsible and critical practices in AI-assisted communication by designing iterative prompts, evaluating generated outputs, and applying ethical frameworks in academic and professional contexts.		
Course Content: Fundamentals of Communication			

Module 1	Communication in a New World	CA1 + Project Milestone	Deep Listening and Contextual Response Lab	06 Sessions
The topics will be discussed through case studies: • The Communication Process: How Messages Are Sent, Received, and Misread • Verbal, Non-Verbal, and Vocal Communication in Everyday Contexts • Communication Breakdown: Recognising Barriers Across Psychological, Cultural, and Semantic Dimensions • Active Listening: Attention, Paraphrasing, and Purposeful Response; asking for clarification • Interpersonal Communication and Self-Awareness (<i>including cross-cultural dimensions</i>)				
Module 2	Contemporary Speaking Practices	CA2, CA3, CA4	Narrative and Adaptive Oral Delivery Facilitated Stakeholder Group Discussion Simulated Workplace Crisis Briefing	15 Sessions
The topics will be discussed through structured, individual presentations and group discussions: • Elements of Spoken Communication: Voice, Clarity, and Presence • Storytelling and Narrative in Spoken Contexts • Structuring Spoken Messages: From Idea to Delivery • Speaking with Confidence: Managing Anxiety and Building Presence • Communicating in Groups: Participation, Productive Disagreement, and Turn-Taking • Giving and Receiving Feedback Effectively • Presentation Design and Delivery: Structure, Evidence, and Audience Engagement; eliciting responses • Speaking for Technical and Non-Technical Audiences				
Module 3	Critical Reading	Project Milestone	Source Audit and Fallacy Brief	09 Sessions
The topics will be discussed through case studies and real-world scenarios: • Critical Reading: Purpose, Context, and Analytical Approach • Perspective and Persuasion: Identifying Tone, Bias, and the Fact-Opinion Distinction • Understanding Flawed Reasoning (<i>patterns of weak argumentation in everyday contexts</i>) • Evaluating Sources: Credibility, Reliability, and AI-Generated Content; accuracy of the content • Reading Multimodal Texts: Images, Infographics, and Digital Media				
Module 4	Writing with Purpose	CA 5 + Project Milestone	Technical Translation and Annotated Script	09 Sessions
The topics will be discussed through case studies: • Principles of Effective Writing: Clarity, Coherence, Precision and Concise; audience analysis • Plain Language and Technical Communication: Making Complex Ideas Accessible • Writing for Professional Contexts: Email, Formal Reports, and Correspondence • Argumentative Writing: Claim, Evidence, and Logical Progression				

Summary Writing and Short-Form Academic Response				
Module 5	Communication in the Age of AI	CA 6 – Project Milestone	CA 6: Cross-Disciplinary Case Analysis and Defence	06 Sessions
The topics will be discussed through case studies featuring real-world examples: - AI as a Communication Tool: Capabilities, Limitations, and the Hallucination Problem - Prompt Engineering: Communicating Effectively with AI Systems - AI-Assisted Writing: Drafting, Refining, and Maintaining Your Voice - Ethical and Legal Dimensions of AI in Communication (<i>includes India's Digital Personal Data Protection Act, 2023</i>) - AI, Bias, and Critical Evaluation: Reading AI Outputs Responsibly				
Targeted Application & Tools that can be used: Otter.ai, Microsoft Word/PowerPoint/Teams/Clipchamp/Whiteboard/OneNote/Forms/Copilot, Miro, Padlet, Canva, Mentimeter, AllSides Media Bias Ratings, Ad Fontes Media Bias Chart, Snopes/factcheck.rog, Zotero, NotebookLM, Canva, Adobe Express, CapCut, OBS Studio, Claude, ChatGPT, Gemini, DeepSeek, Google Docs				
Project Work / Assignment:				
- Multimodal Communication Portfolio (Project) with multiple milestones: Continuous, milestone based. Developed across all five modules. Final consolidated submission at the end of the course. Develop a progressive, publicly shareable communication portfolio that demonstrates the student's ability to move a technical concept across multiple audiences, formats, and media. - Active Listening and Contextual Response Lab (Individual): Students listen to an unedited, multi-speaker audio recording of a real or simulated professional scenario. Under timed conditions, students produce a structured written brief consisting of four parts. Assessed on comprehension accuracy, analytical precision, quality of questioning, and practicality of the proposed strategy. - Narrative and Adaptive Oral Delivery (Individual): Each student delivers a 4-minute structured presentation. At the 3-minute mark, the faculty assessor introduces a counter-perspective or an unexpected constraint, and the student must adapt their argument in real time for the remaining minute. The assessment evaluates: the quality of the prepared 3 minutes; the quality of the improvised adaptation; and the degree to which the full 4 minutes function as a coherent whole. - Facilitated Stakeholder Group Discussion (Group): Groups of 5–6 students are each assigned a distinct stakeholder position within a shared scenario. and must engage in a live, facilitated discussion to resolve a stated policy impasse. No prepared scripts are permitted; students receive their role brief 15 minutes before the discussion begins. Assessment is individual, not group-level. - Simulated Workplace Crisis Briefing (Group): Groups of 4–5 students receive an unexpected corporate or institutional emergency brief. Teams have exactly 30 minutes to read the brief, distil the critical information, assign speaking roles, and prepare a 5-minute status update for a simulated leadership panel. - Technical Translation and Annotated Script (Individual): Students receive a dense, jargon-heavy technical document. Students must: (a) read and fully comprehend the document; (b) identify the core idea that a non-specialist audience needs to understand; and (c) produce an annotated script for a 90-second public explainer. - Cross-Disciplinary Case Analysis and Defence (Group): Groups of 3–4 students select one case from a faculty-curated shortlist of five to six approved cases. The group delivers a 10-minute structured presentation that goes beyond narrative description - identify the specific communication failures, explain the ethical or regulatory dimensions of those failures, and propose what a communication-competent response would have looked like. This is followed				

immediately by a 5-minute live Q&A in which individual group members must defend their analytical assertions against probing questions from both faculty and peers.	
Text Book(s):	
1. Markel, M., & Selber, S. A. (2021). <i>Technical Communication</i> (13th ed.). Bedford/St. Martin's.	
Reference Book(s):	
R1. Carnegie, D. (1936/1998). <i>How to Win Friends and Influence People</i> . Simon & Schuster. R2. Orwell, G. (1946). <i>Politics and the English Language</i> . R3. Strunk, W., & White, E. B. (1999). <i>The Elements of Style</i> (4th ed.). Longman. R4. Kahneman, D. (2011). <i>Thinking, Fast and Slow</i> . Farrar, Straus and Giroux. R5. O'Neil, C. (2016). <i>Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy</i> . Crown Publishers. R6. Adler, R. B., Rodman, G., & DuPré, A. (2019). <i>Understanding Human Communication</i> (14th ed.). Oxford University Press.	
Web Sources:	
• owl.purdue.edu • plainlanguage.gov • allsides.com • adfontesmedia.com • ted.com • economist.com/styleguide • factcheck.org • ailiteracy.org • learnprompting.org • unesco.org/en/artificial-intelligence/ethics	
Topics relevant to "SKILL DEVELOPMENT": Active listening, verbal and non-verbal communication, interpersonal communication and self-awareness, elements of spoken communication — voice, clarity, and presence, storytelling and narrative in spoken contexts, structuring spoken messages, speaking with confidence, group communication and productive disagreement, giving and receiving feedback, presentation design and delivery, speaking for technical and non-technical audiences, critical reading, identifying tone and bias, understanding flawed reasoning, source credibility and information evaluation, reading multimodal texts, principles of effective writing, plain language and technical communication, professional writing — email, reports, and correspondence, argumentative writing, summary writing, prompt engineering, AI-assisted writing, ethical AI use in academic and professional communication, AI bias and critical evaluation.	
Catalogue prepared by	Dr Ajith K and Dr Shahanas Punnilath Shanavas
Recommended by the Board of Studies on	

Date of Approval by the Academic Council	
--	--

Course Code: CSS1500	Course Title: Computational Thinking using Python Type of Course: Lab Integrated Course (ESC)	L-T-P-C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	Nil		
Anti-requisites	NIL		
Course Description	This course efficiently introduces fundamental programming concepts including conditionals, loops, functions, lists, strings, tuples, and dictionaries through practical examples and problem-solving techniques. It also covers advanced programming concepts such as exception handling, file processing, modules, and packages using Python. The course emphasizes computational thinking, algorithm development, and implementation of searching and sorting techniques using Python data structures. Additionally, it introduces secure file handling, intelligent input validation, and AI-assisted data processing concepts for real-world applications.		
Course Objective	To develop computational thinking and problem-solving skills using Python by enabling students to design algorithms, implement programs using core programming constructs, apply data structures and file handling techniques, and develop secure and intelligent solutions for real-world applications.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Apply computational thinking principles to design algorithms for solving basic and intermediate computing problems. [Understand] CO2: Implement Python programs using data types, operators, and expressions to perform computational tasks. [Apply] CO3: Develop programs using control structures, functions, and recursion to solve structured problems. [Apply] CO4: Utilize data structures such as lists, tuples, and dictionaries for efficient data representation and manipulation. [Apply] CO5: Design and implement Python programs for secure file handling, exception management, and intelligent data validation for real-world applications. [Apply]		
Course Content:			
Module 1	Computational Thinking And Problem Solving	6 Sessions	

<u>Topics:</u> Fundamentals of Computing– Identification of Computational Problems Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi		
Module 2	Datatypes, Expressions, Statements	6 Sessions
<u>Topics:</u> Python interpreter and interactive mode,debugging; values and types: int, float, boolean, string , and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.		
Module 3	Control flow, Functions, Strings	6 Sessions
<u>Topics:</u> Conditionals:Boolean values and operators, conditional (if), alternative (if else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.		
Module 4	Lists, Tuples, Dictionaries	6 Sessions
<u>Topics:</u> Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing- list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.		
Module 5	Files	6 Sessions
Files and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter’s age validation, Marks range validation (0-100), Secure File Handling, Intelligent Error Detection using AI Concepts, Smart Input Validation Techniques, Automated Data Validation Applications.		
List of Practical Tasks		30 Sessions
WEEK 1: Computational Thinking & Algorithms		
Lab Sheet 1: Problem Solving & Algorithm Design		
Level 1: Write an algorithm and Python program to find the minimum element in a list of numbers.		
Level 2: Write an algorithm and Python program to insert an element into a sorted list at the correct position.		
Lab Sheet 2: Algorithm Strategies (Iteration & Recursion)		
Level 1: Write a Python program to guess a number within a given range using iteration.		
Level 2: Write a Python program to solve the Tower of Hanoi problem using recursion.		

WEEK 2: Data Types & Expressions

Lab Sheet 3: Data Types & Variables

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

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

Lab Sheet 4: Expressions & Operators

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

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

WEEK 3: Expressions & Problem Solving

Lab Sheet 5: Mathematical Expressions

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

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

Lab Sheet 6: Debugging & Interactive Mode

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

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

WEEK 4: Conditional Statements

Lab Sheet 7: Decision Making

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

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

Lab Sheet 8: Advanced Conditions

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

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

WEEK 5: Iteration (Loops)

Lab Sheet 9: Looping Basics

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

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

Lab Sheet 10: Loop Control

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

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

WEEK 6: Functions

Lab Sheet 11: Functions Basics

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

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

Lab Sheet 12: Recursion & Scope

Level 1: Write a recursive function to calculate exponentiation.

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

WEEK 7: Strings

Lab Sheet 13: String Operations

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

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

Lab Sheet 14: String Methods

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

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

WEEK 8: Lists

Lab Sheet 15: List Basics

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

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

Lab Sheet 16: List Operations

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

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

WEEK 9: Advanced Lists

Lab Sheet 17: List Methods

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

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

Lab Sheet 18: List Comprehension

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

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

WEEK 10: Tuples

Lab Sheet 19: Tuple Basics

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

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

Lab Sheet 20: Tuples as Return Values

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

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

WEEK 11: Dictionaries

Lab Sheet 21: Dictionary Basics

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

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

Lab Sheet 22: Dictionary Applications

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

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

WEEK 12: Advanced List Processing

Lab Sheet 23: Sorting & Searching

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

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

Lab Sheet 24: Data Processing

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

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

WEEK 13: File Handling Basics

Lab Sheet 25: File Operations

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

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

Lab Sheet 26: File Processing

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

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

WEEK 14: Exceptions & Validation

Lab Sheet 27: Exception Handling

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

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

Lab Sheet 28: Validation Programs

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

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

WEEK 15: Modules & Advanced Concepts

Lab Sheet 29: Modules & Packages

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

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

Lab Sheet 30: Command Line Arguments

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

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

WEEK 15.1: Secure and Intelligent Python Applications

Lab Sheet 31: Secure File Handling and Validation

Level 1: Write a Python program for secure file reading and writing with exception handling.

Level 2: Implement intelligent input validation for user data using exception handling techniques.

Lab Sheet 32: AI-based Python Applications

Level 1: Develop a Python program for automated data validation or log analysis.

Level 2: Implement a simple intelligent application using Python modules and file processing concepts.

[Project work/Assignment:](#)

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

[Text Book](#)

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

References

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

Web Resources

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

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

Course Code: ECE2022	Course Title: DIGITAL DESIGN Type of Course: School Core [Theory Only]	L-T-P-C	2-0-0-2
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The purpose of this course is to enable the students to appreciate the fundamentals of digital logic circuits and Boolean algebra. Further to focusing on combinational and sequential digital logic circuits. This course is analytical in nature and needs a fundamental knowledge on logical computation with Boolean algebra. The focus of the course will be to discuss the minimization techniques for making low-cost canonical digital circuit implementations. In this course, we emphasize design procedures of digital electronic circuits. Additionally, this course will create a foundation for future courses that includes Computer Architecture, Microprocessors, Microcontrollers and Embedded Systems etc.		
Course Objectives	The objective of the course is to familiarize the learners with Design and Analysis of Digital Logic Circuits concepts and to improve the learner's SKILL DEVELOPMENT by using PARTICIPATIVE LEARNING TECHNIQUES through Assessments.		
Course Out Comes	On successful completion of this course the students shall be able to: C01. Explain the fundamental concepts of Number system, Boolean algebra and Logic gates C02. Determine the minimum literal Boolean Expression using K-Map minimization techniques C03. Make use of relevant logic gates for constructing the combinational circuit. C04. Produce standard and real world combinational logic circuits using minimum number of logic gates C05. Demonstrate the working principles of flip-flops and latches within the context of sequential logic circuits. C06. Design synchronous and asynchronous logic circuit using Flip-flops		
Module 1	Number System, Logic Gates & Boolean function simplification (<i>Comprehension Level</i>)	Assignment	Problem Solving
10Hours			
Review of Number systems -Overview of Boolean Algebra- Boolean Laws, Rules & Theorems. Boolean function Standard form and Canonical form. Review of logic gates -Basic gates, Universal gates & Special			

Gates- Boolean functions simplifications using Algebraic functions & Karnaugh-map. SOP and POS simplifications & Implementations.				
Module 2	Combinational Logic circuits: (<i>Application Level</i>)	Assignment	Design	10hours
Digital logic circuit Introduction –Classification of circuit- Analysis and Design procedure of Combinational Logic circuits - Binary Adder and Subtractor- Magnitude comparator- , Multiplexers - DeMultiplexers, Decoders, Encoders and Priority Encoders-Real world scenario based application design using logic gates.				
Module 3	Sequential logic circuits (<i>Application Level</i>)	Quiz	Design	10 hours
Introduction to sequential circuits- Storage elements: latches and flip flops- Characteristic tables and equations, excitation table- Analysis of clocked sequential circuits- state diagram-state table and design of finite state machines - Registers & Counters.				
•				
Targeted Application & Tools that can be used:				
NA				
Project Based Learning proposed for this course				
NA				
Text Books:				
1. T1: Thomas L Flyod, “ <i>Digital Fundamentals</i> ”, Eleventh Edition, Pearson Education. ISBN-10: 132737965. (2014).eBook-[PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD abri.engenderhealth.org 2. T2. Mano, M. Morris and Ciletti Michael D., “ <i>Digital Design</i> ”, Pearson Education, 6 th edition ., [PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download (studymaterialz.in)				
References:				
R1 Jain, R. P., “ <i>Modern Digital Electronics</i> ”, McGraw Hill Education (India). R2 Roth, Charles H., Jr and Kinney Larry L., “ <i>Fundamentals of logic Design</i> ”, Cengage Learning.				
E-Resources:				
1. eBook1: Mano, M. Morris and Ciletti Michael D., “ <i>Digital Design</i> ”, Pearson Education. 2. {[PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download } 3. eBook2:Floyd “DIGITAL LOGIC DESIGN” fourth edition- ePub, eBook- [PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD abri.engenderhealth.org . 4. NPTEL Course- NPTEL :: Electrical Engineering - NOC:Digital Electronic Circuits 5. Digital Logic Design PPT Slide 1 (iare.ac.in)				
Catalogue prepared by	Dr.G.MUTHUPANDI			
Recommended by the Board of Studies on	21st BOS held on 9th June 2025			
Date of Approval by the Academic Council	26th Academic Council Meeting held on 25th July 2025			

Course Code: ECE1512	Course Title: Design Thinking in Practice	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces students to Design Thinking frameworks for identifying real-world user needs and transforming them into innovative technology solutions using embedded systems, sensors, IoT platforms, and AI-enabled computing devices. The course blends human-centered innovation methodologies with hands-on development using Arduino, Aries, ESP32, Raspberry Pi, NVIDIA Jetson, and GPU-based embedded platforms. Students will develop problem-solving skills while learning problem identification, ideation, rapid prototyping, sensor integration, edge AI, and intelligent system development for applications in smart automation, healthcare, agriculture, robotics, and Industry 5.0. The course emphasizes experiential and project-based learning to build future-ready innovators capable of developing scalable, intelligent, and socially impactful products.		
Course Objective	The objective of the course is to enhance students’ employability skills through participative learning techniques by integrating Design Thinking, embedded systems, IoT, and AI technologies to develop innovative, intelligent, and socially impactful real-world solutions.		
Course Outcomes	On successful completion of the course, the students shall be able to 1. Explain the principles, framework, and process of design thinking 2. Formulate Engineering Problem from user challenges, customer expectations, and market opportunities using design thinking process. 3. Analyze the Engineering Problem using the basic concepts of microcontroller and sensory devices. 4. Find alternate methods to solve the engineering problem by working with embedded computing platforms. 5. Develop functional prototypes using embedded boards, sensors, and an intelligent system for the chosen method 6. Validate prototype performance through testing, user feedback, and iterative improvements.		
Course Content			
Module 1	Introduction to Design thinking	6 Sessions	
Introduction to Design Thinking Importance of Design Thinking History of Design Thinking- Design Thinking Framework - Design Thinking process - Empathise Define Understanding Purpose - Ideate - What is Creativity? -Thinking Differently Basic Rules of Design Thinking The Creativity Heart Set			

Everyone is a Designer - Activity-based with case studies		
Module 2	Basic Concepts of microcontrollers and sensory devices	8 Sessions
Introduction to Arduino, ESP and NodeMCU - Pin Configuration and Architecture - Device and Platform Features - Concept of Digital and Analog Ports - Familiarizing with Arduino Interfacing Board Programming using Arduino IDE and Embedded C - Datatypes and Variables - I/O Functions Sensors, types and its applications - Humidity Sensor - Temperature Sensor - Water Detector / Sensor - PIR Sensor - Ultrasonic Sensor - Connecting Switches and Actuators - Sensor Interface with ESP Microcontroller - Various Cloud Platforms		
Module 3	Working with embedded computing platforms and 3D printing technology	8 Sessions
Introduction to Raspberry Pi boards, pin diagram, different types of Raspberry Pi boards and their applications, Configuration of Raspberry Pi boards - LED and switch control Exploring various libraries and their functions. GPU fundamentals Configuration of Jetson Nano board, Working with H200 GPU Servers. Introduction to 3D Printer: 3D Printer technology and its working Principles, Applications. Introduction to online Simulators: Working with Tinker CAD Simulator.		
Module 4	Prototype development and validation techniques	8 Sessions
Importance of Prototyping in Design Thinking Concept Refinement Using Prototypes From Idea to Functional Prototype User Testing and Validation Iteration and Design Optimization Prototype Readiness for Product Development (DFM & Market Feasibility overview) Activity-based with case studies.		
Prototype Development Tasks: Total Sessions: 30 1. Smart illumination system using Arduino 2. Intelligent Plant Health Monitoring System. 3. Aries Rover Drive and Control System Development 4. IoT-Based Environmental Intelligence System using ESP32 5. Edge Computing Platform Setup using Raspberry Pi 6. Raspberry Pi-based smart lighting Controller 7. VoiceX: AI-Powered Smart Home Automation System 8. VisionAI: Intelligent Object Recognition System. 9. Smart Traffic Management System using NVIDIA Jetson Nano 10. High-Performance AI Vision Analytics using H200-GPU Computing. 11. Rapid Prototype Development using 3D Design and printing		
Topics: Types of embedded boards, Thonny Python, Python IDLE, sensors, 3D Printer, Jetson Nano board, GPUs		

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, Aries, ESP32, Raspberry Pi, and sensors can be applied. The flexibility and affordability of all embedded boards, combined with the wide range of sensors available, allow for endless possibilities in creating innovative projects.

Professionally Used Software: Students can use open-source software, such as Arduino IDE, Blynk console, Tinker CAD, Thonny Python, Python IDLE, etc.

Project work/Assignment:

Projects: At the end of the course, students will be completing the project work on solving many real-time problem statements on different domains, such as women-centric, agritech, medtech, climate cell, robotics, drones, and smart cities etc.

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 to the library resources and write a report on their understanding of the assigned article in an appropriate format.

Presidency University Library Link Presentation: There will be a review presentation (Phase 1, Phase 2, and Phase 3) from among the groups, where the students will be given a project, and they have to demonstrate the working and discuss the applications for the same

References:

1. Design Thinking by Clarke, Rachel Ivy. Series: Library Futures, Vol. 4. Chicago: ALA Neal-Schuman. 2020. eBook., Database: eBook Collection (EBSCOhost)
2. Neerparaj Rai, "Arduino Projects for Engineers," BPB Publishers, first edition, 2016.
3. Stewart Watkiss "Learn Electronics with Raspberry Pi" Apress Berkeley, CA. Second edition, 2020. ISBN978-1-4842-6348-8
4. Jo Prusa, "Basics of 3D printing", Prusa Research, 3rd edition.
5. Volker Ziemann, "A Hands-On Course in Sensors Using the Arduino and Raspberry Pi (Series in Sensors)", CRC Press, 1st Edition. 2018.
6. Professional CUDA C Programming. Professional CUDA C Programming. Wrox Press, 2014.

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

1. Arduino trending Projects - <https://www.projecthub.arduino.cc>
2. Introduction to Arduino - https://onlinecourses.swayam2.ac.in/aic20_sp04/preview
3. Case studies on Wearable technology - <https://www.hticiitm.org/wearables>
4. Raspberry Pi Projects - <https://magpi.raspberrypi.com/articles/category/tutorials/>
5. Introduction to internet of things <https://nptel.ac.in/courses/106105166>
6. Getting Started with Jetson Nano - <https://www.youtube.com/watch?v=DVfmw3Ftu3I>

E-content:

1. Tim Brown, "Design Thinking," Harvard Business Review, vol. 86, no. 6, pp. 84-92, Jun. 2008.
2. Rolf Faste, "Ambidextrous Thinking," in Innovations in Mechanical Engineering Curricula for the 1990s, American Society of Mechanical Engineers, 1994, pp. 1-6.
3. M. H. Hemanth Kumar, Ravi Pratap Singh, Nishu Sharma, and Pragya Singh, "IoT Based Smart Security System Using Arduino," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 8, no. 8, Aug. 2021.
4. Basil and Eliza Sawant, S. D., "IoT-Based Traffic Light Control System Using Raspberry Pi," in Proceedings of ICECDS 2017, 2017. DOI: 10.1109/ICECDS.2017.8389604.
5. Espressif Systems, "ESP32 Series Datasheet," 2024.
6. Josef Průša, "Open-Source 3D Printers and the RepRap Project," RepRap Magazine, 2018.

Topics relevant to the development of "SKILL":

System design for achieving Sustainable Development Goals.

**Catalogue
prepared by**

Dr. Divyarani M S, Dr. M Manikandan

**Recommended
by the Board of
Studies on**

BoS Number:

**Date of Approval
by the Academic
Council**

Academic Council Meeting No. dated on

Course Code: PHY2504	Course Title: Optoelectronics and Quantum Physics Lab Type of Course: 1] School Core	L-T-P-C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The laboratory provides an opportunity to validate the concepts taught and enhances the ability to use the concepts for technological applications. The laboratory tasks aim to develop following skills: An attitude of enquiry, confidence and ability to tackle new problems, ability to interpret events and results, observe and measure physical phenomena, select suitable equipment, instrument and materials, locate faults in systems.		

Course Out Comes	On successful completion of the course the students shall be able to: CO1: To understand electrical and optical properties of materials CO2: Interpret the results of various experiments to verify the concepts used in optoelectronics and advanced devices.
Course Objective	The objective of the course is to familiarize the learners with the concepts of “ Applied Physics for Computer Science Cluster ”and attain Skill Development through Experiential Learning techniques

List of Laboratory Tasks:

Experiment No. 1: Experimental errors and uncertainty using excel

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

Level 2: propagation of errors in addition, subtraction, multiplication and division.

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

Level 1: Determination of Wavelength of Laser

Level 2: Finding the particle size of lycopodium powder.

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

Level 1: To determine the proportionality of Hall Voltage and magnetic flux density

Level 2: To determine the polarity of Charge carrier.

Experiment No. 4: To study the I-V characteristics of a given zener diode in forward and reverse bias conditions.

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

Level 2: To study I –V characteristics of the given Zener diode in forward bias and to determine knee voltage and forward resistance.

Experiment No. 5: To study input and output characteristics of a given Transistor.

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

Level 2: To determine current transfer characteristics and transistor parameters of a given transistor.

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

Level 1: Determination of Fermi energy and Fermi temperature of given metal wire.

Level 2: Determination of Fermi energy and Fermi temperature of given bimetallic wire.

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

Level 1: To study the I-V characteristics

Level 2: I-R characteristics of a solar cell as a function of the irradiance.

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

Level 1: Calculate the numerical aperture.

Level 2: study the losses that occur in optical fiber cable.

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

Level 1: Plotting I-V characteristics in forward and reverse bias for LEDs

Level 2: Determination of knee voltage.

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

Level 1: Determination of Stefan's constant

Level 2: Verification of Stefan-Boltzmann Law.

Experiment No. 11: : Dielectric constant

Level 1: Determination of Dielectric constant of given material

Level 2: compare the obtain results with other materials

Experiment No. 12: determine the wavelength of monochromatic light, such as sodium light, using Newton's rings.

Level 1: Determination of wavelength

Level 2: determine the radius of curvature of the Plano-convex lens.

Targeted Application & Tools that can be used:

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.

Course Code: PPS1025	Course Title: Industry Readiness Program – I Type of Course: Practical Only Course	L- T - P- C	0-0-2-0
Version No.	1.0		
Course Pre- requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.		
Anti-requisites	NIL		
Course Description	This course is designed to enable students to set SMART goals, form professional & personal ethics for success and learn various email writing techniques. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Employability for Young Professionals” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.		
Course Out Comes	On successful completion of this course the students shall be able to: CO 1 Define their career goals CO 2 Practice ethical habits for better career success CO3 Demonstrate effective email writing technique		
Course Content			
Module 1	Goal Setting & Grooming	Classroom activities	10 Hours

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

Course Code: ECE2052	Course Title: Digital Design Lab Type of Course: School Core [Lab Only]	L-T-P-C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course gives knowledge through the simulation software for design and implementation of various digital logic circuit. The theoretical concepts and working principles of combinational and sequential logic circuits are verified through implementation using openCV. This course provides necessary learning for fundamental concepts of computer and communication engineering. The purpose of this course is to support the students to exhibit the design and analysis skills. The course enhances the Design and Implementation abilities through laboratory tasks. Additionally, it provides an opportunity to verify the theoretical knowledge.		
Course Objectives	The objective of the course is to familiarize the learners with the design concepts of digital logic circuit and attain Skill Development through Experiential Learning by simulation-based laboratory Assessments.		
Course Out Comes	<i>On successful completion of the course the students shall be able to</i> C01. Interpret Boolean laws and the logic gates functionality using available circuit elements in simulator platforms. C02. Apply the fundamental laws and rules of Boolean algebra for simplified circuit simulation using any simulator tool. C03. Develop combinational logic circuits by connecting offered gates for basic arithmetic operation C04. Make use of circuit elements for ensuring the digital logic circuit's truth table perfection. C05. Examine the performance of sequential logic circuit simulation using flip-flop and latch elements C06. Demonstrate the operation of clocked and edge-triggered sequential components in an open-source simulation environment		
LIST OF EXPERIMENTS			
	S. No	Name of the Experiment/Exercise	
	1	Logic Gates	
	2	Boolean Algebra	
	3	Adder	
	4	Subtractor	
	5	Magnitude Comparator	
	6	Design of Combinational Circuits	
	7	Multiplexer, Demultiplexer & Applications of MUX	
	8	Flip-Flops	
	9	Shift Registers	

	10	Counter Design	
Targeted Application & Tools that can be used: Targeted Applications: This course is contributed for placement in core companies, research & development work and also useful to know the existing & developing communications. Professionally Used Software: CIRCUIT VERSE, HDL,VHDL Project Based Learning proposed for this course			
NA			
Text Books: T1. Mano, M. Morris and Ciletti Michael D., "Digital Design", Pearson Education, 6th edition T2. Thomas L. Floyd "DIGITAL LOGIC DESIGN" , Pearson Education, fourth edition.			
References: R1. Jain, R. P., "Modern Digital Electronics", McGraw Hill Education (India), Fourth Edition R2. Roth, Charles H., Jr and Kinney Larry L., "Fundamentals of logic Design", Cengage Learning, Seventh Edition • Presidency Lab Manual- for ECE2052 DIGITAL DESIGN LAB			
E-Resources: 1. eBook1: Mano, M. Morris and Ciletti Michael D., "<i>Digital Design</i>", Pearson Education. Book Free Download (studymaterialz.in) [PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download 2. eBook2:Floyd "DIGITAL LOGIC DESIGN" fourth edition- ePub, eBook- [PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD abri.engenderhealth.org. 3. NPTEL Course- NPTEL :: Electrical Engineering - NOC:Digital Electronic Circuits 4. Digital Logic Design PPT Slide 1 (iare.ac.in) 5. Lab Tutorial: Multisim Tutorial for Digital Circuits - Bing video CircuitVerse - Digital Circuit Simulator online Learn Logisim ➡ Beginners Tutorial Easy Explanation! - Bing video Digital Design 5: LOGISIM Tutorial & Demo			
Catalogue prepared by	Dr.G.MUTHUPANDI		
Recommended by the Board of Studies on	21st BOS held on 9th June 2025		
Date of Approval by the Academic Council	26th Academic Council Meeting held on 25th July 2025		

Course Code: CIV7601	Course Title: Universal Human Values and Ethics Type of Course: MAC course	L-T-P-C	-	-	-	0
Course Pre-requisites	NIL					

Anti-requisites	NIL			
Course Description	The purpose of the course is to develop a holistic perspective in students' life. The course adopts a self-reflective methodology of teaching and is designed to equip the students to explore their role in all aspects of living as a part of the society. It presents a universal approach to value education by developing the right understanding of reality through the process of self-exploration. This self-exploration develops more confidence and commitment in students enabling them to critically evaluate their pre-conditioning and present beliefs. As an outcome of the holistic approach, the students will be able to practice the ethical conduct in the social and professional life. The prime focus throughout the course is toward affecting a qualitative transformation in the life of the student rather than just a transfer of information. This course is designed to cater to Human Values and Professional Ethics.			
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using 'SELF LEARNING' techniques			
Course Outcomes	On successful completion of this course the students shall be able to: Recognize the importance of Value Education through the process of self-exploration Explain the human being as the co-existence of the self and the body in harmony. Describe the role of foundational values in building harmonious relationships. Summarize the importance of a holistic perspective in developing ethical professional behavior.			
Course Content:				
Module 1	Introduction to Value Education	Online Assessment	MCQ Quiz	5 Sessions
Topics: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education), Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfil the Basic Human Aspirations.				
Module 2	Harmony in the Human Being	Online Assessment	MCQ Quiz	5 Sessions
Topics: Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health				
Module 3	Harmony in the Family and Society	Online Assessment	MCQ Quiz	5 Sessions
Topics:				

Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.				
Module 4	Implications of the Holistic Understanding – A Look at Professional Ethics	Online Assessment	MCQ Quiz	5 Sessions
Topics: Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics, Holistic Technologies, Strategies for Transition towards Value-based Life and Profession				
Targeted Application & Tools that can be used: Application areas are Personal life, Education and Career, Workplace , Society and Environmental Responsibility Tools: Online Tools – NPTEL and Swayam.				
Project work/Assignment:				
Assessment Type Online exams (MCQs) will be conducted by the Department of Civil Engineering through Linways.				
Online Link*: UHV II – 1. https://www.youtube.com/watch?v=NhFBzn5qKIM&list=PLWDeKF97v9SO8vvjC1KyqteziTbTjN1So&pp=0gcICWMEOCosWNin 2. Lecture by Dr. Kumar Sambhav, NPTEL course: Universal Human Values, https://onlinecourses.swayam2.ac.in/aic22_ge23/preview 3. Lecture by Dr. Padmavati, Dr Narendran Thiruthy, NPTEL Course: Biodiversity Protection, Farmers and Breeders Rights, https://nptel.ac.in/courses/129105008, 2024. * Other source links are available in below Resources link. Text Book T1: A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1 T2: Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2019. T3: Premvir Kapoor, Professional Ethics and Human Values, Khanna Book Publishing, New Delhi, 2022.				
Reference Books R1: E.F. Schumacher, 1973, Small is Beautiful: a study of economics as if people mattered, Blond & Briggs, Britain. R2: Sussan George, 1976, How the Other Half Dies, Penguin Press. Reprinted 1986.				

R3: Donella H. Meadows, Dennis L. Meadows, Jorgen Randers, William W. Behrens III, 1972, Limits to Growth – Club of Rome’s report, Universe Books.

R4: A Nagraj, 1998, Jeevan Vidya Ek Parichay, Divya Path Sansthan, Amarkantak.

R5: P L Dhar, RR Gaur, 1990, Science and Humanism, Commonwealth Publishers.

R6: A N Tripathy, 2003, Human Values, New Age International Publishers.

R7: E G Seebauer & Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers, Oxford University Press

R8: M Govindrajran, S Natrajan & V.S. Senthil Kumar, Engineering Ethics (including Human Values), Eastern Economy Edition, Prentice Hall of India Ltd.

R9: B P Banerjee, 2005, Foundations of Ethics and Management, Excel Books.

R10: William P. Cunningham and Mary Ann Cunningham (2020), Principles of Environmental Science: Inquiry & Applications, 9th Edition, McGraw-Hill Education, USA.

Resources:

https://onlinecourses.swayam2.ac.in/imb25_mg195/preview

https://onlinecourses.nptel.ac.in/noc25_mg141/preview

https://onlinecourses.swayam2.ac.in/ini25_hs52/preview

https://onlinecourses.nptel.ac.in/noc25_hs219/preview

https://onlinecourses.swayam2.ac.in/cec25_mg14/preview

https://onlinecourses.swayam2.ac.in/imb25_mg195/preview

https://onlinecourses.swayam2.ac.in/imb25_mg196/preview

Topics relevant to Skill Development:

An attitude of enquiry.

Write reports

The topics related to Human values and Professional ethics:

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

Course Code: MAT2311	Course Title: Vector Calculus, Numerical Methods and Sampling Distributions [Type of Course:1] School Core	L-T- P- C	3-1-0-4
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		

Course Description	This course introduces the fundamental concepts and methods of ordinary differential equations, numerical techniques, and statistical inference used in engineering and scientific applications. The course covers analytical methods for solving differential equations, numerical methods for solving linear and nonlinear equations, interpolation, numerical differentiation and integration, and numerical solutions of ordinary differential equations. It also introduces sampling distributions and hypothesis testing for statistical decision-making. The course enables students to apply mathematical and statistical techniques to solve computational and real-world engineering problems.	
Course Objective	The objectives of this course are to develop the ability to solve ordinary differential equations arising in scientific and engineering applications. Introduce numerical techniques for solving systems of linear equations and nonlinear algebraic equations. Apply interpolation, differentiation, and integration methods for approximating mathematical functions and evaluating definite integrals. Employ numerical methods for obtaining approximate solutions of ordinary differential equations. Apply concepts of sampling distributions and statistical inference to estimate population parameters and support decision-making. Apply hypothesis testing techniques to real-world problems involving means, proportions, variances, and associations among variables.	
Course Outcomes	After successful completion of the course, students will be able to: C01: Employ vector differential operators to solve problems involving scalar and vector fields and classify them as solenoidal or irrotational (L3) C02: Apply vector integration techniques and integral theorems to solve engineering problems. (L3) C03: Use numerical methods to solve systems of linear and nonlinear equations. (L3) C04: Implement numerical techniques for interpolation, differentiation, integration, and ordinary differential equations. (L3) C05: Utilize sampling distributions and statistical inference techniques to solve statistical problems. (L3) C06: Perform hypothesis testing for statistical decision-making. (L3)	
Course Content:		
Module 1	Vector Calculus	(4L+5T=19 Sessions)
Vector & Scalar Functions and Fields, Derivatives, Curve, Arc length, Curvature & Torsion, Gradient of Scalar Field, Directional Derivative, Divergence of a Vector Field, Curl of a Vector Field, Physical interpretation, Solenoidal and Irrotational Vector Fields. Problems. Line Integrals, Path Independence of Line Integrals, Surface and Volume Integrals, Green's Theorem in the plane, Divergence Theorem of Gauss, Stokes's Theorem. Problems on Image Edge Detection, Robot Path Planning and 3D Surface Modelling		
Module 2	Solution of Linear Systems and non-linear of Equation	(9L+3T=12 Sessions)
Introduction, Jacobi iteration method, Gauss-Seidel iteration method. Solution of Algebraic and Transcendental Equations: Bisection method, Regula-Falsi method, Newton-Raphson method, secant method, fixed point iteration method. Page Ranking, Network Analysis, Circuit Simulation, Root Finding, Load Balancing, and Computational Optimization Problems.		
Module 3	Numerical Interpolations and Differential Equations	(10L+3T=13 Sessions)
Newton's method, Divided difference method, Lagrange's method, Numerical differentiation, Numerical integration: Trapezoidal rule, Simpson's one-third rule, Simpson's three-eighth rule, Gaussian quadrature rule. Solution of ordinary differential equations: Taylor series method, modified Euler's method, Runge-Kutta method for 4th order. Problems on Signal Reconstruction, Sensor Data Interpolation and Trajectory Prediction.		
Module 4	Sampling Distributions	(2L+4T=16 Sessions)
Random sampling, sampling distributions, Standard Error, Type I & Type II errors, Testing of Hypothesis, Test of significance - Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations, Test for single mean, difference of means, test		

for ratio of variances - Chi-square test for goodness of fit and independence of attributes. A/B Testing, Fraud Detection, Spam Classification Evaluation, and Network Traffic Analysis.

Targeted Applications & Mathematical Tools:

- Analysis of fluid flow in open channels and pipelines, groundwater movement, stress and strain fields, terrain modelling, and 3D surface representation in civil engineering projects.
- Structural analysis, load distribution in trusses and frames, hydraulic network analysis, foundation design computations, and optimization problems in transportation and water resource systems.
- Estimation of terrain elevations, survey data interpolation, rainfall-runoff modelling, settlement prediction, traffic flow analysis, and numerical solutions of engineering differential equations.
- Quality control of construction materials, reliability analysis, traffic studies, environmental data analysis, hydrological investigations, and hypothesis testing in engineering experiments.
- Analysis of vector fields in fluid flow, electromagnetics, and heat transfer; image edge detection, robot path planning, and 3D surface modelling using gradients, divergence, curl, and integral theorems.

Software: MATLAB, Python, Microsoft Excel Solver, LINDO/LINGO, AutoCAD, ETABS, Scilab.

Assignment:

Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor.

Text Book

1. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, 7th ed. New Delhi, India: New Age International Publishers, 2019.
2. B. S. Grewal, *Higher Engineering Mathematics*, 45th ed. New Delhi, India: Khanna Publishers, 2022.
3. S. C. Gupta and V. K. Kapoor, *Fundamentals of Mathematical Statistics*, 12th ed. New Delhi, India: Sultan Chand & Sons, 2020.

References:

1. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, 7th ed. New Delhi, India: New Age International Publishers, 2019.
2. S. S. Sastry, *Introductory Methods of Numerical Analysis*, 5th ed. New Delhi, India: PHI Learning Pvt. Ltd., 2012.
3. J. Kiusalaas, *Numerical Methods in Engineering with Python 3*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2025.
4. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 8th ed. Hoboken, NJ, USA: John Wiley & Sons, 2023.
5. D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications*, 7th ed. Belmont, CA, USA: Cengage Learning, 2008.
6. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
7. R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*, 11th ed. Boston, MA, USA: Cengage Learning, 2021.
8. S. C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2018.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_53751
2. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=TEXTBOOK_LIBRARY01_06082022_117
3. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7313
4. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_228965
5. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_8838
6. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=SPRINGER_OPEN_ACCESS_BOOKS_10022025_1861

7. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7312	
Topics relevant to SKILL DEVELOPMENT: Identification and solution of differential equations, application of numerical methods for algebraic and differential equations, implementation of interpolation and numerical integration techniques, statistical data analysis and hypothesis testing, and use of Python/MATLAB for engineering problem solving.	
Catalogue prepared by	Dr. V Ramalatha
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: CSS2251	Course Title: Data Communications and Computer Networks Type of Course: Theory Course (PCC)	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The objective of this course is to provide knowledge in data communications and computer networks, its organization and its implementation, and gain practical experience in the installation, monitoring, and troubleshooting of LAN systems. The associated laboratory is designed to implement and simulate various networks using Cisco packet tracer, NS2. All the lab exercises will focus on the fundamentals of creating multiple networks, topologies and analyzing the network traffics.		
Course Objectives	To develop a strong foundation in data communication and computer networks by enabling students to understand networking concepts, analyze network architectures and protocols, explore transmission media and switching techniques, and evaluate network performance and emerging networking technologies.		
Course Outcomes	On successful completion of the course, the students shall be able to: C01: Explain the fundamental concepts of data communication and computer networks. [Understand] C02: Analyze the functions and protocols of the Data Link Layer. [Analyze] C03: Apply IP addressing schemes and routing techniques in computer networks. [Apply] C04: Explain the working principles of the Transport and Application layers. [Understand] C05: Analyze network protocols and evaluate their performance for efficient data communication. [Analyze]		

Course Content:		
Module 1	Introduction and Physical Layer	11 Sessions
Topics: Introduction to Computer Networks and Data communications, Network Components – Topologies, Transmission Media –Reference Models -OSI Model – TCP/IP Suite. Physical Layer -Analog and Digital Signals – Digital and Analog Signals – Transmission - Multiplexing and Spread Spectrum.		
Module 2	Reference Models and Data Link Layer	11 Sessions
Topics: Data Link Layer - Error Detection and Correction – Parity, LRC, CRC, Hamming Code, Flow Control and Error Control, Stop and Wait, ARQ, Sliding Window, Multiple Access Protocols, CSMA/CD,CSMA/CA, IEEE 802.3, IEEE 802.11 Ethernet.		
Module 3	Network Layer	11 Sessions
Topics: Network Layer Services - Network Layer Services, Switching Techniques, IP Addressing methods- IPv4 IPV6 – Subnetting. Routing, - Distance Vector Routing – RIP-BGP-Link State Routing –OSPF-Multi cast Routing-MOSPF- DVMRP – Broad Cast Routing. EVPN- VXLAN, VPLS, ELAN.		
Module 4	Transport and Application Layer	12 Sessions
Topics: Transport Layers - Connection management – Flow control – Retransmission, UDP, TCP, congestion control, – Congestion avoidance (DECbit, RED) . The Application Layer: Domain Name System (DNS), Domain Name Space, SSH, FTP, Electronic Mail (SMTP, POP3, IMAP, MIME) – HTTP – – SNMP, Web Services, Virtual Networking. AI-driven network traffic analysis and prediction, AI-based intrusion and anomaly detection in networks, IoT networking protocols (MQTT, CoAP), IoT security challenges and secure communication.		
Targeted Application & Tools that can be used: Cisco Packet Tracer, Wireshark, and NS2. Case Study/Assignment: Choose and analyze a network from any organization/Assignment proposed for this course in CO1-CO4 Problem Solving: Choose and appropriate devices and implement various network concepts. Programming: Simulation of any network using NS2.		

Text Book(s):

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

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

Reference(s):

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

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

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

E- Resources:

1. <https://archive.nptel.ac.in/courses/106/105/106105183/>

2. <http://www.nptelvideos.com/course.php?id=393>

3. <https://www.youtube.com/watch?v=3DZLltfbqtQ>

4. https://www.youtube.com/watch?v=_fldQ4yfsfM

5. <https://www.digimat.in/keyword/106.html>

6. <https://puniversity.informaticsglobal.com/login>

Course Code: CSS2252	Course Name: Data Communications and Computer Networks Lab Type of Course: Laboratory Course (PCC)	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre- requisites	NIL		
Anti- requisites	NIL		

Course Description	This lab-based course provides hands-on experience in the principles and practices of data communications and computer networking. It is designed to complement theoretical concepts covered in the associated lecture course. Through a series of structured experiments and practical exercises, students will gain proficiency in configuring, analyzing, and troubleshooting computer networks. Key topics include network topology design, IP addressing and subnetting, Ethernet and LAN technologies, routing and switching, TCP/IP protocol suite, and basic network security measures. Students will work with industry-standard tools and equipment, including routers, switches, protocol analyzers, and network simulation software such as Cisco Packet Tracer or Wireshark.
Course Objective	To develop practical skills in computer networking by enabling students to gain hands-on experience with networking tools and protocols, simulate and analyze data communication processes, implement socket programming for network applications, and evaluate network performance through experimental studies.
Course Outcomes	On successful completion of this course, students will be able to: C01: Design and configure basic network topologies using routers, switches, and end devices. [Apply] C02: Analyze and troubleshoot network connectivity and performance issues using networking tools and simulators. [Analyze] C03: Demonstrate the working of key networking protocols such as TCP/IP, ARP, ICMP, and DHCP. [Apply] C04: Apply IP addressing and subnetting techniques for efficient network resource management. [Apply] C05: Evaluate network performance and optimize configurations for improved reliability and efficiency. [Evaluate]
Course Content:	
List of Laboratory Tasks: 30 Sessions WEEK 1: Introduction to Networks Lab Sheet 1: Network Basics Level 1: Identify different types of networks (LAN, WAN, MAN) and list their characteristics. Level 2: Analyze a real-world network setup and classify its components and topology. Lab Sheet 2: Network Topologies Level 1: Draw and explain Bus, Star, Ring, and Mesh topologies. Level 2: Compare topologies and suggest the best topology for a given scenario.	

WEEK 2: Transmission Media & Models

Lab Sheet 3: Transmission Media

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

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

Lab Sheet 4: OSI & TCP/IP Models

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

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

WEEK 3: Physical Layer Concepts

Lab Sheet 5: Signals & Transmission

Level 1: Differentiate between analog and digital signals.

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

Lab Sheet 6: Multiplexing

Level 1: Explain FDM, TDM, and WDM.

Level 2: Design a multiplexing scenario for multiple users.

WEEK 4: Spread Spectrum & Encoding

Lab Sheet 7: Spread Spectrum

Level 1: Explain FHSS and DSSS.

Level 2: Compare spread spectrum techniques in wireless communication.

Lab Sheet 8: Encoding Techniques

Level 1: Demonstrate digital encoding techniques.

Level 2: Analyze encoding schemes for efficient transmission.

WEEK 5: Error Detection & Correction

Lab Sheet 9: Error Detection

Level 1: Implement Parity and LRC.

Level 2: Implement CRC algorithm.

Lab Sheet 10: Error Correction

Level 1: Implement Hamming Code.

Level 2: Analyze error correction efficiency.

WEEK 6: Flow Control

Lab Sheet 11: Stop and Wait

Level 1: Simulate Stop-and-Wait protocol.

Level 2: Analyze efficiency and delay.

Lab Sheet 12: Sliding Window

Level 1: Implement Sliding Window protocol.

Level 2: Compare with Stop-and-Wait.

WEEK 7: Multiple Access Protocols

Lab Sheet 13: CSMA/CD

Level 1: Explain working of CSMA/CD.

Level 2: Simulate collision detection.

Lab Sheet 14: CSMA/CA

Level 1: Explain CSMA/CA.

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

WEEK 8: IEEE Standards

Lab Sheet 15: Ethernet (802.3)

Level 1: Study Ethernet frame structure.

Level 2: Analyze Ethernet performance.

Lab Sheet 16: Wireless LAN (802.11)

Level 1: Study Wi-Fi architecture.

Level 2: Analyze wireless communication challenges.

WEEK 9: Network Layer Basics

Lab Sheet 17: IP Addressing

Level 1: Identify IPv4 address classes.

Level 2: Convert IPv4 to binary and vice versa.

Lab Sheet 18: IPv6

Level 1: Understand IPv6 format.

Level 2: Compare IPv4 and IPv6.

WEEK 10: Subnetting

Lab Sheet 19: Subnetting Basics

Level 1: Calculate subnet mask.

Level 2: Design subnet for given requirements.

Lab Sheet 20: Advanced Subnetting

Level 1: Perform CIDR calculations.

Level 2: Solve real-world subnetting problems.

WEEK 11: Routing

Lab Sheet 21: Distance Vector Routing

Level 1: Explain RIP algorithm.

Level 2: Simulate routing table updates.

Lab Sheet 22: Link State Routing

Level 1: Explain OSPF.

Level 2: Compare RIP and OSPF.

WEEK 12: Advanced Routing

Lab Sheet 23: Multicast Routing

Level 1: Explain multicast concepts.

Level 2: Analyze MOSPF/DVMRP.

Lab Sheet 24: Modern Networking

Level 1: Study VXLAN/EVPN.

Level 2: Analyze virtual networking concepts.

WEEK 13: Transport Layer

Lab Sheet 25: TCP & UDP

Level 1: Compare TCP and UDP.

Level 2: Analyze use cases.

Lab Sheet 26: Congestion Control

Level 1: Explain congestion control methods.

Level 2: Analyze RED/DECbit techniques.

WEEK 14: Application Layer

Lab Sheet 27: DNS & HTTP

Level 1: Explain DNS working.

Level 2: Analyze HTTP request-response.

Lab Sheet 28: Email Protocols

Level 1: Explain SMTP, POP3, IMAP.

Level 2: Compare email protocols.

WEEK 15: Network Applications

Lab Sheet 29: SNMP & Web Services

Level 1: Explain SNMP architecture.

Level 2: Analyze web services.

Lab Sheet 30: Virtual Networking

Level 1: Study virtual networking basics.

Level 2: Analyze real-world virtual network implementation.

Lab Sheet 31: AI-Based Network Traffic Monitoring using IoT

Level 1: Study IoT-based network traffic monitoring using sensors and smart devices.

Level 2: Implement AI-based anomaly detection for identifying unusual network traffic patterns.

Lab Sheet 32: Smart IoT Network Automation using AI

Level 1: Study IoT-enabled smart network automation concepts.

Level 2: Design an AI-based system for automatic device management and network optimization.

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

Case Study/Assignment: Choose and analyze a network from any organization/Assignment proposed for this course.

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

Programming: Simulation of any network using NS2.

Text Book(s):

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

Reference(s):

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

E- Resources:

1. <https://archive.nptel.ac.in/courses/106/105/106105183/>
2. <http://www.nptelvideos.com/course.php?id=393>
3. <https://www.youtube.com/watch?v=3DZLItfbqtQ>
4. https://www.youtube.com/watch?v=_fldQ4yfsfM
5. <https://www.digimat.in/keyword/106.html>
6. <https://puniversity.informaticsglobal.com/login>

Course Code: CSS2253	Course Name: Data Structures Type of Course: Theory Course (PCC)	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the fundamental concepts of data structures and emphasizes the importance of selecting appropriate data structures and techniques for efficient program development. The course includes both theory and laboratory components focusing on the implementation and applications of data structures using the Java programming language. It also introduces emerging applications of data structures in cybersecurity, Artificial Intelligence, Machine Learning, blockchain, and intelligent computing systems. With strong knowledge of data structure fundamentals and practical implementation skills, students will be able to design and develop efficient, secure, and intelligent software applications.		
Course Objective	To develop a strong foundation in data structures by enabling students to understand fundamental data structures and their operations, analyze algorithm efficiency, implement linear and non-linear data structures, and apply appropriate techniques to solve real-world problems, including applications in cybersecurity, AI/ML, blockchain, and intelligent systems, while evaluating trade-offs between different data structures.		
Course Outcomes	On successful completion of this course, students will be able to: C01: Explain the concepts of basic data structures such as arrays, stacks, and queues and their operations. [Understand] C02: Apply linked list data structures to solve real-world problems. [Apply] C03: Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios. [Apply] C04: Implement and demonstrate various searching and sorting algorithms and Analyze the		

	efficiency of data structures and algorithms in terms of time and space complexity [Apply] CO5:. Apply data structure concepts in cybersecurity, AI/ML, blockchain, and intelligent computing applications. [Apply]	
Course Content:		
Module 1	Introduction to Data Structure and Linear Data Structure – Stacks and Queues	9 Sessions
Introduction – Introduction to Data Structures, Types and concept of Arrays. Stack - Concepts and representation, Stack operations, stack implementation using array and Applications of Stack. Queues - Representation of queue, Queue Operations, Queue implementation using array, Types of Queue and Applications of Queue.		
Module 2	Linear Data Structure- Linked List	12 Sessions
Topics: Linked List - Singly Linked List, Operation on linear list using singly linked storage structures, Circular List, Applications of Linked list. Recursion - Recursive Definition and Processes.		
Module 3	Non-linear Data Structures - Trees	12 Sessions
Topics: Trees - Introduction to Trees, Binary tree: Terminology and Properties, Use of Doubly Linked List, Binary tree traversals: Pre-Order traversal, In-Order traversal, Post - Order traversal, Binary Search Tree, AVL Trees - Red Black Tree, Expression Tree , Heaps.		
Module 4	Non-linear Data Structures – Graphs, Hashing, Searching and Sorting	6 Sessions
Topics: Graphs: Basic Concept of Graph Theory and its Properties, Representation of Graphs . ADT, Elementary graph operations, Minimum Cost spanning trees, Shortest path and Transitive closure. Hashing: Introduction, Static Hashing, Dynamic Hashing Searching & Sorting Sequential and Binary Search, Sorting – Selection and Insertion sort, Quick sort, Merge Sort, Bubble sort.		
Module 5	Security and AI/ML Applications of Data Structures	6 Sessions
Topic: Secure Data Structures, Hashing Techniques, Secure Searching and Sorting, Trees and Graphs in Cybersecurity, Data Structures for AI & ML, Graph-based Intelligent Systems, Heaps in Machine Learning, Applications in Fraud and Intrusion Detection, Blockchain Data Structures, Quantum Computing Data Representation.		
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 et al., 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: CSS2254	Course Name: Data Structures Lab Type of Course: Laboratory Course (PCC)	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the fundamental concepts of data structures and to emphasize the importance of choosing an appropriate data structure and technique for program development. This course has theory and lab component which emphasizes on understanding the implementation and applications of data structures using Java programming language. With a good knowledge in the fundamental concepts of data structures and practical experience in implementing them, the student can be an effective designer, developer for new software applications.		
Course Objective	To develop programming and problem-solving skills by enabling students to implement various data structures, analyze their performance through experimentation and apply structured approaches to solve computational problems effectively.		

Course Outcomes	On successful completion of this course, students will be able to: C01: Explain the concepts of fundamental data structures such as arrays, stacks, and queues and their operations. [Understand] C02: Implement linear data structures such as linked lists to solve real-world problems. [Apply] C03: Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios. [Apply] C04: Implement and demonstrate various searching and sorting algorithms and Analyze the efficiency of data structures and algorithms in terms of time and space complexity [Apply] C05:. Apply data structure concepts in cybersecurity, AI/ML, blockchain, and intelligent computing applications. [Apply]
Course Content:	30 Sessions
List of Laboratory Tasks:	30 Sessions WEEK 1: Introduction & Arrays Lab Sheet 1: Arrays Basics Level 1: Write a program to create an array and perform insertion and deletion operations. Level 2: Write a program to find largest, smallest, and average elements in an array. Lab Sheet 2: Array Applications Level 1: Write a program to reverse an array. Level 2: Write a program to merge two arrays and remove duplicates. WEEK 2: Stack Lab Sheet 3: Stack Implementation Level 1: Implement stack using array (push and pop operations). Level 2: Write a program to check overflow and underflow conditions. Lab Sheet 4: Stack Applications Level 1: Write a program to convert infix expression to postfix. Level 2: Write a program to evaluate postfix expression. WEEK 3: Queue Lab Sheet 5: Queue Implementation

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

Level 2: Implement circular queue.

Lab Sheet 6: Queue Applications

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

Level 2: Implement priority queue.

WEEK 4: Linked List Basics

Lab Sheet 7: Singly Linked List

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

Level 2: Perform deletion and display operations.

Lab Sheet 8: Linked List Applications

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

Level 2: Reverse a linked list.

WEEK 5: Advanced Linked List

Lab Sheet 9: Circular Linked List

Level 1: Create and traverse a circular linked list.

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

Lab Sheet 10: Linked List Problems

Level 1: Merge two linked lists.

Level 2: Detect loop in linked list.

WEEK 6: Recursion

Lab Sheet 11: Recursion Basics

Level 1: Write a recursive program for factorial.

Level 2: Write a recursive program for Fibonacci series.

Lab Sheet 12: Recursion Applications

Level 1: Solve Tower of Hanoi using recursion.

Level 2: Compare recursive and iterative approaches.

WEEK 7: Trees Basics

Lab Sheet 13: Binary Tree

Level 1: Create a binary tree and perform traversal.

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

Lab Sheet 14: Tree Properties

Level 1: Count nodes and height of tree.

Level 2: Check if tree is balanced.

WEEK 8: Binary Search Tree

Lab Sheet 15: BST Implementation

Level 1: Insert elements into BST.

Level 2: Delete elements from BST.

Lab Sheet 16: BST Applications

Level 1: Search an element in BST.

Level 2: Find minimum and maximum elements.

WEEK 9: Advanced Trees

Lab Sheet 17: AVL Trees

Level 1: Implement AVL tree insertion.

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

Lab Sheet 18: Heaps

Level 1: Implement max heap.

Level 2: Perform heap sort.

WEEK 10: Expression Trees & Red-Black Trees

Lab Sheet 19: Expression Tree

Level 1: Construct expression tree from postfix.

Level 2: Evaluate expression tree.

Lab Sheet 20: Red-Black Tree

Level 1: Understand insertion rules.

Level 2: Implement basic operations.

WEEK 11: Graph Basics

Lab Sheet 21: Graph Representation

Level 1: Represent graph using adjacency matrix.

Level 2: Represent graph using adjacency list.

Lab Sheet 22: Graph Traversal

Level 1: Implement BFS.

Level 2: Implement DFS.

WEEK 12: Graph Algorithms

Lab Sheet 23: Shortest Path

Level 1: Implement Dijkstra's algorithm.

Level 2: Analyze shortest path in weighted graph.

Lab Sheet 24: Minimum Spanning Tree

Level 1: Implement Prim's algorithm.

Level 2: Implement Kruskal's algorithm.

WEEK 13: Hashing

Lab Sheet 25: Hash Tables

Level 1: Implement hashing using division method.

Level 2: Handle collisions using chaining.

Lab Sheet 26: Advanced Hashing

Level 1: Implement linear probing.

Level 2: Compare static vs dynamic hashing.

WEEK 14: Searching

Lab Sheet 27: Searching Techniques

Level 1: Implement sequential search.

Level 2: Implement binary search.

Lab Sheet 28: Searching Applications

Level 1: Compare searching techniques.

Level 2: Analyze time complexity.

Lab Sheet 29: Basic Sorting

Level 1: Implement bubble sort and selection sort.

Level 2: Implement insertion sort.

Lab Sheet 30: Advanced Sorting

Level 1: Implement quick sort.

Level 2: Implement merge sort and compare performance

WEEK 15: Sorting Lab Sheet 31: Secure Data Structures and Hashing Level 1: Implement hashing techniques for secure data storage and retrieval. Level 2: Demonstrate collision handling methods and simple password verification using hashing. Lab Sheet 32: AI/ML and Intelligent Data Structure Applications Level 1: Implement graph or tree-based data representation for intelligent systems. Level 2: Develop a simple application using data structures for recommendation, prediction, or intrusion detection systems.
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 et al., 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: CSS2255	Course Name: Object Oriented Programming Using Java Type of Course: Theory		L-T- P- C	3-0-0-3
Version No.	2.0			
Course Pre-requisites	CSE1502			
Anti-requisites	Nil			
Course Description	This course introduces the core concepts of Object-Oriented Programming. The course includes both theory and laboratory components that emphasize the implementation and application of object-oriented programming paradigms using Java. It enables students to develop real-time, secure, and intelligent applications by applying object-oriented concepts for effective problem solving. The course also introduces modern concepts related to AI/ML-based application development, secure programming practices, and intelligent software systems. Students will understand the significance of object-oriented programming in designing scalable, reusable, secure, and intelligent applications.			
Course Objective	The objective of the course is to familiarize learners with problem-solving concepts using Java and to attain skill development through experiential learning techniques. The course also aims to develop the ability to design secure, scalable, and intelligent applications using object-oriented programming concepts with exposure to AI/ML-oriented programming practices and modern software development approaches.			
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 Java I/O concepts for intelligent data processing, secure file handling, and AI/ML-based application development. [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	Advanced Java I/O, Security and AI/ML Applications	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, AI-based File Processing Concepts, Intelligent Data Stream Management, Machine Learning Applications in File and Data Handling.				
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”: 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: CSS2256	Course Name: Object Oriented Programming Using Java Lab Type of Course: Laboratory Course (PCC)	L-T- P- C	0-0-2-1
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the core concepts of Object-Oriented Programming. The course includes both theory and laboratory components that emphasize the implementation and application of object-oriented programming paradigms using Java. It enables students to develop real-time, secure, and intelligent applications by applying object-oriented concepts for effective problem solving. The course also introduces modern concepts related to AI/ML-based application development, secure programming practices, and intelligent software systems. Students will understand the significance of object-oriented programming in designing scalable, reusable, secure, and intelligent applications.		
Course Objective	The objective of the course is to familiarize learners with problem-solving concepts using Java and to attain skill development through experiential learning techniques. The course also aims to develop the ability to design secure, scalable, and intelligent applications using object-oriented programming concepts with exposure to AI/ML-oriented programming practices and modern software development approaches.		
Course Outcomes	On successful completion of the course, students shall be able to: CO1: Demonstrate fundamental programming concepts. [Apply] CO2: Apply the concepts of classes, objects, and methods to solve problems. [Apply] CO3: Apply the concepts of arrays and strings. [Apply] CO4: Implement inheritance and polymorphism to develop secure applications. [Apply] CO5: Apply Java I/O concepts for intelligent data processing, secure file handling, and AI/ML-based application development. [Apply]		
Course Content:		30 Sessions	
List of Laboratory Tasks: Week 1: Lab Sheet 1: Java Program Structure & Basic I/O Level 1: Write a Java program to display “Hello, Java”. Demonstrate use of variables and data types. Level 2:			

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

Week 2:

Lab Sheet 2: Operators and Expressions

Level 1:

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

Level 2:

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

Week 3:

Lab Sheet 3: Control Statements – Branching

Level 1:

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

Level 2:

Implement a menu-driven calculator using switch-case.

Week 4:

Lab Sheet 4: Control Statements – Looping

Level 1:

Write programs using loops:

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

Level 2:

Generate Fibonacci series and check whether a number is prime.

Week 5:

Lab Sheet 5: Classes and Objects

Level 1:

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

Level 2:

Create multiple objects and demonstrate accessing members using reference variables.

Week 6:

Lab Sheet 6: Methods and Access Specifiers

Level 1:

Write a class with methods to perform addition and subtraction.

Level 2:

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

Week 7:

Lab Sheet 7: Constructors and Overloading

Level 1:

Implement a class with a default constructor and parameterized constructor.

Level 2:

Demonstrate constructor overloading and use of this keyword.

Week 8:

Lab Sheet 8: Static Keyword & Nested Classes

Level 1:

Write a program demonstrating static variables and static methods.

Level 2:

Create a nested class and access its members from the outer class.

Week 9:

Lab Sheet 9: Arrays (1D and 2D)

Level 1:

Write programs for:

1. Sum of array elements
2. Find largest element

Level 2:

Perform matrix addition and multiplication using 2D arrays.

Week 10:

Lab Sheet 10: Array of Objects

Level 1:

Create an array of objects (e.g., Employee class).

Level 2:

Perform operations like sorting objects based on a field.

Week 11:

Lab Sheet 11: Strings and StringBuffer

Level 1:

Perform basic string operations: Length, concatenation, comparison

Level 2:

Demonstrate mutable operations using StringBuffer and StringBuilder.

Week 12:

Lab Sheet 12: Inheritance

Level 1:

Create a base class and derived class using single inheritance.

Level 2:

Implement multilevel inheritance and demonstrate use of super keyword.

Week 13:

Lab Sheet 13: Polymorphism & Abstract Classes

Level 1:

Demonstrate method overriding (runtime polymorphism).

Level 2:

Create an abstract class and implement its methods in a subclass.

Week 14:

Lab Sheet 14: Exception Handling

Level 1:

Handle exceptions using try-catch-finally.

Level 2:

Create custom exceptions and demonstrate multiple catch blocks.

Week 15:

Lab Sheet 15: File Handling and Serialization Level 1: Write a Java program to Create a file & Write and read data using File I/O Level 2: Implement object serialization and deserialization. Use File streams and buffered streams for efficient I/O. Week 15.1: AI/ML and Secure File Processing Lab Sheet 16: Intelligent File Handling and Data Processing Level 1: Implement a Java program for secure file handling using buffered streams and file channels. Level 2: Develop a simple application for intelligent data analysis or log monitoring using Java File I/O and serialization concepts.			
Targeted Application & Tools that can be used : JDK /Eclipse IDE/Visual Studio Code / net Beans IDE.			
Text Book 1. Herbert Schildt, "The Complete Reference Java 2", Tata McGraw Hill Education, 11th Edition, 2019.			
References 1. Cay S Horstmann and Cary Gornell, "CORE JAVA volume I-Fundamentals", Tenth Edition, Pearson 2015. 2. James W. Cooper, "Java TM Design Patterns – A Tutorial", Addison-Wesley Publishers. 4th Edition, 2000. 3. E. Balagurusamy, "Programming with Java", Tata McGraw Hill Education, 6th Edition, 2019. E book link R1: http://rmi.yaht.net/bookz/core.java/9780134177373-Vol-1.pdf E book link R2: Java(tm) Design Patterns: A Tutorial([PDF] [7qmsenjl97t0] (vdoc.pub) Web resources https://youtube.com/playlist?list=PLu0W_9lII9agS67Uits0UnjYrYiXhDS6q https://puniversity.informaticsglobal.com:2229/login.aspx			
Topics relevant to development of "Skill Development": 1. Static Polymorphism 2. Method overloading, constructors 3. constructor overloading 4. this keyword 5. static keyword and Inner classes 6. Inheritance and Polymorphism. This is attained through assessment component mentioned in course handout.			

Course Code: CSS2257	Course Name: Computer Organization and Architecture Type of Course: Theory Course (PCC)	L-T- P- C	3-0-0-3
Version No.	2.0		

Course Pre-requisites	NIL	
Anti-requisites	NIL	
Course Description	This course introduces the core principles of computer architecture and organization from basic to intermediate level. This theory-based course emphasizes understanding the interaction between computer hardware and software. It equips students with the intuition behind assembly-level instruction set architectures and helps them interpret the operational concepts of computer technology and performance enhancement. The course also introduces emerging concepts in computer security, AI-enabled computing, intelligent processor optimization, and modern secure computing architectures.	
Course Objective	To develop a comprehensive understanding of computer system organization by enabling students to analyze processor architecture and instruction execution, explore memory hierarchy and cache mechanisms, examine input/output organization and interfacing, evaluate system performance and optimization techniques, and understand emerging applications in secure computing, AI-enabled architectures, and intelligent system design.	
Course Outcomes	On successful completion of this course, students will be able to: CO1: Describe the fundamental components of a computer system and their interconnections. [Remember] CO2: Explain the concepts of Instruction Set Architecture (ISA) and memory organization. [Understand] CO3: Apply techniques to perform arithmetic and logical operations in computer systems. [Apply] CO4: Explain the organization and functioning of processor and memory subsystems. [Understand] CO5: Apply concepts of computer architecture in security, AI-enabled computing, and intelligent system optimization. [Apply]	
Course Content:		
Module 1	Basic Structure of Computer	12 Sessions
Topics: Computer Types, Functional Units, Basic Operational concepts, Bus Structures, Computer systems RISC & CISC, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Arithmetic Operations on Signed numbers. Instructions and Instruction Sequencing, Instruction formats, Memory Instructions.		
Module 2	Instruction Set Architecture and Memory Unit	12 Sessions
Topics: Instruction Set Architecture: Addressing Modes, Stacks and Subroutines.		

Memory System: Memory Location and Addresses, Memory Operations, Semiconductor RAM Memories, Internal Organization of Memory chips, Cache memory mapping Techniques.		
Module 3	Arithmetic And Input/output Design	10 Sessions
Topics: Arithmetic: Carry lookahead Adder, Signed-Operand Multiplication, Integer Division, and Floating point operations. Input/output Design: Accessing I/O Devices, I/O communication, Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits		
Module 4	BPU and Pipelining	05 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.		
Module 5	Security, AI and Intelligent Computing Architecture	06 Sessions
Computer Security Fundamentals, Secure Instruction Execution, Memory Security and Protection Techniques, Cache and Processor Security, Cryptographic Fundamentals in Computer Architecture, AI-assisted Processor Design, Machine Learning Concepts in Hardware Optimization, Intelligent Memory Management Techniques, AI-based Performance Optimization, Applications of AI and Security in Modern Computing Systems.		
Targeted Application & Tools that can be used: Targeted employment sector is processor manufacturing and memory chip fabrication vendors like Intel, AMD, Motorola, NVidia, Samsung, Micron Technology, western Digital etc. Targeted job profiles include Memory circuit design and verification engineers, Physical system design engineer, System programmer, Fabrication engineer etc. Tools: Virtual Lab, IIT KGP Tejas – Java Based Architectural Simulator, IIT Delhi		
Project work/Assignment:		
Each batch of students (self-selected batch mates – up to 4 in a batch) will be allocated case studies/assignments		
Textbook(s): 1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Computer Organization”, Sixth Edition, McGraw-Hill Higher Education, 2023 reprint. 2. William Stallings, “Computer Organization & Architecture – Designing for Performance”, 11th Edition, Pearson Education Inc., 2019.		
References 1. David A. Patterson & John L. Hennessy, “Computer Organization and Design MIPS Edition- The Hardware/Software Interface”, 6th Edition, Morgan Kaufmann, Elsevier Publications, November 2020.		

2. Web References:
3. NPTEL Course on “Computer architecture and organization” IIT Kharagpur By Prof. Indranil Sengupta, Prof. Kamalika Datta. https://nptel.ac.in/courses/106105163
4. NPTEL Course on “Computer Organization”, IIT Madras By Prof. S. Raman.
5. https://nptel.ac.in/courses/106106092
6. https://puniversity.informaticsglobal.com:2229/login.aspx
Topics relevant to “SKILL DEVELOPMENT”: Generation of Computers, CISC and RISC processors, Bus Arbitration, Collaboration and Data collection for Term assignments and Case Studies for Skill Development through Participative Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CDV2000	Course Name: DevOps Foundations Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites			
Anti-requisites	NIL		
Course Description	This course 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 Objective	- 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 Out Comes	On successful completion of the course the students shall be able to: C01: Describe DevOps principles, processes, and benefits (Understand) C02: Examine CI/CD practices and DevOps toolchains (Analyze) C03: Implement version control, automated builds, and test pipelines (Apply) C04: Deploy containerized applications using basic orchestration (Apply)		

Course Content:				
Module 1	Introduction to DevOps and Culture	Assignment	Conceptual Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Tool-Based Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Practical Assignment	11 Sessions
<u>Topics:</u> Containers vs VMs, Docker architecture, Dockerfile and images, Docker Compose, Container registry, Application deployment and rollback, Kubernetes overview				
Module 4	Infrastructure and Monitoring	Assignment	Case Study / Mini Project	12 Sessions
<u>Topics:</u> 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 Targeted Application & Tools that can be used: Git, GitHub, GitLab, Jenkins, GitHub Actions, Maven, Gradle, Docker, Docker Compose, Kubernetes, Terraform, Ansible, Prometheus, Grafana, and cloud platforms such as AWS, Azure, and Google Cloud Professionally used software : Git, Jenkins, Docker, Kubernetes, Terraform, Ansible, Prometheus, Grafana, GitHub Actions, and cloud platforms like AWS, Azure, and Google Cloud				
Text Book T1: Emily Freeman, DevOps for Dummies, Wiley, 2023 T2: Mikael Krief, Learning DevOps: Continuously Deliver Better Software, Packt Publishing, 2023				
References R1: Gene Kim, Jez Humble, Patrick Debois, <i>The DevOps Handbook</i> , IT Revolution, 2022 R2: Len Bass et al., <i>DevOps: A Software Architect's Perspective</i> , 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>

Topics relevant to "SKILL DEVELOPMENT": Develop practical skills in implementing DevOps practices, including version control, CI/CD pipeline automation, containerization, cloud deployment, and monitoring for efficient and continuous software delivery.

Course Code: CSS2271	Course Name: Software Design and Development Type of Course: Theory Course (PCC)	L- T- P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces foundational concepts of software engineering and emphasizes a practical, structured approach to software system development. It spans software requirement analysis, process modeling, system design, Agile principles, DevOps culture, testing strategies, quality assurance, and software maintenance. Hands-on assignments and problem-solving activities foster understanding of contemporary software practices and industry tools such as CASE tools, GitHub, and Selenium. The course also introduces secure software engineering practices, DevSecOps concepts, and AI-enhanced tools for documentation, automation, intelligent testing, and software quality prediction workflows.		
Course Objectives	To develop a comprehensive understanding of software engineering by enabling students to analyze software development life cycle models, apply design principles and patterns, develop modular and maintainable software systems, perform testing and quality assurance, manage software projects effectively, and understand secure and AI-driven software development practices.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain software engineering principles, ethics, and various software development life cycle models. [Understand] CO2: Identify software requirements and develop design models, including Software Requirement Specification (SRS) documents. [Apply] CO3: Apply Agile development practices and DevOps principles in software development. [Apply] CO4: Apply software testing strategies, configuration management, and maintenance techniques. [Apply] CO5: Evaluate secure and AI-driven software engineering practices to ensure software quality, reliability, scalability, and intelligent automation. [Evaluate]		

Course Content:		
Module 1	Introduction to Software Engineering and Process Models	10 Sessions
Introduction: Need for Software Engineering, Professional Software Development, Software Engineering Ethics, Software Engineering Practice-Essence of Practice, General Principles Software Development Life Cycle. Models: Waterfall Model – Classical Waterfall Model, Iterative Waterfall Model, Evolutionary model-Spiral, Prototype.		
Module 2	Software Requirements, Analysis and Design	12 Sessions
Requirements Engineering: Eliciting requirements, Functional and non- Functional requirements, Software Requirements Specification (SRS), Requirement Analysis and validation. Requirements modelling- Introduction to Use Cases, Activity diagram and Swim lane diagram. CASE support in Software Life Cycle, Characteristics of CASE Tools, Architecture of a CASE Environment. Design: Design concepts, Architectural design, Component based design, User interface design.		
Module 3	Agile Principles & Devops	10 Sessions
Agile: Scrum Roles and activities, Sprint Agile software development methods - Scaling, User Stories, Agile estimation techniques, Product backlogs, Stake holder roles, Dynamic System Development Method. DevOps: Introduction, definition, history, tools.		
Module 4	Software Testing and Maintenance	08 Sessions
Software Testing-verification and validation, Test Strategies - White Box Testing, Black box Testing. Automation Tools for Testing. Software Quality Assurance-Elements of software quality assurance, SQA Tasks, Goals and Metrics, Software configuration management- SCM process, SCM Tools (GitHub). Maintenance- Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models.		
Module 5	Secure, Intelligent and AI-Driven Software Engineering	05 Sessions
Secure Software Development Practices, Software Security Testing, AI in Software Engineering, Intelligent Requirement Analysis, AI-based Testing and Debugging, DevSecOps Concepts, Machine Learning for Software Quality Prediction, Smart Project Management Tools, Applications of AI in Modern Software Development.		
Targeted Application & Tools that can be used: Git hub/Lambda/Taskade/Bit.ai/Testsigma/TestRigor		
Text Books: T1. Roger S. Pressman, “Software Engineering – A Practitioner’s Approach”, VII Edition, McGraw-Hill, 2017. T2. Bob Hughes, Mike Cotterell, Rajib Mall, “Software Project Management”, VI Edition, McGraw-Hill, 2018.		
References: R1. Rajib Mall, “Fundamentals of Software Engineering”, VI Edition, PHI learning private limited, 2015. R2. Ian Sommerville, “Software Engineering”, IX Edition, Pearson Education Asia, 2011. R3. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002 Journals / Magazines IEEE Software		

• ACM Transactions on Software Engineering and Methodology • Journal of Systems and Software SWAYAM / NPTEL / MOOCs • NPTEL – Software Engineering (Prof. Rajib Mall, IIT KGP) • Coursera – Agile Development (University of Virginia) • Udemy – DevOps Masterclass • EdX – Software Testing Fundamentals (TUM)
Topics Relevant to “Skill Development”: Black box Testing, White box Testing, AI tools for design, modelling, testing and document Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout

Course Code: APT4002	Course Title: Introduction to Aptitude (Audited)	L- P- C	0-2-0
Version No.	1.0		
Course Pre-requisites	Students should know the basic Mathematics & aptitude along with understanding of English		
Anti-requisites	Nil		
Course Description	The objective of this course is to prepare the trainees to tackle the questions on various topics and various difficulty levels based on Quantitative Ability, and Logical Reasoning asked during the placement drives. There will be sufficient focus on building the fundamentals of all the topics, as well as on solving the higher order thinking questions. The focus of this course is to teach the students to not only get to the correct answers, but to get there faster than ever before, which will improve their employability factor.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Aptitude and attain Skill Development through Problem Solving techniques.		

Course Outcomes	On successful completion of the course the students shall be able to: C01] Recall all the basic mathematical concepts they learnt in high school. C02] Identify the principle concept needed in a question. C03] Solve the quantitative and logical ability questions with the appropriate concept. C04] Analyze the data given in complex problems. C05] Rearrange the information to simplify the question			
Course Content:				
Module 1	Quantitative Ability	Assignment	Bloom's Level : Application	12 Hours
Topics: Introduction to Aptitude, working of Tables, Squares, Cubes				
Module 2	Logical Reasoning	Assignment	Bloom's Level : Application	18 Hours
Topics: Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Ordering and Ranking, Clocks and Calendars, Number Series, Wrong number series, Visual Reasoning				
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS				
Text Book - Quantitative Aptitude by R S Aggarwal - Verbal & Non-Verbal Reasoning by R S Aggarwal				
References www.indiabix.com www.youtube.com/c/TheAptitudeGuy/videos				
Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques . This is attained through assessment component mentioned in course handout.				

Course Code: CHE7601	Course Name: Environmental Studies Type of Course: MOOC course	L- T-P- C	1-0-0-0
Version No.	1.0		
NCrF Level-	5.0		
Course Pre-requisites	NIL		

Co-requisites	NIL										
Anti-requisites	CSEXXXX										
Course Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PARTICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamental environmental concepts and their relevance to business operations, preparing them to address forthcoming sustainability challenges. It is designed to equip students with the knowledge and skills needed to make decisions that account for environmental consequences, fostering environmentally sensitive and responsible future managers. This course is designed to cater to Environment and Sustainability										
Course Objective	The objective of the course is ‘SKILL DEVELOPMENT’ of the student by using ‘PARTICIPATIVE LEARNING’ techniques										
Course Outcomes	On successful completion of the course the students shall be able to: <table><tr><td>CO1: Describe the issues related to natural resources, ecosystems and biodiversity</td><td>L1– Remember</td></tr><tr><td>CO2: Identify environmental hazards affecting air, water and soil quality</td><td>L2– Understand</td></tr><tr><td>CO3:Recognize the importance of healthy environment and finding the sustainable methods to protect the environment</td><td>L3 – Apply</td></tr><tr><td>CO4: Convert skills to address immediate environmental concerns through changes in environmental processes, policies, and decisions</td><td>L4 – Analyze</td></tr></table>			CO1: Describe the issues related to natural resources, ecosystems and biodiversity	L1– Remember	CO2: Identify environmental hazards affecting air, water and soil quality	L2– Understand	CO3:Recognize the importance of healthy environment and finding the sustainable methods to protect the environment	L3 – Apply	CO4: Convert skills to address immediate environmental concerns through changes in environmental processes, policies, and decisions	L4 – Analyze
CO1: Describe the issues related to natural resources, ecosystems and biodiversity	L1– Remember										
CO2: Identify environmental hazards affecting air, water and soil quality	L2– Understand										
CO3:Recognize the importance of healthy environment and finding the sustainable methods to protect the environment	L3 – Apply										
CO4: Convert skills to address immediate environmental concerns through changes in environmental processes, policies, and decisions	L4 – Analyze										
Course Content:											
Module 1	Understanding Environment, Natural Resources, and Sustainability										
Topics: Classification of natural resources, issues related to Population growth and their overutilization, and strategies for their conservation. Water, air, soil, mineral, energy and food source. Effect of human activities on natural resources. Concept of sustainability- Sustainable Development Goals (SDGs)- targets and indicators, challenges and strategies for SDGs; Sustainable practices in managing resources, including deforestation, water conservation, Desalination – types, energy security, and food security issues, Life Cycle thinking and Circular Economy.											
Module 2	Ecosystems, Biodiversity, and Sustainable Practices										
Topics:											

Ecosystems and ecosystem services: Various natural ecosystems, Major ecosystem types in India and their basic characteristics; forests, wetlands, grasslands, agriculture, coastal and marine; Ecosystem services- classification and their significance. The importance of biodiversity, Types of biodiversity, Biodiversity and Climate Change, the threats it faces, hotspots, and the methods used for its conservation. Strategies for in situ and ex situ conservation, mega diverse nation.			
Module 3	Environmental Pollution, Waste Management, and Sustainable Development		
Topics: Types of pollution- Chemical, - Biological, Biomedical, noise, air, water, soil, thermal, radioactive and marine pollution, and their impacts on society. Urbanization and Urban environmental problems; effects, and mitigation. Causes of pollution, such as global climate change, ozone layer depletion, the greenhouse effect, and acid rain, with a particular focus on pollution episodes in India. Importance of adopting cleaner technologies; Solid waste management; Sustainable Materials and Technologies: Biodegradable and compostable materials, Recycled and reclaimed materials (E-waste management), Sustainable manufacturing processes.			
Module 4	Social Issues, Legislation, and Practical Applications		
Topics: Overview of key environmental legislation and the judiciary's role in environmental protection, including the Water (Prevention and Control of Pollution) Act of 1974, the Environment (Protection) Act of 1986, and the Air (Prevention and Control of Pollution) Act of 1981. Hazardous waste Rule 1989, Biomedical Waste handling 1998, Fly Ash Rule 1999, Municipal Solid Waste Rule 2000, Battery Rules 2001, E- Waste Rules 2011, Plastic waste management Rules 2016, Construction Demolition waste Rules 2016 National Biodiversity Action Plan (NBAP) Major International Environmental Agreements: Convention on Biological Diversity (CBD), The Biological Diversity (Amendment) Act, 2023, United Nations Framework Convention on Climate Change (UNFCCC); Kyoto Protocol; Paris Agreement. Major International organisations and initiatives: United Nations Environment Programme (UNEP), United Nations Educational, Scientific and Cultural Organization (UNESCO), Intergovernmental Panel on Climate Change (IPCC).			
Targeted Application & Tools that can be used: Application areas are Energy, Environment and sustainability Tools: Online Tools – NPTEL and Swayam.			
Project work/Assignment:			
Assessment Type • Online exams (MCQs) will be conducted by the department of Chemistry			
Online Link*:			

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

* Other source links are available in below Resources link.

Text Book

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

Reference Books

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

Resources:

1. <https://nptel.ac.in/courses/109105203>
2. <https://archive.nptel.ac.in/courses/120/108/120108004/>
3. <https://nptel.ac.in/courses/127105018>
4. https://onlinecourses.nptel.ac.in/noc23_lw06/preview
5. https://onlinecourses.swayam2.ac.in/ini25_bt02/preview
6. <https://archive.nptel.ac.in/courses/120/108/120108002/>
7. https://onlinecourses.swayam2.ac.in/ini25_bt02/preview
8. <https://nptel.ac.in/courses/102104088>
9. <https://nptel.ac.in/courses/124107165>
10. <https://nptel.ac.in/courses/109106200>
11. <https://archive.nptel.ac.in/content/storage2/courses/120108004/module1/lecture1.pdf>
12. https://onlinecourses.swayam2.ac.in/nou25_ge19/preview
13. https://onlinecourses.swayam2.ac.in/ini25_hs01/preview
14. <http://kcl.digimat.in/nptel/courses/video/105105184/L32.html>
15. <https://nptel.ac.in/courses/105105169>

Topics relevant to Skill Development:

1. An attitude of enquiry.
2. Write reports

The topics related to Environment and Sustainability :

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

Course Code: MAT2321	Course Title: Discrete Mathematical Structures Type of Course:1] School Core	L-T- P- C	3-1-0-4
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the fundamental concepts of Discrete Mathematics, providing the mathematical foundations essential for computer science and related disciplines. It covers sets, relations, functions, closures, equivalence relations, and primitive recursive functions, followed by partially ordered sets, lattices, and Boolean algebras. The course also explores graph theory, including graph types, isomorphism, Euler and Hamiltonian paths, circuits, and graph coloring. In addition, students study algebraic structures such as semigroups, monoids, and groups, residue arithmetic, and group codes. Emphasis is placed on developing logical reasoning, mathematical modeling, and problem-solving skills applicable to computing, information systems, and network analysis.		
Course Objective	The primary objective of this course is to provide students with a strong foundation in discrete mathematical structures and logical reasoning that underpin computer science and related disciplines. It aims to develop the ability to construct mathematical arguments, analyze discrete problems, and apply formal methods to problem solving. The course introduces fundamental concepts of logic, sets, relations, functions, Boolean algebra, graph theory, and algebraic structures such as semigroups, monoids, and groups, while also exploring formal languages and grammar. Through these topics, students will acquire the analytical and mathematical skills necessary for applications in algorithm design, digital systems, coding theory, cryptography, and other areas of computing.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Apply propositional and predicate logic, logical equivalences, normal forms, inference rules, and proof techniques to analyze arguments and solve problems in mathematical reasoning and computer science. (L3) CO2: Apply the concepts of sets, relations, functions, closures, equivalence relations, and primitive recursive functions to model and solve problems in discrete mathematics and computer science. (L3) CO3: Employ partially ordered sets, lattices, Boolean algebras, and Boolean functions, and apply their properties to solve problems in discrete mathematical structures and logical systems. (L3) CO4: Apply the concepts of lattices, Boolean algebras, and Boolean functions to solve logical and computational problems. (L3) CO5: Apply graph theoretic concepts and techniques to solve problems in networks, connectivity, and optimization applications. (L3) CO6: Employ group theory and residue arithmetic techniques in coding, computation, and information processing. (L3)		
Course Content:			
Module 1	Fundamentals of Logic	(11L+4T=14 Sessions)	
Propositional Logic, Propositional Logic, Equivalences, Normal forms, Inference rules, Introduction to Proofs, Conversion to clausal form, Predicate calculus, The Statement function, Inference theory of the Predicate Calculus. Compiler optimization and program correctness verification. Cybersecurity and cryptographic protocol verification.			

Module 2	Relations and Functions	(10L+3T=13 Sessions)
Sets and set-operations, Relations, Functions, Properties of Relations & Representations of Relation, closure of different types of relations, equivalence relations, primitive recursive function. Database Management Systems, Artificial Intelligence and Machine Learning.		
Module 3	Boolean Algebra and Graph Theory	(13L+4T=17 Sessions)
Partially Ordered Sets, Lattices, Boolean Algebras, Functions of Boolean Algebras, Boolean Functions as Boolean Polynomials. Computer Networks: Routing and switching logic. Robotics: Decision-making using Boolean logic. Graph Theory-Basic Concept of Graph Theory, Types of Graphs, Isomorphism, Euler Paths and Circuits, Hamiltonian Paths and Circuits, Colouring of Graph. Social Networks (Facebook, Instagram, LinkedIn).and GPS & Navigation: Shortest path and route optimization.		
Module 4	Groups	(11L+4T=15 Sessions)
Semi Groups, Monoids, Groups, The applications of Residue Arithmetic to Computers, Group codes. Functional programming and distributed systems. Compiler design and programming languages.		
Targeted Application & Tools that can be used: • Compiler optimization, program correctness verification, cybersecurity protocol verification, automated reasoning, formal software verification. • Database Management Systems (DBMS), Artificial Intelligence (AI), Machine Learning (ML), knowledge representation, data modeling and query processing. • Digital circuit design, computer networks (routing and switching), robotics decision-making systems, logic controller design, social network analysis, GPS and navigation systems, route optimization, communication networks, and recommendation systems. • Cryptography, error-correcting codes, compiler design, distributed systems, residue arithmetic in computer architecture. Software: MATLAB, Lean 4, Mathematica, Maple, Python, GeoGebra, Octave, R, Scilab.		
Assignment: Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor. Additionally, experiential learning activities will be conducted using Lean 4, enabling students to develop formal proofs, verify logical propositions, and gain hands-on experience in theorem proving and formal verification.		
Text Book 6. K. H. Rosen, <i>Discrete Mathematics and Its Applications</i>, 8th ed. New York, NY, USA: McGraw-Hill Education, 2019. 7. F. Harary, <i>Graph Theory</i>. Reading, MA, USA: Addison-Wesley Publishing Company, 1969.		
References: 1. S. Epp, <i>Discrete Mathematics with Applications</i>, 5th ed. Boston, MA, USA: Cengage Learning, 2019. 2. K. A. Ross and C. R. B. Wright, <i>Discrete Mathematics</i>, 6th ed. New Delhi, India: Pearson Education, 2018. 3. L. Mott, A. Kandel, and T. P. Baker, <i>Discrete Mathematics for Computer Scientists and Mathematicians</i>, 3rd ed. New Delhi, India: PHI Learning, 2016. E-resources/ Web links: 4. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=TEXTBOOK_LIBRARY01_06082022_289 5. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_8919 6. https://nptel.ac.in/courses/106108227 7. https://nptel.ac.in/courses/111106102 8. https://nptel.ac.in/courses/111106113		
Topics relevant to SKILL DEVELOPMENT: The course develops skills in logical reasoning, mathematical modeling, problem-solving, algorithmic thinking, and graph analysis through the study of logic, relations, Boolean algebra, and graph theory. It also enhances computational and analytical skills by applying algebraic structures, formal languages, recursion, and coding techniques to real-world computing problems.		
Catalogue prepared by	Dr. Obbu Ramesh	
Recommended by the Board of Studies on		
Date of Approval by the Academic Council		

Course Code: CSS2258	Course Title: Web Technologies Type of Course : Theory Course (PCC)	L-T-P- C	3 – 0 – 0 - 3
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course highlights the fundamentals of 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 web technologies, enhancing web pages with layout techniques, text formatting, graphics, images, and multimedia. The course focuses on key technologies required to build Internet- and web-based applications that interact with databases and other applications. It also introduces concepts of secure web development, intelligent web applications, and AI-enabled web technologies.		
Course Objective	To develop a strong foundation in web technologies by enabling students to understand internet protocols, design and develop static and dynamic web applications, explore client-side scripting and frameworks, implement server-side programming with database integration, evaluate modern web application architectures, and develop secure and intelligent web applications using AI concepts.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the fundamentals of web technologies, XHTML structure, and web architecture concepts. [Understand] CO2: Apply CSS and advanced styling techniques to design responsive and interactive web pages. [Apply] CO3: Develop dynamic web applications using JavaScript, DOM manipulation, and event handling techniques. [Apply] CO4: Build server-side web applications using PHP and database connectivity concepts. [Apply] CO5: Apply web technologies to develop secure and intelligent web applications using AI concepts. [Apply]		
Course Content:			
Module 1	Introduction to XHTML	10 Sessions	
Topics: Basics: Web, WWW, Web browsers, Web servers, Internet. XHTML: Origins and Evolution of HTML and XHTML: Basic Syntax, Standard XHTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, Frames, Syntactic Differences between HTML and XHTML.			
Module 2	Advanced CSS	10 Sessions	
Topics: CSS: Introduction to CSS, Defining & Applying a style, Creating style sheets, types of style sheet, selectors, CSS font properties, border properties, Box model, opacity, CSS pseudo class and pseudo-elements Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Responsive Design, CSS Frameworks XML: Basics, demonstration of applications using XML			
Module 3	Fundamentals of JavaScript	10 Sessions	
Topics: JavaScript: Introduction to JavaScript, Basic JavaScript Instructions, Functions, Methods & Objects, Decisions and Loops, Document Object Model, Event handling, handling window pop-ups, JavaScript validation.			
Module 4	PHP – Application Level	08 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.			

Module 5	Web Security and AI Applications	07 Sessions
Web Security Basics, Secure Web Development, Input Validation and Session Handling, Secure PHP and Database Access, Introduction to AI in Web Applications, AI-based Chatbots, Intelligent User Interfaces, Smart Web Applications.		
Targeted Application & Tools that can be used: Xampp web server to be used to demonstrate PHP.		
Project work/Assignment:		
Assignments are given after completion of each module which the student need to submit within the stipulated deadline.		
Textbook(s): T1. Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 8th Edition, 2015. T2. CSS Notes for Professionals, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022) T3. Deitel, Deitel, Goldberg, "Internet & World Wide Web How to Program", Fifth Edition, Pearson Education, 2021.		
References: R1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", Pearson Education India, 1st Edition, 2016. R2. Jeffrey C. Jackson, "Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition, 2016.		
Topics related to development of "FOUNDATION": Web, WWW, Web browsers, Web servers, Internet. CSS, PHP. Designing for healthcare. For Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.		
E-References: 1. W3schools.com 2. Developer.mozilla.org/en-US/docs/Learn 3. docs.microsoft.com 4. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking 5. https://presiuniv.knimbus.com/user#/home		

Course Code: CSS2259	Course Title: Web Technologies Lab Type of Course: Laboratory Course (PCC)	L- T-P- C	0-0-2-1
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course highlights the fundamentals of 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 web technologies, enhancing web pages with layout techniques, text formatting, graphics, images, and multimedia. The course focuses on key technologies required to build Internet- and web-based applications that interact with databases and other applications. It also introduces concepts of secure web development, intelligent web applications, and AI-enabled web technologies.		
Course Objective	To develop a strong foundation in web technologies by enabling students to understand internet protocols, design and develop static and dynamic web applications, explore client-side scripting and frameworks, implement server-side programming with database integration, evaluate modern web application architectures, and develop secure and intelligent web applications using AI concepts.		

Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the fundamentals of web technologies, XHTML structure, and web architecture concepts. [Understand] CO2: Apply CSS and advanced styling techniques to design responsive and interactive web pages. [Apply] CO3: Develop dynamic web applications using JavaScript, DOM manipulation, and event handling techniques. [Apply] CO4: Build server-side web applications using PHP and database connectivity concepts. [Apply] CO5: Apply web technologies to develop secure and intelligent web applications using AI concepts. [Apply]
COURSE CONTENT:	
List of Laboratory Tasks:	30 Sessions
WEEK 1: Introduction to Web & XHTML Lab Sheet 1: Web Basics Level 1: Create a simple XHTML page explaining Web, WWW, browsers, and servers. Level 2: Design a webpage illustrating Internet architecture using text and images. Lab Sheet 2: XHTML Structure Level 1: Create a basic XHTML document with proper syntax and structure. Level 2: Design a webpage using headings, paragraphs, and formatting tags. WEEK 2: XHTML Elements Lab Sheet 3: Lists and Links Level 1: Create ordered, unordered, and definition lists in XHTML. Level 2: Create hyperlinks for navigation between multiple pages. Lab Sheet 4: Images and Tables Level 1: Insert images into a webpage. Level 2: Design a table with rows, columns, and formatting. WEEK 3: XHTML Forms & Frames Lab Sheet 5: Forms Level 1: Create a form with text fields and buttons. Level 2: Design a registration form with validation requirements. Lab Sheet 6: Frames Level 1: Create a webpage using frames. Level 2: Design a multi-frame layout for a website. WEEK 4: Introduction to CSS Lab Sheet 7: CSS Basics Level 1: Apply inline and internal CSS styles. Level 2: Create an external stylesheet and apply it to multiple pages. Lab Sheet 8: Selectors Level 1: Use element, class, and ID selectors. Level 2: Apply multiple selectors for styling complex pages. WEEK 5: CSS Properties Lab Sheet 9: Font and Text Level 1: Apply font properties (size, style, family). Level 2: Design a styled webpage using text formatting. Lab Sheet 10: Box Model Level 1: Demonstrate margin, padding, and border. Level 2: Design layout using box model concepts. WEEK 6: Advanced CSS Lab Sheet 11: Positioning Level 1: Use static and relative positioning. Level 2: Design layout using absolute and fixed positioning. Lab Sheet 12: Floating & Layout	

- Level 1: Apply float properties.
- Level 2: Create a multi-column layout using float.

WEEK 7: Responsive Design

Lab Sheet 13: Responsive Web Design

- Level 1: Use media queries.
- Level 2: Design a responsive webpage.

Lab Sheet 14: CSS Frameworks

- Level 1: Use a CSS framework (Bootstrap basics).
- Level 2: Design responsive webpage using Bootstrap.

WEEK 8: XML Basics

Lab Sheet 15: XML Structure

- Level 1: Create a simple XML document.
- Level 2: Display XML data in webpage.

Lab Sheet 16: XML Applications

- Level 1: Validate XML structure.
- Level 2: Use XML for data representation in web pages.

WEEK 9: JavaScript Basics

Lab Sheet 17: JavaScript Introduction

- Level 1: Write basic JavaScript program for output.
- Level 2: Perform arithmetic operations using JavaScript.

Lab Sheet 18: Control Statements

- Level 1: Use if-else statements.
- Level 2: Implement loops in JavaScript.

WEEK 10: Functions & DOM

Lab Sheet 19: Functions

- Level 1: Create and call functions.
- Level 2: Use functions with parameters and return values.

Lab Sheet 20: DOM Manipulation

- Level 1: Access HTML elements using DOM.
- Level 2: Modify content dynamically using JavaScript.

WEEK 11: Events & Validation

Lab Sheet 21: Event Handling

- Level 1: Handle click events.
- Level 2: Create interactive webpage using events.

Lab Sheet 22: Form Validation

- Level 1: Validate input fields using JavaScript.
- Level 2: Create complete form validation system.

WEEK 12: Advanced JavaScript

Lab Sheet 23: Objects & Methods

- Level 1: Create JavaScript objects.
- Level 2: Use methods and object properties.

Lab Sheet 24: Pop-ups

- Level 1: Implement alert and prompt.
- Level 2: Create dynamic pop-up-based interaction.

WEEK 13: PHP Basics

Lab Sheet 25: PHP Introduction

- Level 1: Write basic PHP script.
- Level 2: Handle form input using PHP.

Lab Sheet 26: PHP Control Structures

- Level 1: Use if-else in PHP.
- Level 2: Implement loops in PHP.

WEEK 14: PHP Applications Lab Sheet 27: PHP Forms Level 1: Process form data using PHP. Level 2: Create login system using PHP. Lab Sheet 28: PHP File Handling Level 1: Read/write files using PHP. Level 2: Store and retrieve data dynamically. WEEK 15: Integration & Mini Project Lab Sheet 29: Web Integration Level 1: Combine XHTML, CSS, JavaScript. Level 2: Build dynamic webpage using all technologies. Lab Sheet 30: Mini Project Level 1: Design a complete website. Level 2: Develop full web application using XHTML, CSS, JS, PHP. WEEK 15.1: Secure and Intelligent Web Applications Lab Sheet 31: Web Security and Smart Interfaces Level 1: Implement secure form validation and session handling in web applications. Level 2: Develop an intelligent web interface with chatbot or recommendation features. Lab Sheet 32: AI-based Web Mini Project Level 1: Design a secure responsive web application. Level 2: Develop a smart web application integrating XHTML, CSS, JavaScript, PHP, and basic AI concepts. Targeted Application & Tools that can be used: Xampp web server to be used to demonstrate PHP. Project work/Assignment: Assignments are given after completion of each module which the student need to submit within the stipulated deadline. Textbook(s): T1. Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 8th Edition, 2015. T2. CSS Notes for Professionals, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022) T3. Deitel, Deitel, Goldberg, "Internet & World Wide Web How to Program", Fifth Edition, Pearson Education, 2021. References: R1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", Pearson Education India, 1st Edition, 2016. R2. Jeffrey C. Jackson, "Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition, 2016. E-References: 1. W3schools.com 2. Developer.mozilla.org/en-US/docs/Learn 3. docs.microsoft.com 4. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking 5. https://presiuniv.knimbus.com/user#/home Topics related to development of "FOUNDATION": Web, WWW, Web browsers, Web servers, Internet. CSS, PHP. Designing for healthcare For Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.
--

Course Code: CSS2259	Course Title: Web Technologies Lab Type of Course: Laboratory Course (PCC)	L- T-P- C	0-0-2-1
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course highlights the fundamentals of 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 web technologies, enhancing web pages with layout techniques, text formatting, graphics, images, and multimedia. The course focuses on key technologies required to build Internet- and web-based applications that interact with databases and other applications. It also introduces concepts of secure web development, intelligent web applications, and AI-enabled web technologies.		
Course Objective	To develop a strong foundation in web technologies by enabling students to understand internet protocols, design and develop static and dynamic web applications, explore client-side scripting and frameworks, implement server-side programming with database integration, evaluate modern web application architectures, and develop secure and intelligent web applications using AI concepts.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the fundamentals of web technologies, XHTML structure, and web architecture concepts. [Understand] CO2: Apply CSS and advanced styling techniques to design responsive and interactive web pages. [Apply] CO3: Develop dynamic web applications using JavaScript, DOM manipulation, and event handling techniques. [Apply] CO4: Build server-side web applications using PHP and database connectivity concepts. [Apply] CO5: Apply web technologies to develop secure and intelligent web applications using AI concepts. [Apply]		
COURSE CONTENT:			
List of Laboratory Tasks:		30 Sessions	
WEEK 1: Introduction to Web & XHTML			
Lab Sheet 1: Web Basics			
Level 1: Create a simple XHTML page explaining Web, WWW, browsers, and servers.			
Level 2: Design a webpage illustrating Internet architecture using text and images.			
Lab Sheet 2: XHTML Structure			
Level 1: Create a basic XHTML document with proper syntax and structure.			
Level 2: Design a webpage using headings, paragraphs, and formatting tags.			
WEEK 2: XHTML Elements			
Lab Sheet 3: Lists and Links			
Level 1: Create ordered, unordered, and definition lists in XHTML.			
Level 2: Create hyperlinks for navigation between multiple pages.			
Lab Sheet 4: Images and Tables			
Level 1: Insert images into a webpage.			
Level 2: Design a table with rows, columns, and formatting.			
WEEK 3: XHTML Forms & Frames			
Lab Sheet 5: Forms			
Level 1: Create a form with text fields and buttons.			
Level 2: Design a registration form with validation requirements.			

Lab Sheet 6: Frames

Level 1: Create a webpage using frames.

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

WEEK 4: Introduction to CSS

Lab Sheet 7: CSS Basics

Level 1: Apply inline and internal CSS styles.

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

Lab Sheet 8: Selectors

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

Level 2: Apply multiple selectors for styling complex pages.

WEEK 5: CSS Properties

Lab Sheet 9: Font and Text

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

Level 2: Design a styled webpage using text formatting.

Lab Sheet 10: Box Model

Level 1: Demonstrate margin, padding, and border.

Level 2: Design layout using box model concepts.

WEEK 6: Advanced CSS

Lab Sheet 11: Positioning

Level 1: Use static and relative positioning.

Level 2: Design layout using absolute and fixed positioning.

Lab Sheet 12: Floating & Layout

Level 1: Apply float properties.

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

WEEK 7: Responsive Design

Lab Sheet 13: Responsive Web Design

Level 1: Use media queries.

Level 2: Design a responsive webpage.

Lab Sheet 14: CSS Frameworks

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

Level 2: Design responsive webpage using Bootstrap.

WEEK 8: XML Basics

Lab Sheet 15: XML Structure

Level 1: Create a simple XML document.

Level 2: Display XML data in webpage.

Lab Sheet 16: XML Applications

Level 1: Validate XML structure.

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

WEEK 9: JavaScript Basics

Lab Sheet 17: JavaScript Introduction

Level 1: Write basic JavaScript program for output.

Level 2: Perform arithmetic operations using JavaScript.

Lab Sheet 18: Control Statements

Level 1: Use if-else statements.

Level 2: Implement loops in JavaScript.

WEEK 10: Functions & DOM

Lab Sheet 19: Functions

Level 1: Create and call functions.

Level 2: Use functions with parameters and return values.

Lab Sheet 20: DOM Manipulation

Level 1: Access HTML elements using DOM.

Level 2: Modify content dynamically using JavaScript.

WEEK 11: Events & Validation Lab Sheet 21: Event Handling Level 1: Handle click events. Level 2: Create interactive webpage using events. Lab Sheet 22: Form Validation Level 1: Validate input fields using JavaScript. Level 2: Create complete form validation system. WEEK 12: Advanced JavaScript Lab Sheet 23: Objects & Methods Level 1: Create JavaScript objects. Level 2: Use methods and object properties. Lab Sheet 24: Pop-ups Level 1: Implement alert and prompt. Level 2: Create dynamic pop-up-based interaction. WEEK 13: PHP Basics Lab Sheet 25: PHP Introduction Level 1: Write basic PHP script. Level 2: Handle form input using PHP. Lab Sheet 26: PHP Control Structures Level 1: Use if-else in PHP. Level 2: Implement loops in PHP. WEEK 14: PHP Applications Lab Sheet 27: PHP Forms Level 1: Process form data using PHP. Level 2: Create login system using PHP. Lab Sheet 28: PHP File Handling Level 1: Read/write files using PHP. Level 2: Store and retrieve data dynamically. WEEK 15: Integration & Mini Project Lab Sheet 29: Web Integration Level 1: Combine XHTML, CSS, JavaScript. Level 2: Build dynamic webpage using all technologies. Lab Sheet 30: Mini Project Level 1: Design a complete website. Level 2: Develop full web application using XHTML, CSS, JS, PHP. WEEK 15.1: Secure and Intelligent Web Applications Lab Sheet 31: Web Security and Smart Interfaces Level 1: Implement secure form validation and session handling in web applications. Level 2: Develop an intelligent web interface with chatbot or recommendation features. Lab Sheet 32: AI-based Web Mini Project Level 1: Design a secure responsive web application. Level 2: Develop a smart web application integrating XHTML, CSS, JavaScript, PHP, and basic AI concepts. Targeted Application & Tools that can be used: Xampp web server to be used to demonstrate PHP. Project work/Assignment: Assignments are given after completion of each module which the student need to submit within the stipulated deadline. Textbook(s):

T1. Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 8th Edition, 2015. T2. CSS Notes for Professionals, ebook available at https://books.goalkicker.com/CSSBook/ (Retrieved on Jan. 20, 2022) T3. Deitel, Deitel, Goldberg, "Internet & World Wide Web How to Program", Fifth Edition, Pearson Education, 2021.
References: R1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", Pearson Education India, 1st Edition, 2016. R2. Jeffrey C. Jackson, "Web Technologies: A Computer Science Perspective", Pearson Education, 1st Edition, 2016. E-References: 2. W3schools.com 3. Developer.mozilla.org/en-US/docs/Learn 4. docs.microsoft.com 5. informit.com/articles/ The Relationship Between Web 2.0 and Social Networking 6. https://presiuniv.knimbus.com/user#/home
Topics related to development of "FOUNDATION": Web, WWW, Web browsers, Web servers, Internet. CSS, PHP. Designing for healthcare For Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CSS2260	Course Title: Database Management Systems Type of Course: Theory Course (PCC)	L-T-P-C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the foundational principles of database management systems, including data models, schemas, and architectures. This course provides a solid foundation on the relational model of data and the use of relational algebra. It develops skills in SQL for data definition, manipulation, and control, enabling students to construct and execute complex queries. The course also introduces the concept of object oriented and object relational databases and modern database technologies like NoSQL. The also course allows the students to gain insights into data storage structures and indexing strategies for optimizing query performance.		
Course Objective	To develop a strong foundation in database management systems by enabling students to understand database concepts and architecture, design relational databases using ER modeling, apply normalization techniques to eliminate redundancy, execute SQL queries and transactions, and analyze query optimization and database performance.		

Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the fundamental concepts of database systems, data models, schemas, and their role in modern applications. [Understand] CO2: Apply relational algebra operations to retrieve and manipulate data in relational databases. [Apply] CO3: Apply SQL queries and analyze query processing and optimization techniques. [Apply] CO4: Design and develop database systems by applying normalization principles, and demonstrate transaction processing, recovery, and security mechanisms. [Apply] CO5: Explain advanced database concepts and their applications in real-world scenarios. [Understand]	
Course Content:		
Module 1	Introduction to Database Modelling and Relational Algebra	10 Sessions
Topics: Introduction to Database: Schema, Instance, 3-shema architecture, physical and logical data independence, Data isolation problem in traditional file system, advantages of database over traditional file systems. Entity Relationship (ER) Model, ER Model to Relational Model, Examples on ER model. Relational Algebra with selection, projection, rename, set operations, Cartesian product, joins (inner and outer joins), and division operator. Examples on Relational Algebra Operations.		
Module 2	Fundamentals of SQL and Query Optimization	10 Sessions
Topics: SQL Database Querying, DDL, DML, Constraints, Operators, Set Operators, Aggregate Functions, Joins, Views, Procedures, Functions and Triggers. Database programming issues and techniques: Embedded SQL, Dynamic SQL; SQL / PSM and NoSQL. Query Optimization: Purpose, transformation of relational expressions, estimating cost and statistics of expression, choosing evaluation plans, linear and bushy plans, dynamic programming algorithms.		
Module 3	Relational Database Design & Transaction Management	11 Sessions
Topics: Relational database design: Problems in schema design, redundancy and anomalies, Normal Forms based on Primary Keys-(1NF,2NF, 3NF), Boyce-Codd Normal Form, Multi valued Dependency (Fourth Normal Form), Join Dependencies (Fifth Normal Form), lossy and lossless decompositions, Database De-normalization. Transaction Management: The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; The write-ahead log protocol; Check pointing; Recovering from a System Crash; Media Recovery; Other approaches and interaction with concurrency control.		

Module 4	Advanced DBMS Topics, AI and Blockchain in Databases	14 Sessions
Topics: Advanced topics: Object oriented database management systems, Deductive database management systems, Spatial database management systems, Temporal database management systems, Constraint database management systems. New database applications and architectures such as Data warehousing, Multimedia, Mobility, NoSQL, Native XML databases (NXD), Document-oriented databases, Statistical databases. AI-driven query optimization techniques, AI-assisted anomaly and fraud detection in database transactions, Introduction to Blockchain databases, Blockchain transactions and consensus mechanisms.		
Targeted Application & Tools that can be used: Application Area: Relational database systems for Business, Scientific and Engineering Applications. Tools/Simulator used: MySQL DB for student practice. Also demonstration of ORACLE DB on object-relational database creation and JDBC connection.		
Assignments: 1. Problem Solving: Constructing ER-Diagrams for a given real time requirements, Normalizing the databases, querying the databases using relational algebra. 2. Programming: Implementation of any given scenario using MySQL.		
Text Books: T1. Elmasri R and Navathe S B, "Fundamentals of Database System", Pearson Publication, 7th Edition, 2018. T2. RamaKrishna & Gehrke, "Database Management Systems" 3rd Edition, 2018, McGraw-Hill Education. T3. W. Lemahieu, S. vanden Broucke and B. Baesens, "Principles of Database Management: Practical Guide to Storing, Managing and Analyzing Big and Small Data", Cambridge University Press, 2018.		
References: R1 Avi Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill, 7th Edition, 2019. R2 M. Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems", O'Reilly, 2017.		
Topics relevant to development of "FOUNDATION SKILLS": S - Skill Development: Relational database design using ER-Relational mapping, Implementation of given database scenario using MYSQLDB. Topics relevant to development of Employability: Develop, test and implement computer databases, creating sophisticated, interactive and secure database applications Topics relevant to "HUMAN VALUES & PROFESSIONAL ETHICS": Nil		

Course Code: CSS2261	Course Title: Database Management Systems Lab Type of Course: Laboratory Course (PCC)	L-T-P-C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The Database Management Systems (DBMS) Laboratory is designed to provide students with hands-on experience in database design, implementation, and management using SQL and database management tools such as MySQL. The lab complements theoretical concepts learned in database courses by allowing students to practice database creation, querying, and optimization techniques. The DBMS Lab enables students to develop industry-relevant skills in database management, preparing them for careers in software development, data engineering, and database administration.		
Course Objective	To develop practical skills in database systems by enabling students to design and implement relational database schemas, perform SQL queries and operations, implement constraints, triggers, and stored procedures, and develop database-driven applications.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Create databases, tables, and constraints using SQL Data Definition Language (DDL) commands. [Understand] CO2: Apply SQL Data Manipulation Language (DML) commands to insert, update, delete, and retrieve data from relational databases. [Apply] CO3: Implement advanced SQL queries using joins, subqueries, views, indexes, and aggregate functions to solve real-world problems. [Apply] CO4: Design and normalize relational schemas and enforce integrity constraints using keys, triggers, and stored procedures. [Apply] CO5: Perform database transaction management, including concurrency control, backup, recovery, and security mechanisms using DBMS tools. [Apply]		
Course Content:			
List of Laboratory Tasks:		30 Sessions	
WEEK 1: Introduction to Database Concepts			
Lab Sheet 1: DBMS Basics			
Level 1: Identify schema and instance for a given database scenario.			
Level 2: Analyze differences between traditional file systems and DBMS with examples.			
Lab Sheet 2: Database Architecture			
Level 1: Explain 3-schema architecture with diagram.			
Level 2: Illustrate physical and logical data independence using examples.			

WEEK 2: ER Modeling

Lab Sheet 3: ER Diagram Basics

Level 1: Draw ER diagram for a student database.

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

Lab Sheet 4: ER to Relational Mapping

Level 1: Convert ER diagram into relational schema.

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

WEEK 3: Relational Algebra

Lab Sheet 5: Basic Operations

Level 1: Perform selection and projection operations.

Level 2: Apply rename and set operations.

Lab Sheet 6: Advanced Operations

Level 1: Perform joins (inner join).

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

WEEK 4: SQL Basics

Lab Sheet 7: DDL Commands

Level 1: Create and modify tables using SQL.

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

Lab Sheet 8: DML Commands

Level 1: Insert, update, delete records.

Level 2: Perform queries with conditions and operators.

WEEK 5: SQL Queries

Lab Sheet 9: Joins

Level 1: Write SQL queries using joins.

Level 2: Solve complex queries using multiple joins.

Lab Sheet 10: Aggregate Functions

Level 1: Use COUNT, SUM, AVG.

Level 2: Solve grouping queries with HAVING clause.

WEEK 6: Advanced SQL

Lab Sheet 11: Views

Level 1: Create and use views.

Level 2: Update data using views.

Lab Sheet 12: Procedures & Functions

Level 1: Write stored procedures.

Level 2: Implement user-defined functions.

WEEK 7: Triggers & Embedded SQL

Lab Sheet 13: Triggers

Level 1: Create trigger for insert/update.

Level 2: Design trigger for auditing changes.

Lab Sheet 14: Embedded & Dynamic SQL

Level 1: Demonstrate embedded SQL.

Level 2: Implement dynamic SQL queries.

WEEK 8: Query Optimization

Lab Sheet 15: Query Optimization Basics

Level 1: Explain query transformation techniques.

Level 2: Estimate cost of queries.

Lab Sheet 16: Query Plans

Level 1: Analyze linear execution plans.

Level 2: Compare bushy vs linear plans.

WEEK 9: Normalization

Lab Sheet 17: Normal Forms

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

Level 2: Convert schema to 3NF and BCNF.

Lab Sheet 18: Advanced Normalization

Level 1: Apply 4NF and 5NF.

Level 2: Analyze lossy vs lossless decomposition.

WEEK 10: Transactions

Lab Sheet 19: ACID Properties

Level 1: Explain ACID with examples.

Level 2: Analyze transaction schedules.

Lab Sheet 20: Concurrency Control

Level 1: Implement locking mechanisms.

Level 2: Analyze serializability and recoverability.

WEEK 11: Recovery Techniques

Lab Sheet 21: Logging

Level 1: Explain write-ahead logging.

Level 2: Simulate crash recovery.

Lab Sheet 22: Checkpointing

Level 1: Explain checkpointing.

Level 2: Design recovery process after failure.

WEEK 12: Advanced DB Concepts

Lab Sheet 23: NoSQL

Level 1: Compare SQL and NoSQL.

Level 2: Design document-based database.

Lab Sheet 24: Data Warehousing

Level 1: Explain data warehouse concepts.

Level 2: Design simple data warehouse schema.

WEEK 13: Advanced Database Systems

Lab Sheet 25: Object-Oriented DBMS

Level 1: Explain OODBMS concepts.

Level 2: Compare relational vs object-oriented DBMS.

Lab Sheet 26: Spatial & Temporal DB

Level 1: Explain spatial databases.

Level 2: Design temporal database application.

WEEK 14: Specialized Databases

Lab Sheet 27: Multimedia DB

Level 1: Explain multimedia database concepts.

Level 2: Design multimedia database system. Lab Sheet 28: XML Databases Level 1: Explain XML databases. Level 2: Design document-oriented database. WEEK 15: Case Study & Integration Lab Sheet 29: Case Study Level 1: Design database for real-world application. Level 2: Optimize queries and schema. Lab Sheet 30: Mini Project Level 1: Implement database application. Level 2: Integrate SQL, optimization, and transactions. Lab Sheet 31: AI-Based Intelligent Database Query Optimization Level 1: Study AI techniques used in database query optimization. Level 2: Implement an AI-based approach to predict and optimize query execution plans. Lab Sheet 32: Blockchain-Based Secure Database Management Level 1: Study blockchain fundamentals and distributed database concepts. Level 2: Design a blockchain-enabled secure database system for tamper-proof data storage.			
Targeted Application & Tools that can be used: Application Area: Relational database systems for Business, Scientific and Engineering Applications. Tools/Simulator used: MySQL DB for student practice. Also demonstration of ORACLE DB on object-relational database creation and JDBC connection.			
Text Books: T1. Elmasri R and Navathe S B, "Fundamentals of Database System", Pearson Publication, 7th Edition, 2018. T2. RamaKrishna & Gehrke, "Database Management Systems" 3rd Edition, 2018, McGraw-Hill Education. T3. W. Lemahieu, S. vanden Broucke and B. Baesens, "Principles of Database Management: Practical Guide to Storing, Managing and Analyzing Big and Small Data", Cambridge University Press, 2018.			
References: R1. Avi Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill ,7th Edition, 2019. R2. M. Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems", O'Reilly, 2017.			
Topics relevant to development of "FOUNDATION SKILLS": S - Skill Development: Relational database design using ER- Relational mapping, Implementation of given database scenario using MySQLDB. Topics relevant to development of Employability: Develop, test and implement computer databases, creating sophisticated, interactive and secure database applications Topics relevant to "HUMAN VALUES & PROFESSIONAL ETHICS": Nil			

Course Code: CDV2501	Course Name: DevOps Tools Internals Type of Course: Theory	L-T- P- C	3-0-0-3
----------------------	---	-----------	---------

Version No.	2.0			
Course Pre-requisites	CDV2000-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 Jenkins. 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 implement the various tools usage and internals practically.			
Course Objective	- Understand the architecture and functionality of key DevOps tools - Learn version control workflows using Git - Explore containerization and configuration management techniques - Implement automated testing and CI/CD pipelines using DevOps tools			
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Apply the features and common Git workflow. (Application) CO2: Practice the filters and plugins to populate, manipulate, and manage data used by Ansible Playbooks. (Application) CO3: Compute the features of selenium IDE. (Application) CO4: Interpret the installation and features of Jenkins and build jobs. (Application)			
Course Content:				
Module 1	Git	Assignment	Quiz on Git Commands	09 Sessions
<u>Topics:</u> 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 Using Docker	Assignment	Quiz on Docker & Ansible Concepts	09 Sessions
<u>Topics:</u>				

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	Assignment	Practical Assignment on Selenium Testing	09 Sessions
<u>Topics:</u> 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	Jenkins Pipeline Implementation	09 Sessions
<u>Topics:</u> 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				
Targeted Application & Tools that can be used: Git & GitHub , Docker , Ansible , Selenium IDE / Selenium WebDriver ,Jenkins Professionally Used Software: Git Version Control System, GitHub / GitLab, Docker Container Platform, Ansible Automation Platform, Selenium Automation Framework, Jenkins CI/CD Server				
Text Book T1. Craig Berg, “ <i>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)</i> ”, Paperback – June 12, 2020. T2. Ferdinando Santacroce, “ <i>Git Essentials</i> ”, Packt Publishing, April 2015, ISBN: 9781785287909 John Ferguson Smart. “ <i>Jenkins: The Definitive Guide</i> ”, O'Reilly Media, Inc., July 2011, ISBN: 9781449305352 T3. John Ferguson Smart, <i>Jenkins: The Definitive Guide</i> , O'Reilly Media, 2011.				
References R1. Jeff Geerling, “ <i>Ansible for DevOps: Server and configuration management for humans</i> ”, Leanpub, August 5, 2020 Unmesh Gundecha, Carl Cocchiaro, “ <i>Learn Selenium</i> ”, Packt Publishing, July 2019, ISBN: 9781838983048 R2. Gaurav Agarwal, “ <i>Modern DevOps Practices: Implement and secure DevOps in the public cloud with cutting- edge tools, tips, tricks, and techniques</i> ”, July 2021. R3. Mikael Krief, “ <i>Learning DevOps: The complete guide to accelerate collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps</i> ”, October 2019 Web Resources				

W1. https://git-scm.com/book/en/v2
W2. https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner
W3. https://www.javatpoint.com/selenium-tutorial
W4. https://www.javatpoint.com/ansible
W5. https://www.tutorialspoint.com/jenkins/jenkins_managing_plugins.htm
W6. https://nptel.ac.in/courses/128106012
Topics relevant to "SKILL DEVELOPMENT": Develop the ability to use industry-standard DevOps tools, automate software deployment workflows, implement CI/CD pipelines, and manage version control and testing automation in modern software development environments.

Course Code: CDV2502	Course Name: DevOps Tools Internals Laboratory Type of Course: Laboratory	L- T-P- C	0-0-2-1
Version No.	2.0		
Course Pre-requisites	CDV2000-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 Jenkins. 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 implement the various tools usage and internals practically.		
Course Objective	- Understand the architecture and functionality of key DevOps tools - Learn version control workflows using Git - Explore containerization and configuration management techniques - Implement automated testing and CI/CD pipelines using DevOps tools		
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Apply the features and common Git workflow. (Application) CO2: Practice the filters and plugins to populate, manipulate, and manage data used by Ansible Playbooks. (Application) CO3: Compute the features of selenium IDE. (Application) CO4: Interpret the installation and features of Jenkins and build jobs. (Application)		

Lab Experiments:	
List of lab experiments: 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 the second 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.txt file, 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 archiving Level 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 Playbook Level 1: Ansible Shell Examples	

Level 2: Execute a Single Command with Ansible Shell Level 2: Execute a Command with Pipe and Redirection

7. Level 1: Run playbook

Level 2: Create the file on the target machines or servers as mentioned in the inventory file and the webserver's group, 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

8. Level 1: Selenium IDE Download and Install

Level 2: Selenium IDE - First Test Case, Login Test and command usage

9. 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 to google.co.in.

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:

1. Setup a Jenkins Job with Apache Ant Build Tool
2. Setup a Jenkins Job with Apache Maven Level 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: Professionally Used Software: Tracking changes in the source code and source code management Automates web browsers Configuration Management and IT automation. Integration of Individual Jobs and Effortless Auditing Tools: Git, Ansible, Selenium and Jenkins
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 implement with the most suitable 2 or 3 antecedents.
Text Book T1. Craig Berg, <i>“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)”</i>, Paperback – June 12, 2020. T2. Ferdinando Santacroce, <i>“Git Essentials”</i>, Packt Publishing, April 2015, ISBN: 9781785287909 John Ferguson Smart. <i>“Jenkins: The Definitive Guide”</i>, O'Reilly Media, Inc., July 2011, ISBN: 9781449305352 T3. John Ferguson Smart, <i>Jenkins: The Definitive Guide</i>, O'Reilly Media, 2011.
References R1. Jeff Geerling, <i>“Ansible for DevOps: Server and configuration management for humans”</i>, Leanpub, August 5, 2020 Unmesh Gundecha, Carl Cocchiaro, <i>“Learn Selenium”</i>, Packt Publishing, July 2019, ISBN: 9781838983048 R2. Gaurav Agarwal, <i>“Modern DevOps Practices: Implement and secure DevOps in the public cloud with cutting- edge tools, tips, tricks, and techniques”</i>, July 2021. R3. Mikael Krief, <i>“Learning DevOps: The complete guide to accelerate collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps”</i>, October 2019
Web Resources W1. https://git-scm.com/book/en/v2 W2. https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner W3. https://www.javatpoint.com/selenium-tutorial W4. https://www.javatpoint.com/ansible W5. https://www.tutorialspoint.com/jenkins/jenkins_managing_plugins.htm W6. https://nptel.ac.in/courses/128106012

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

Course Code: CSS2262	Course Name: Analysis of Algorithms Type of Course: Theory Course (PCC)	L- T-P- C	3-1-0-4
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	Nil		
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of application. The course discusses classic algorithm design approaches such as Divide and Conquer, Dynamic Programming, and Greedy Methods. It also covers strategies for searching solution spaces using Backtracking and Branch-and-Bound techniques. The course emphasizes the analysis of algorithms and their classification into various complexity classes. In addition, it introduces emerging concepts in AI-based optimization, intelligent search techniques, and algorithmic approaches for secure and smart computing applications.		
Course Objective	To develop the ability to design and analyze efficient algorithms by enabling students to evaluate time and space complexity, apply various algorithmic paradigms such as divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound, compare different approaches for solving computational problems, and understand intelligent optimization techniques used in AI-enabled and secure computing systems.		
Course Outcomes	On successful completion of this course, students will be able to: C01: Explain the efficiency of algorithms in terms of time and space complexity. [Understand] C02: Apply divide-and-conquer and decrease-and-conquer techniques to solve searching and sorting problems. [Apply] C03: Apply dynamic programming and greedy techniques to solve computational problems. [Apply] C04: Apply time-space trade-off techniques and analyze the limitations of algorithms for problem solving. [Apply] C05: Analyze problems using backtracking, branch-and-bound, and AI-based optimization techniques to develop optimal solutions. [Analyze]		
Course Content:			
Module 1	Introduction		14 Sessions
Introduction: Notion of algorithm, Importance of algorithm, Important Problem Types, Fundamentals of analysis of algorithm efficiency: General equation, Order of growth, basic efficiency classes, asymptotic notations, mathematical analysis of recursive & non-recursive algorithms, Brute Force technique: Bubble sort, selection sort, linear search, brute force string matching, Exhaustive search: Travelling salesperson problem, knapsack problem.			

Module 2	Divide-and-conquer	11 Sessions
Divide & Conquer technique: General Method, General recursive equation, Masters Theorem, Binary search, Merge sort, quick sort. Decrease & Conquer technique: General method, insertion sort, topological sorting		
Module 3	Dynamic programming	13 Sessions
Dynamic Programming: General method, Warshall algorithm, Floyd algorithm, 0/1 knapsack, Greedy method: General method, prim's algorithm, Kruskal algorithm, Dijkstra's algorithm, Huffman coding		
Module 4	Space and time tradeoff:	11 Sessions
Space and time tradeoff: introduction, Input enhancement in string matching: Horspool algorithm and Boyer-Moore algorithm, Limitations of algorithm power: Introduction, lower bound arguments, decision trees, P, NP and NP-complete problems		
Module 5	Coping with Limitations of Algorithm Power and Intelligent Optimization	11 Sessions
Introduction to limitations of algorithm power, Backtracking: n-Queens Problem, Subset Sum Problem, Hamiltonian Circuit Problem, Branch and Bound: Travelling Salesperson Problem, Knapsack Problem, Job Assignment Problem, AI-based Search and Optimization Techniques, Machine Learning Approaches for Optimization Problems, Intelligent Decision-Making Algorithms, Applications in Secure and Smart Computing Systems.		
Text Book: 1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", 3rd edition, Pearson Education, 2018. 2. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", 4th edition, MIT Press, 2022.		
Reference Books: R1. AV Aho, J Hopcroft, JD Ullman, "The Design and Analysis of Algorithms", Addison-Wesley, 1974 R2. Tim Roughgarden, "Algorithms Illuminated" (books 1 through 3), "Operating Systems Design and Implementation", Soundlikeyourself Publishing, 2017-2019. R3. J. Kleinberg and E. Tardos, "Algorithm Design", Addison-Wesley, 2005		
Online Resources: 1. EBook: https://presiuniv.knimbus.com/user#/home 2. https://onlinecourses.nptel.ac.in/noc26_cs67/ 3. www.youtube.com/watch?v=so4zT-u09LM&list=PL8GCet93Cu1iRF611mrUu-Rhy12fMwZO		
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:	Course Title: Analysis of Algorithms Lab		0-0-2-1
CSS2263	Type of Course: Laboratory Course (PCC)	L- T-P- C	
Version No.	1.0		
Course Pre-requisites	NIL		

Anti-requisites	NIL
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of application. The course discusses classic algorithm design approaches such as Divide and Conquer, Dynamic Programming, and Greedy Methods. It also covers strategies for searching solution spaces using Backtracking and Branch-and-Bound techniques. The course emphasizes the analysis of algorithms and their classification into various complexity classes. In addition, it introduces emerging concepts in AI-based optimization, intelligent search techniques, and algorithmic approaches for secure and smart computing applications.
Course Objective	To develop the ability to design and analyze efficient algorithms by enabling students to evaluate time and space complexity, apply various algorithmic paradigms such as divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound, compare different approaches for solving computational problems, and understand intelligent optimization techniques used in AI-enabled and secure computing systems.
Course Outcomes	On successful completion of this course, students will be able to: C01: Compute the efficiency of algorithms in terms of time and space complexity. [Apply] C02: Apply divide-and-conquer techniques to solve searching and sorting problems. [Apply] C03: Apply dynamic programming techniques to solve computational problems. [Apply] C04: Apply greedy techniques for solving optimization problems. [Apply] C05: Analyze problems using backtracking, branch-and-bound, and AI-based optimization techniques to develop optimal solutions. [Analyze]
Course Content	
List of Laboratory Tasks: 30 Sessions WEEK 1: Algorithm Basics & Analysis Lab Sheet 1: Algorithm Fundamentals Level 1: Write an algorithm and program to find minimum and maximum elements in a list. Level 2: Analyze the number of comparisons and derive time complexity. Lab Sheet 2: Asymptotic Notations Level 1: Write programs to demonstrate linear and quadratic time complexity. Level 2: Compare growth rates of different functions using Big-O, Ω, and Θ. WEEK 2: Efficiency Analysis Lab Sheet 3: Recursive Algorithms	

Level 1: Write a recursive program for factorial.

Level 2: Solve recurrence relations for recursive algorithms.

Lab Sheet 4: Non-Recursive Analysis

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

Level 2: Derive general equations and order of growth.

WEEK 3: Brute Force Techniques

Lab Sheet 5: Sorting Algorithms

Level 1: Implement Bubble Sort and Selection Sort.

Level 2: Compare their time complexity experimentally.

Lab Sheet 6: Searching & String Matching

Level 1: Implement Linear Search.

Level 2: Implement brute-force string matching algorithm.

WEEK 4: Exhaustive Search

Lab Sheet 7: Knapsack Problem

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

Level 2: Analyze solution space complexity.

Lab Sheet 8: Travelling Salesperson Problem

Level 1: Solve TSP using brute force approach.

Level 2: Compare performance for different inputs.

WEEK 5: Divide and Conquer

Lab Sheet 9: Binary Search

Level 1: Implement Binary Search.

Level 2: Analyze recurrence relation.

Lab Sheet 10: Merge Sort

Level 1: Implement Merge Sort.

Level 2: Analyze using Master's Theorem.

WEEK 6: Advanced Divide & Conquer

Lab Sheet 11: Quick Sort

Level 1: Implement Quick Sort.

Level 2: Compare best and worst cases.

Lab Sheet 12: Master's Theorem

Level 1: Solve recurrence equations.

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

WEEK 7: Decrease and Conquer

Lab Sheet 13: Insertion Sort

Level 1: Implement Insertion Sort.

Level 2: Compare with Bubble Sort.

Lab Sheet 14: Topological Sorting

Level 1: Implement Topological Sort.

Level 2: Apply on real-world dependency graph.

WEEK 8: Dynamic Programming

Lab Sheet 15: Warshall Algorithm

Level 1: Implement Warshall algorithm.

Level 2: Compute transitive closure.

Lab Sheet 16: Floyd Algorithm

Level 1: Implement Floyd's algorithm.

Level 2: Compare with Dijkstra.

WEEK 9: DP Applications

Lab Sheet 17: Knapsack DP

Level 1: Solve 0/1 Knapsack using DP.

Level 2: Compare with brute force approach.

Lab Sheet 18: DP Optimization

Level 1: Identify overlapping subproblems.

Level 2: Convert recursive solution to DP.

WEEK 10: Greedy Algorithms

Lab Sheet 19: MST Algorithms

Level 1: Implement Prim's algorithm.

Level 2: Implement Kruskal's algorithm.

Lab Sheet 20: Shortest Path

Level 1: Implement Dijkstra's algorithm.

Level 2: Compare greedy vs DP approach.

WEEK 11: Huffman Coding

Lab Sheet 21: Huffman Tree

Level 1: Construct Huffman Tree.

Level 2: Analyze compression efficiency.

Lab Sheet 22: Greedy Analysis

Level 1: Demonstrate greedy choice property.

Level 2: Prove correctness of greedy algorithms.

WEEK 12: String Matching Optimization

Lab Sheet 23: Horspool Algorithm

Level 1: Implement Horspool algorithm.

Level 2: Compare with brute-force method.

Lab Sheet 24: Boyer-Moore Algorithm

Level 1: Implement Boyer-Moore algorithm.

Level 2: Analyze efficiency improvements.

WEEK 13: Complexity Theory

Lab Sheet 25: Lower Bounds

Level 1: Analyze decision tree for sorting.

Level 2: Derive lower bounds.

Lab Sheet 26: NP Problems

Level 1: Classify problems as P or NP.

Level 2: Analyze NP-complete problems.

WEEK 14: Backtracking

Lab Sheet 27: N-Queens

Level 1: Solve N-Queens problem.

Level 2: Analyze search tree.

Lab Sheet 28: Subset Sum

Level 1: Solve subset sum problem.

Level 2: Compare with DP solution.

WEEK 15: Branch & Bound

Lab Sheet 29: TSP

Level 1: Solve TSP using Branch & Bound.

Level 2: Compare with brute-force approach.

Lab Sheet 30: Job Assignment

Level 1: Solve assignment problem.

Level 2: Analyze pruning efficiency.

WEEK 15.1: AI-Based Optimization and Intelligent Search

Lab Sheet 31: Intelligent Optimization Techniques

Level 1: Implement AI-based heuristic search for optimization problems.

Level 2: Compare heuristic search with Branch & Bound approach for performance analysis.

Lab Sheet 32: Smart Decision-Making Applications

Level 1: Implement a simple recommendation or path optimization system using graph/search techniques.

Level 2: Analyze intelligent pruning and optimization efficiency in real-world problem solving.

Targeted Application & Tools that can be used :

PyTorch/Jupyter Notebook – For Python programming

Text Book :

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

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

References :

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

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

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

R4. Donald E. Knuth, *“The Art of Computer Programming”*, Volumes 1 and 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: CSS2264	Course Title: Essentials of AI Type of Course: -Theory Course (ESC)	L-T-P-C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the fundamentals of Artificial Intelligence, AI tools, and prompt engineering techniques. It covers AI applications, evaluation of AI tools, ethical and responsible AI usage, and the social and economic impacts of AI. The course also introduces advanced AI technologies such as chatbots, voice assistants, image recognition, robotics, and intelligent automation through practical activities and case studies.		
Course Objectives	The course aims to introduce the fundamentals and applications of Artificial Intelligence, popular AI tools, and basic prompt engineering techniques. It also helps students evaluate AI tools, understand ethical and responsible AI usage, and gain knowledge of advanced AI technologies through practical examples and case studies.		
Course Outcomes	On successful completion of this course, students will be able to: CO1:Explain the fundamentals of Artificial Intelligence, its evolution, types, key concepts, and applications across various domains. [Understand] CO2: Apply basic prompt engineering techniques for generating effective AI responses. [Apply] CO3:Evaluate and compare AI tools based on suitability, functionality, ethical considerations, privacy, and security requirements. [Analyze] CO4:Understand the social, ethical, and economic impacts of AI technologies and their influence on society, industries, and employment. [Understand] CO5:Analyze advanced AI technologies and their real-world applications through case studies and practical scenarios. [Analyze]		
Course Content:			
Module 1	Introduction to Artificial Intelligence		9 Sessions
Topics: Introduction to Artificial Intelligence – Definition and History of AI; Evolution of AI; Goals and Scope of AI; Types of AI – Narrow AI, General AI, Super AI. Key Concepts in AI – Machine Learning, Deep Learning, Neural Networks, Natural Language Processing (NLP), Computer Vision, Generative AI. AI vs Human Intelligence. AI Development Lifecycle. Applications of AI across domains – Healthcare, Education, Finance, Agriculture, Cybersecurity, and E-Commerce.			
Module 2	AI Tools and Effective Prompting Techniques		9 Sessions
Topics: Introduction to Popular AI Tools – Introduction to AI Tools – Overview of popular AI tools and their applications. AI tools for text generation, image generation, predictive analysis, and chatbot interactions. Prompt Engineering-Introduction to prompts and AI-generated responses. Importance of clear			

instructions and effective questioning techniques. Understanding how AI interprets prompts.		
Module 3	Suitability and Evaluation of AI Tools	9 Sessions
Topics: Introduction to AI Tool Evaluation, Evaluating Tool Suitability, AI Tools for Different Purposes, Comparison of AI Tools, Ethical Use of AI, Bias and Fairness in AI, Data Privacy and Security, Ethical Guidelines for Responsible AI Use		
Module 4	Social, Ethical, and Economic Impacts of AI	9 Sessions
Topics: Social, Ethical, and Economic Effects of AI – Ethical issues in AI, future challenges and opportunities, impact of AI on the economy and employment, and responsible use of AI technologies.		
Module 5	Advanced AI Technologies casestudy	9 Sessions
AI Technologies – Voice assistants, image recognition applications, AI chatbots, robotic and intelligent automation systems, recommendation systems, and recent advancements and future trends in AI technologies.		
Targeted Application & Tools that can be used: - Implementation of AI-based content generation and chatbot interaction using prompt engineering techniques. Tools: ChatGPT, Google Gemini, Hugging Face - Implementation of image generation and creative AI applications using generative AI tools. Tools: Canva AI, DALL-E, Adobe Firefly - Implementation of predictive analytics and machine learning models for simple real-world datasets. Tools: Google Colab, TensorFlow, Kaggle - Implementation of NLP-based applications such as sentiment analysis, text summarization, and translation. Tools: NLTK, spaCy, Hugging Face Transformers - Development of AI-based mini projects for solving real-world problems in healthcare, education, finance, agriculture, and cybersecurity. Tools: IBM Watson, Microsoft Azure AI, Teachable Machine		
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course Group project on one of the topics mentioned above (Eg. Adversarial search).		
Textbook(s): - Stuart Russell and Peter Norvig, <i>Artificial Intelligence: A Modern Approach</i>, 4th Edition, Pearson Education, 2022. - Elaine Rich and Kevin Knight, <i>Artificial Intelligence</i>, 3rd Edition, McGraw Hill Education, 2009. - Tom Taulli, <i>Artificial Intelligence Basics: A Non-Technical Introduction</i>, 1st Edition, Apress, 2019.		
References: - Ethem Alpaydin, <i>Introduction to Machine Learning</i>, 4th Edition, MIT Press, 2020. - Ian Goodfellow, Yoshua Bengio, Aaron Courville, <i>Deep Learning</i>, 1st Edition, MIT Press, 2016. - Melanie Mitchell, <i>Artificial Intelligence: A Guide for Thinking Humans</i>, 1st Edition, Farrar, Straus and Giroux, 2019. - Pedro Domingos, <i>The Master Algorithm</i>, 1st Edition, Basic Books, 2015. - Kai-Fu Lee, <i>AI Superpowers</i>, 1st Edition, Houghton Mifflin Harcourt, 2018.		
Weblinks OpenAI Google AI IBM Artificial Intelligence Microsoft AI Learning Coursera – AI Courses NPTEL – Introduction to Artificial Intelligence NPTEL – Artificial Intelligence: Search Methods for Problem Solving		

Course Code: CSS2265	Course Title: Essentials of AI Lab Type of Course: Laboratory Course (PCC)	L-T-P-C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces students to the fundamentals and practical applications of Artificial Intelligence through hands-on laboratory exercises. Students will explore AI tools, prompt engineering, generative AI, Natural Language Processing (NLP), basics of Machine Learning, chatbot development, recommendation systems, ethical AI, and voice assistant simulation. The course concludes with the development of AI-based mini projects for real-world applications.		
Course Objectives	To develop practical skills in Artificial Intelligence by enabling students to explore AI tools, apply prompt engineering techniques, implement basic NLP and Machine Learning tasks, develop intelligent AI-based applications, and analyze ethical and real-world aspects of AI systems.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Apply AI tools and prompt engineering techniques for generating effective AI-based outputs. [Apply] CO2: Implement AI-based applications for text generation, image generation, and chatbot development. [Apply] CO3: Apply Natural Language Processing and basic Machine Learning techniques for solving simple real-world problems. [Apply] CO4: Analyze datasets and AI model outputs using preprocessing, visualization, and evaluation techniques. [Analyze] CO5: Analyze ethical issues, bias, and the impact of AI systems in practical applications. [Analyze]		
Course Content:			
Lab Sheet 1: Introduction to AI Tools ● Problem Statement: ● Explore different AI platforms such as ChatGPT, Gemini, and Hugging Face, and compare their outputs for various tasks based on relevance and accuracy. Lab Sheet 2: Prompt Engineering Basics ● (Using AI tools explored in Lab 1) ● Problem Statement: ● Design effective prompts for text generation and summarization, and analyze how prompt structure influences AI-generated responses. Lab Sheet 3: Advanced Prompt Engineering ● (Continuing Lab 2) ● Problem Statement:			

- Develop role-based and instruction-based prompts to optimize AI outputs for accuracy, detail, and contextual understanding.

Lab Sheet 4: AI Content Generation

- *(Using prompts designed in previous labs)*
- Problem Statement:
- Generate blogs, emails, resumes, and reports using AI tools and improve the outputs through iterative prompt refinement.

Lab Sheet 5: AI Image Generation

- *(Extending prompt engineering concepts)*
- Problem Statement:
- Generate AI-based posters, logos, and advertisements using generative AI tools and compare outputs generated using different prompt styles.

Lab Sheet 6: Dataset Collection & Preprocessing

- Problem Statement:
- Collect textual datasets from reviews or social media platforms and perform preprocessing techniques such as tokenization, stemming, and stopword removal.

Lab Sheet 7: Sentiment Analysis using NLP

- *(Using dataset prepared in Lab 6)*
- Problem Statement:
- Implement sentiment analysis on textual datasets using NLP libraries and analyze sentiment classification performance.

Lab Sheet 8: Text Summarization & Translation

- *(Continuing NLP tasks from Lab 6 & 7)*
- Problem Statement:
- Implement AI-based text summarization and language translation techniques and compare generated outputs with original content.

Lab Sheet 9: Introduction to Machine Learning

- *(Using processed datasets from previous labs)*
- Problem Statement:
- Understand supervised and unsupervised learning concepts by loading, splitting, training, and testing datasets using Scikit-learn.

Lab Sheet 10: Data Visualization & Feature Analysis

- *(Continuing Lab 9 using same dataset)*
- Problem Statement:
- Visualize datasets using Matplotlib and analyze feature relationships and data distributions using graphical techniques.

Lab Sheet 11: AI Chatbot Development

- *(Using NLP and prompt engineering concepts)*
- Problem Statement:
- Develop a rule-based chatbot using Python and integrate contextual AI-generated responses for interactive communication.

Lab Sheet 12: Recommendation System

- *(Using prediction concepts from previous labs)*
- Problem Statement:
- Implement a content-based recommendation system and analyze recommendation accuracy using similarity metrics.

Lab Sheet 13: Ethical AI and Bias Analysis

- *(Evaluating outputs generated in previous labs)*
- Problem Statement:
- Identify bias, hallucination, and ethical issues in AI-generated outputs and analyze their impact on responsible AI systems.

Lab Sheet 14: Voice Assistant Simulation

- *(Using NLP and chatbot concepts from previous labs)*
- Problem Statement:
- Develop a basic voice assistant using Python speech libraries and implement voice commands for search, greetings, and simple automation tasks.

Lab Sheet 15: AI Mini Project

- *(Combining concepts from all previous labs)*
- Problem Statement:
- Develop and demonstrate an AI-based application integrating concepts such as NLP, machine learning, chatbot systems, or recommendation systems.

Targeted Application & Tools that can be used:

6. ChatGPT, Gemini, Hugging Face, Canva AI, Python
7. Python IDEs like PyCharm-Jupyter Notebook /Google Collab

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

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

Textbook(s):

1. Stuart Russel and Peter Norvig. Artificial Intelligence: A Modern Approach. 4th Edition. Pearson Education. 2022.
2. Prateek Joshi and Alberto Artasanchez. Artificial Intelligence with Python. 2nd Edition. Packt. 2020.
3. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 2022

References:

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

Course Code: FIN1002	Course Title: Essentials of Finance Type of Course: HSMC		L-T-P-C	3-0-0-3
Version No.	1.0			
Course Pre-requisites	This course is designed to be accessible to all students, regardless of their prior financial knowledge.			
Anti-requisites				
Course Description	This course is designed to equip students with a foundational understanding of key financial concepts and principles. It will enable them to comprehend the core functions of finance, delve into the intricacies of financial management within organizations, and gain insights into the fundamental aspects of taxation. The course aims to develop students' abilities to interpret financial statements, evaluate investment opportunities, understand capital structure decisions, and navigate the basics of tax implications.			
Course Objective	Upon successful completion of this course, students will be able to: Understand the basic forms of business organization and their financial implications. Understand the fundamental principles and concepts that influence financial decision-making in various contexts. Analyse and interpret financial statements to assess the financial health and performance of an organization. Identify income under various heads of income as per Income Tax Act, 1961 and determine the tax liability.			
Course Outcomes	List the course outcomes On successful completion of this course the students shall be able to: Understand the basic concepts of finance and financial markets and organizations. Apply and interpret financial information for business decision making. Identify various heads of income and deduction under Income Tax Act, 1961.			
Course Content:				
Module 1	Introduction to Finance	Assignment/ Quiz	Numerical solving Task	10 Sessions
Definition and Scope of Finance, Areas of Finance: Corporate Finance, Investments, Financial Institutions, International Finance; Types of Financial Markets: Money Markets vs. Capital Markets, Primary vs. Secondary Markets; Forms of Business Organization and Financial Goals: Shareholder Wealth Maximization vs. Profit Maximization; Understanding Financial Statements: Balance Sheet and Income Statement- Simple Numerical.				

Module 2	Financial Management	Assignment/ Quiz	Numerical solving Task	18 Sessions
Capital Budgeting Decisions: Payback Period, Net Present Value (NPV), Profitability Index (PI), Internal Rate of Return (IRR); Leverage- Basic Numerical; Capital Structure Decisions: Optimal Capital Structure, Trade-off Theory of Capital Structure; Cost of Capital: Equity, Debt, WACC; Dividend Policy: Factors influencing Dividend Policy.				
Module 3	Taxation	Assignment/ Quiz	Numerical solving Task	17 Sessions
Principles of a Good Tax System: Equity, Certainty, Convenience, Economy; Direct vs. Indirect Taxes; Residential Status of an Individual- Basic Problems; Heads of Income; Salary, House Property- Basic Numerical; Deductions under Chapter VI-A; Computation of Taxable Income and Tax Liability; E-Filing procedure.				
Targeted Application & Tools that can be used: Textbooks, PPT, Spreadsheet Software (e.g., Microsoft Excel), Official Website of Income Tax Department.				
Project Work/ Assignment: 1. Presentation: There will be a group presentation, where the students will be given a topic. They will have to explain/demonstrate the working and discuss the applications for the same. 2. Case Study: - At the end of the course students will be given a 'real-world' cases like business models of successful companies or tax evasion by reputed companies on which they have to come up with detailed analysis and assessment.				
Text Book(s): Dr. Vinod K. Singhania & Dr. Monica Singhania. (Latest Assessment Year Edition). Students' Guide to Income Tax including GST. Taxmann Publications. Pandey, I. M. (2025). Financial Management. Vikas Publishing House.				
Reference Book (s): Bhole, L.M., & Mahakud, J. (Current Edition). Financial Institutions and Markets: Structure, Growth and Innovations. McGraw Hill Education India. Mehrotra, H.C., & Goyal, S.P. (Latest Assessment Year Edition). Income Tax Law & Practice. Sahitya Bhawan Publications. Gordon, E., & Natarajan, K. (Current Edition). Financial Markets and Services. Himalaya Publishing House. Online Resources (e-books, notes, ppts, video lectures etc.): https://presidencyuniversity.linways.com https://onlinecourses.nptel.ac.in/noc24_ec01/preview https://www.incometax.gov.in/iec/foportal/				
Topics relevant to "SKILL DEVELOPMENT": This course is designed to provide practical financial skills through participative learning techniques. Students will engage in performing suitable calculations to				

determine financial parameters (e.g., time value of money, investment returns, tax liabilities) and analysing financial statements to assess organizational performance and make informed decisions.

Course Code: APT4004	Course Title: Aptitude Training-Intermediate Type of Course: Practical Only Course	L- T - P- C	0-0-2-0
Version No.	1.0		
Course Pre-requisites	Students should have the basic concepts of Quantitative aptitude along with its applications in real life problems.		
Anti-requisites	NIL		
Course Description	This is a skill-based training program for the students. This course is designed to enable the students to enhance their skills in Quantitative Aptitude.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Aptitude and attain Skill Development through Problem Solving techniques.		
Course OutComs	On successful completion of this course the students shall be able to: CO1: Recall all the basic mathematical concepts. CO2: Identify the principle concept needed in a question. CO3: Solve the quantitative and logical ability questions with the appropriate concept. CO4: Analyze the data given in complex problems.		
Course Content:			
Module 1	Quantitative Ability 1		
Topics: Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss			
Module 2	Quantitative Ability 2		
Topics: Time Speed and Distance, Boats and Streams, Simple Interest, Compound Interest, Probability, Permutation and Combination			
Targeted Application & Tools that can be used:Application area: Placement activities and Competitive examinations. Tools: LMS			
Continuous Evaluation:			
CA1 – Online Test CA2 – Online Test CA3 – Online Test			

Assignm ent
Text Book: Fast Track Objective by Rajesh Verma R S Aggarwal Rakesh Yadav References: www.indiabix.com www.testbook.com www.youtube.com/c/TheAptitudeGuy/videos
Topics relevant to Skill Development: Quantitative aptitude for Skill Development through Problem solving Techniques. This is attained through components mentioned in course handout.

Course Code: CSS2266	Course Title: Theory of Computation Type of Course: Theory Course (PCC)	L- T-P- C	3-0-0-3
Version No.	2.0		
Course Pre- requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the theoretical foundations of computation through the study of formal languages and automata. It covers the design and analysis of finite automata and regular languages, context-free grammar and pushdown automata, and Turing machines as models of computation. The course further explores key concepts such as ambiguity, normal forms, computability, decidability, and complexity theory including classes like P and NP. It also highlights the relevance of Theory of Computation in modern computing domains such as artificial intelligence, natural language processing, cybersecurity, compilers, and intelligent systems.		
Course Objective	To build a strong foundation in formal languages and automata by enabling students to model and analyze computational systems using finite automata, pushdown automata, and Turing machines; to evaluate computability and complexity of problems; and to apply theoretical concepts to modern domains such as AI, cybersecurity, NLP, and intelligent systems.		
Course Outcomes	On successful completion of this course, students will be able to: C01: Explain fundamental concepts of automata, formal languages, and finite state machines. [Understand] C02: Design and analyze deterministic and nondeterministic finite automata and regular expressions for language recognition. [Apply]		

	C03: Construct and evaluate context-free grammar and pushdown automata, including ambiguity removal and normal forms. [Apply] C04: Design Turing machines and interpret their role in solving computational problems and language recognition. [Apply] C05: Explain the applications of Theory of Computation concepts in real-world domains such as AI/ML, NLP, cybersecurity, compilers, and robotics. [Understand]	
Course Content:		
Module 1	Introduction to automata theory	7 Sessions
Topics: Introduction to Automata Theory, Applications of Automata Theory, Alphabets, Strings, Languages & operations on languages, Representation of automata, Language recognizers, Finite State Machines (FSM): Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, Basic concepts of Finite automata, DFA- definitions of DFA, Deterministic Accepters Transition Graphs and Languages, and DFA's		
Module 2	Finite Automata and Regular languages	12 Sessions
Topics: Regular Languages, NFA- Definition of a Nondeterministic Acceptor, Languages and NFA's Why Non-determinism? Equivalence of Deterministic and Nondeterministic Finite Accepters, Reduction of the Number of States in Finite Automata. Formal Definition of a Regular Expression, Languages Associated with Regular Expressions, Languages, Regular Languages (RL) and Non-regular Languages: Closure properties of RLs, to show some languages are not RLs, Closure Properties of Regular		
Module 3	Context Free Grammar and Pushdown automata	12 Sessions
Topics: Context Free Grammars-Examples of Context-Free Languages, Leftmost and Rightmost Derivations, Derivation Trees, Relation Between Sentential Forms and Derivation Trees, Ambiguity in Grammars and Languages: Ambiguous Grammars, Removing Ambiguity, Chomsky Normal Form, Gribiche Normal Form. Definition of a Pushdown Automaton, Language Accepted by a Pushdown Automaton, Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack Equivalence of PDA's and CFG's: From Grammars to Pushdown Automata.		
Module 4	Turing Machines	06 Sessions
Topics: Definition of a Turing Machine, Turing Machines as Language Accepters, Example Languages to construct Turing machine, Turing Machines as Transducers, Halting Programming Techniques for Turing Machines		
Module 5	Complexity and Modern Applications of ToC	08 Sessions
Topics: Complexity Theory - Time and Space Complexity, Complexity Classes: P, NP, NP-Complete (introductory), Intractability and Hard Problems-Applications of ToC in AI/ML/DL - Computability and Learnability (basic idea), Role of algorithms and complexity in ML, Expressive power of neural networks, NLP & Compilers -Regular Languages in Tokenization, CFG in Syntax Parsing, Ambiguity in Natural Language Processing-Cybersecurity & Blockchain- Hardness assumptions in cryptography, Automata in intrusion detection systems, Smart contracts as state machines, Decidability limits in security Robotics & Intelligent Systems - Finite State Machines in robot control, Planning as a computational problem, Real-time decision constraints		
Targeted Application & Tools that can be used:		

Targeted Application: 1. Text Processing 2. Compilers 3. Text Editors 4. Robotics Applications 5. Artificial Intelligence Tools: 1. JFLAP (Java Formal Language and Automata Package) Software simulation tool. Its interactive educational software written in Java to experiment topics in automata theory. 2. Turing machine Online simulators.
Text Book(s): 1. Peter Linz, "An introduction to Formal Languages and Automata", Jones and Bartlett Publications 6th Ed, 2018. 2. Michael Sipser, "Introduction to the Theory of Computation", Cengage Learning, 3rd Ed, 2012
Reference(s): 1. Aho, Ullman and Hopcroft, "Theory of Computation", Pearson India 3rd Edition 2008. 2. Michael Sipser, "Theory of Computation", Cengage India 3rd Ed, 2014.
E Resources 1. NPTEL 2. Opencourseware 3. TOC Notes

Course Code: CDV2001	Course Name: Agile Structures and Frameworks Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	Software Engineering		
Anti-requisites	NIL		
Course Description	This course introduces the principles, values, and frameworks of Agile software development. It focuses on iterative development, adaptive planning, continuous delivery, and collaborative approaches used in modern software engineering. Students will explore widely used Agile methodologies such as Scrum, Extreme Programming (XP), Feature Driven Development (FDD), and Unified Process. The course also emphasizes Agile estimation, quality assurance practices, and Agile testing techniques supported by industry tools.		

Course Objective	- Understand the core principles and values of Agile software development - Explore major Agile frameworks such as Scrum, XP, and FDD - Analyze differences between traditional and Agile software development approaches - Apply Agile practices for project planning, estimation, and quality assurance			
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Understand the basic concepts of Agile Software Process. (Knowledge level) CO2: Comprehend the various Agile Methodologies. (Comprehension level) CO3: Develop Agile Software Process. (Knowledge level) CO4: Apply principles of Agile Testing. (Application level)			
Course Content:				
Module 1	Introduction	Assignment	Agile Estimation	08 Sessions
<u>Topics:</u> 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 technologies with traditional methods	09 Sessions
<u>Topics:</u> 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 product roles and practices.				
Module 3	Agile methodology	Assignment	Case Study	12 Sessions
<u>Topics:</u> 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 using Programming	09 Sessions

<u>Topics:</u> 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.
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course 1. Agile Estimation 2. 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 4. Installation and features of JIRA tool.
Targeted Application & Tools that can be used: JIRA
Professionally Used Software: JIRA Software, Azure DevOps, GitHub / GitLab, Trello Agile Boards
Text Book T1. Kenneth S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison-Wesley. T2. Robert C. Martin, Agile Software Development: Principles, Patterns, and Practices, Pearson Education.
References R1. Mike Cohn, Agile Estimating and Planning, Prentice Hall. R2. Kent Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley. R3. Alistair Cockburn, Agile Software Development, Addison-Wesley. R4. Jeff Sutherland, Scrum: The Art of Doing Twice the Work in Half the Time, Crown Business. Web Resources W1. https://www.agilealliance.org W2. https://www.scrum.org W3. https://www.atlassian.com/agile W4. https://www.mountangoatsoftware.com/agile W5. https://www.scrumguides.org
Topics relevant to “SKILL DEVELOPMENT”: Develop the ability to plan and manage Agile software projects, perform sprint planning and backlog prioritization, collaborate effectively in Agile teams, and apply Agile testing and continuous improvement practices in software development.

Course Code: CSS2269	Course Name: Operating Systems Type of Course: Theory Course (PCC)	L-T- P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the concepts of Operating Systems operations, Operating System structure and its design and implementation. It covers the Operating Systems internal algorithms such as process scheduling, synchronization, deadlocks detection and recovery and memory management. The course also enhances the problem solving, systems programming ability and case studies.		
Course Objective	To develop a comprehensive understanding of operating system principles by enabling students to analyze process management and scheduling algorithms, explore memory and virtual memory management, examine file systems and storage organization and evaluate synchronization mechanisms and deadlock handling techniques.		
Course Outcomes	On successful completion of this course, students will be able to: C01: Explain the fundamental concepts of operating systems and relevant case studies. [Understand] C02: Apply various CPU scheduling algorithms for process management. [Apply] C03: Apply synchronization techniques to handle concurrency problems. [Apply] C04: Apply deadlock detection and recovery methods in operating systems. [Apply] C05: Explain various memory management techniques used in operating systems. [Apply]		
Course Content:			
Module 1	Introduction to Operating Systems	9 Sessions	
Topics: Introduction to OS , Operating-System Operations, Operating Systems Services, , System Calls and its types, Operating Systems Structure, System Program and its types, Linkers and Loaders, Overview of OS design and implementation, Open-source Operating Systems			
Module 2	Process Management	11 Sessions	
Topics:			

Process Concept, Operations on Processes, Inter Process Communication, Communication in client-server systems (sockets, RPC, Pipes), Introduction to threads - Multithreading Models, Thread Libraries, Threading Issues, Process Scheduling- Basic concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF, RR and Priority.		
Module 3	Process Synchronization and Deadlocks	11 Sessions
Topics: The Critical-Section Problem- Peterson's Solution, Synchronization hardware, Semaphores, Classic Problems of Synchronization with Semaphore Solution- Producer-Consumer Problem, Reader-Writer problems, Dining Philosopher's Problem, . Introduction to Deadlocks, Necessary conditions for deadlock, Resource allocation Graph, Methods for handling deadlock: Deadlock Prevention and Implementation, Deadlock Avoidance and Implementation, Deadlock detection & Recovery from Deadlock.		
Module 4	Memory Management , AI and OS Security	14 Sessions
Topics: Introduction to Memory Management, Basic hardware-Base and Limit Registers, Memory Management Unit(MMU), Dynamic loading and linking, Swapping, Contiguous and Non-Contiguous Memory Allocation, Segmentation, Paging - Structure of the Page Table – Virtual Memory and Demand Paging – Page Faults and Page Replacement Algorithms, Copy-on-write, Allocation of Frames, Thrashing Introduction to File system management: File System Interface (access methods, directory structures), File system implementation. AI-assisted resource allocation and intelligent memory optimization, Machine Learning based CPU and memory usage prediction, Malware, ransomware and rootkit protection in OS, File system security and access permissions.		
Targeted Application: Application area is traffic management system, banking system, health care and many more systems where in there are resources and entities that use and manage the resources.		
Software Tools: Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]. Used to install and work on multiple guest Operating Systems on top of a host OS. Intel Processor identification utility: This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.		
Project work/Assignment :		
1. Demonstrate process concepts in LINUX OS. 2. Simulation of CPU scheduling algorithms. 3. Develop program to demonstrate use of Semaphores in threads. 4. Develop program to demonstrate use of deadlock avoidance algorithms. 5. Develop program to demonstrate use of page replacement algorithms. 6. Simulation of memory allocation strategies [first fit, best fit and worst fit].		

Text Book :
1. Silberschatz A, Galvin P B and Gagne G , “Silberschatz's Operating System Concepts”, Paperback, Global Edition Wiley, 2019
References :
1. Silberschatz A, Galvin P B and Gagne G, “Operating System Concepts”, 10th edition Wiley, 2018.
2. William Stallings, “Operating Systems”, Ninth Edition, By Pearson Paperback ,1 March 2018.
3. Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, “ Cracking the Operating System skills”, Dreamtech, paperback, 2020
4. Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , “Operating Systems: Three Easy Pieces, Amazon digital Services”, September 2018.
E-resources/Weblinks :
1. https://www.os-book.com/OS9/
2. https://pages.cs.wisc.edu/~remzi/OSTEP/
3. https://codex.cs.yale.edu/avi/os-book/OS10/index.html

Course Code:	Course Title: Operating Systems Lab	L-T- P- C	0-0-2-1
CSS2270	Type of Course: Laboratory Course (PCC)		
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This laboratory course provides hands-on experience with the core concepts of operating systems through practical assignments, simulations, and case studies. It covers foundational aspects such as system calls, process and thread management, inter-process communication, synchronization, deadlocks, memory management, and file systems. Students will implement and simulate real-time OS components and scheduling algorithms, fostering deeper understanding of OS architecture and design. The lab also introduces modern OS tools, programming interfaces, and the basics of open-source OS environments.		
Course Objectives	To develop practical skills in operating systems by enabling students to implement scheduling and memory management algorithms, simulate process synchronization and deadlock handling, and build system-level programming capabilities.		
Course Outcomes	On successful completion of this course, students will be able to: CO1: Demonstrate system-level programming using system calls and operating system structures. [Apply] CO2: Simulate process scheduling and multithreading techniques. [Apply] CO3: Apply synchronization techniques using semaphores and shared memory to handle concurrency problems. [Apply]		

	C04: Demonstrate memory management and file system concepts using simulation or scripting tools. [Apply] C05: Analyze the performance of operating system algorithms related to scheduling, synchronization and memory management. [Analyze]
Course Content:	
List of Laboratory Tasks: 30 Sessions WEEK 1: Introduction to OS Lab Sheet 1: OS Basics Level 1: Write a program to demonstrate basic system calls (file operations like open, read, write). Level 2: Analyze different OS services with real-time examples. Lab Sheet 2: OS Structure Level 1: Illustrate OS structures (monolithic, layered, microkernel). Level 2: Compare open-source OS (Linux) with proprietary OS. WEEK 2: System Programs Lab Sheet 3: Linkers & Loaders Level 1: Demonstrate compilation process (preprocessing, compiling, linking). Level 2: Analyze object files and executable generation. Lab Sheet 4: System Utilities Level 1: Write simple shell commands using system calls. Level 2: Implement a mini shell program. WEEK 3: Process Concept Lab Sheet 5: Process Creation Level 1: Write a program using fork() system call. Level 2: Demonstrate parent-child communication. Lab Sheet 6: Process Operations Level 1: Implement process states and transitions. Level 2: Simulate process control block (PCB). WEEK 4: Inter Process Communication	

Lab Sheet 7: Pipes

Level 1: Implement communication using pipes.

Level 2: Implement bidirectional communication using pipes.

Lab Sheet 8: Sockets/RPC

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

Level 2: Simulate RPC communication.

WEEK 5: Threads

Lab Sheet 9: Thread Creation

Level 1: Create threads using pthread library.

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

Lab Sheet 10: Thread Issues

Level 1: Demonstrate thread synchronization issues.

Level 2: Solve race condition problem.

WEEK 6: CPU Scheduling

Lab Sheet 11: FCFS & SJF

Level 1: Implement FCFS scheduling.

Level 2: Implement SJF and compare results.

Lab Sheet 12: Advanced Scheduling

Level 1: Implement Round Robin scheduling.

Level 2: Implement Priority scheduling and analyze performance.

WEEK 7: Synchronization

Lab Sheet 13: Critical Section

Level 1: Implement Peterson's solution.

Level 2: Analyze critical section problem.

Lab Sheet 14: Semaphores

Level 1: Implement semaphore operations.

Level 2: Solve producer-consumer problem using semaphores.

WEEK 8: Classical Problems

Lab Sheet 15: Reader-Writer

Level 1: Implement Reader-Writer problem.

Level 2: Analyze starvation issues.

Lab Sheet 16: Dining Philosophers

Level 1: Implement Dining Philosophers problem.

Level 2: Avoid deadlock using strategies.

WEEK 9: Deadlocks

Lab Sheet 17: Deadlock Detection

Level 1: Implement Resource Allocation Graph.

Level 2: Detect deadlock using algorithm.

Lab Sheet 18: Deadlock Avoidance

Level 1: Implement Banker's algorithm.

Level 2: Analyze safe and unsafe states.

WEEK 10: Memory Management Basics

Lab Sheet 19: Memory Allocation

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

Level 2: Compare memory utilization.

Lab Sheet 20: Paging

Level 1: Simulate paging mechanism.

Level 2: Analyze page table structure.

WEEK 11: Virtual Memory

Lab Sheet 21: Page Replacement

Level 1: Implement FIFO page replacement.

Level 2: Implement LRU and Optimal algorithms.

Lab Sheet 22: Page Faults

Level 1: Calculate page faults.

Level 2: Analyze thrashing condition.

WEEK 12: Advanced Memory Management

Lab Sheet 23: Segmentation

Level 1: Implement segmentation.

Level 2: Compare segmentation vs paging.

Lab Sheet 24: Demand Paging

Level 1: Simulate demand paging.

Level 2: Analyze copy-on-write mechanism.

WEEK 13: File System

Lab Sheet 25: File Operations

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

Level 2: Simulate directory structure.

Lab Sheet 26: File Allocation

Level 1: Implement file allocation methods.

Level 2: Compare allocation techniques.

WEEK 14: File System Implementation

Lab Sheet 27: File System Design

Level 1: Design simple file system.

Level 2: Analyze file system performance.

Lab Sheet 28: Access Methods

Level 1: Implement sequential access.

Level 2: Implement indexed access.

WEEK 15: Integration & Simulation

Lab Sheet 29: OS Simulation

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

Level 2: Integrate scheduling and memory management.

Lab Sheet 30: Mini Project

Level 1: Implement OS concept-based project.

Level 2: Analyze performance and optimization.

Lab Sheet 31: AI-Based Intrusion Detection System (IDS)

Level 1: Study the basics of Intrusion Detection Systems and identify different types of cyber-attacks.

Level 2: Implement an AI-based anomaly detection system to identify suspicious system activities.

Lab Sheet 32: AI-Based Malware Detection and System Protection

Level 1: Study malware types and basic system protection techniques.

Level 2: Develop an AI-based malware detection model using system behavior analysis.	
Project work/Assignment :	
1. Demonstrate process concepts in LINUX OS. 2. Simulation of CPU scheduling algorithms. 3. Develop program to demonstrate use of Semaphores in threads. 4. Develop program to demonstrate use of deadlock avoidance algorithms. 5. Develop program to demonstrate use of page replacement algorithms. 6. Simulation of memory allocation strategies [first fit, best fit and worst fit].	
Targeted Application: Application area is traffic management system, banking system, health care and many more systems where in there are resources and entities that use and manage the resources.	
Software Tools: 1. Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]. Used to install and work on multiple guest Operating Systems on top of a host OS. 2. Intel Processor identification utility: This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.	
Text Book : 1. Silberschatz A, Galvin P B and Gagne G , “Silberschatz's Operating System Concepts”, Paperback, Global Edition Wiley, 2019	
References : 1. Silberschatz A, Galvin P B and Gagne G, “Operating System Concepts”, 10th edition Wiley, 2018. 2. William Stallings, “Operating Systems”, Ninth Edition, By Pearson Paperback, 1 March 2018. 3. Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, “ Cracking the Operating System skills”, Dreamtech, paperback, 2020 4. Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , “Operating Systems: Three Easy Pieces, Amazon digital Services”, September 2018. E-resources/Weblinks 1. https://www.os-book.com/OS9/ 2. https://pages.cs.wisc.edu/~remzi/OSTEP/ 3. https://codex.cs.yale.edu/avi/os-book/OS10/index.html	

Course Code: CDV2503	Course Name: Software Testing Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
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 Objective	- 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	On successful completion of the course the students shall be able to: CO1: Understand various software testing levels, techniques, and models used in industry. (Understand) CO2: Analyze software testing strategies and defect tracking mechanisms. (Apply) CO3: Apply white-box and black-box testing techniques to evaluate software functionality and performance. (Apply) CO4: Implement and manage automated test scripts using industry tools. (Apply)			
Course Content:				
Module 1	Fundamentals of Software Testing	Assignment	Conceptual Analysis	11 Sessions
<u>Topics:</u> 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	Assignment	Test Case Design Exercise	11 Sessions
<u>Topics:</u> 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	Assignment	Automation Testing Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Performance Testing Case Study	12 Sessions
<u>Topics:</u> Performance Testing (JMeter), Load Testing, Security Testing (OWASP overview), Mutation Testing, Test Data Generation, Model-Based Testing, Introduction to AI-driven Testing Tools.				
Targeted Application & Tools that can be used: Selenium WebDriver, JUnit / NUnit, Apache JMeter, Bug Tracking Systems (JIRA / Bugzilla) Professionally used software : Selenium Automation Framework				
Text Book 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.				
References R1: Ron Patton, <i>Software Testing</i> , Pearson Education, 2nd Edition, 2005. R2: Aditya P. Mathur, <i>Foundations of Software Testing</i> , Pearson Education, 2008. R3: Ilene Burnstein, <i>Practical Software Testing</i> , Springer, 2003. R4: Boris Beizer, <i>Software Testing Techniques</i> , Dreamtech Press, 2nd Edition, 2003 Web Resources W1: https://www.selenium.dev W2: https://junit.org W3: https://jmeter.apache.org W4: https://www.guru99.com/software-testing.html W5: https://owasp.org W6: https://www.browserstack.com/guide/software-testing-tutorial				
Topics relevant to “SKILL DEVELOPMENT”: Develop the ability to design test cases, perform manual and automated testing, identify software defects, and implement testing frameworks to ensure software reliability and quality in real-world development environments.				

Course Code: CDV2504	Course Name: Software Testing Lab Type of Course: Laboratory	L-T- P- C	0-0-2-1
Version No.	2.0		
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 Objective	• 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	On successful completion of the course the students shall 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.		
Lab 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
Targeted Application & Tools that can be used: Manual Testing Tools, Unit Testing Tools, Automation Testing Tools, Mobile Application Testing, Integration / CI Tools, Performance and Load Testing Tools, Code Coverage Tools, Code Quality & Static Analysis, Cross-Browser Testing, Professionally used software : Selenium Automation Framework	
Text Book 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.	
References 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.	
Topics relevant to “SKILL DEVELOPMENT”: Develop practical skills in test case design, automated testing using Selenium, performance testing using JMeter, defect tracking, and test report generation to ensure software quality in real-world development projects.	

Course Code: CSS7000	Course Name: Internship Type of Course: PRW	L- T-P- C	0-0-0-2
Version No.	1.0		
Course Pre- requisites	Knowledge and Skills related to all the courses studied in previous semesters.		
Anti-requisites	NIL		

Course Description	Students observe science and technology in action, develop an awareness of the methods of scientific experimentation, and often get opportunities to see, study, and operate sophisticated and costly equipment. They also get opportunities to work with advanced software tools and modern technologies used in industries and research environments. Students use tools and technologies from emerging fields such as Artificial Intelligence (AI), Machine Learning (ML), Cyber Security, Blockchain, and Data Science to develop solutions for real-world and industry-related problems. They also learn about the implementation of management principles taught in the classroom by observing multidisciplinary teams of experts from engineering, science, economics, operations research, and management dealing with techno-economic problems at both micro and macro levels. The course further enables students to develop and refine their language, communication, teamwork, and interpersonal skills through various activities such as seminars, group discussions, project presentations, and report preparation. The broad-based core education, supported by strong foundations in mathematics, science, and analytical tools, provides the necessary background for students to properly understand and solve practical real-life problems. Students have the flexibility to pursue this course either as Project Work and Dissertation within the university, as Project Work in an Industry, Company, or Research Laboratory, or through Internship Programs in industries and organizations.
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, students shall be able to: CO1: Identify engineering problems related to local, regional, national, and global needs. [Level 2 - Understand] CO2: Apply appropriate techniques and modern tools to solve engineering problems. [Level 3 - Apply] CO3: Design experiments in accordance with standard procedures and specifications. [Level 4- Analyze] CO4: Conduct experiments and collect relevant data for analysis. [Level 3 - Apply] CO5: Interpret experimental results and draw meaningful conclusions. [Level 6 - Create]

Course Code: APT4006	Course Name: Logical and Critical Thinking Type of Course: Audited	L- T- P- C	0-0-2-0
Version No.	1.0		
Course Pre-requisites	Students should have the basic concepts of Logical reasoning and Critical thinking, along with its applications in real life problems.		
Anti-requisites	Nil		
Course Description	This is a skill-based training program for the engineering students (Undergraduate). This course is designed to enable the students to enhance their skills in Logical reasoning and Critical thinking.		
Course Objective	The objective of the course is to familiarize the learners with concepts in Logical reasoning and Critical thinking through problem solving techniques suitable for their career development.		

Course Outcomes	On successful completion of the course the students shall be able to: CO1: Understand all the concepts. Level 2 – Understand CO2: Apply the concepts in problem solving. Level 3 – Apply CO3: Analyze and structure reasoning techniques and spatial visualization skills. Level 4 – Analyze			
Course Content:				
Module 1	Logical Thinking	Assignment		16 Hours
Syllogisms, Cubes and Dices, Mirror and Water images, Paper cutting and Folding, Embedded figures & Completion of figures, Data Interpretation, Data sufficiency				
Module 2	Critical Thinking	Assignment		14 Hours
Analogy, Symbol and Notations, Statement and assumption, Cause of action, Statement and conclusion, Puzzles				
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS				
Evaluation	Continuous Evaluation			
	· Topic wise evaluation			
	· Internal Assessments			
Text Book 1. A new approach to reasoning verbal, non-verbal & analytical by BS Sijwali 2. R S Aggarwal 3. Kiran publications				
References 1. www.indiabix.com 2. www.testbook.com 3. www.youtube.com/c/TheAptitudeGuy/videos				
Topics relevant to Skill Development Logical reasoning and Critical thinking for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.				

Course Code: LAW7601	Course Name: Indian Constitution	L- T-P- C	1-0-0-0
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		

Anti-requisites	NIL								
Course Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PARTICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamentals of Indian Constitution concepts and their relevance to 75+ Years of Republic of India (https://constitution75.com/) as well as #AzaadiKaAmrutMahotsav / Azadi Ka Amrit Mahotsav (https://amritmahotsav.nic.in). It is designed to equip students with the knowledge about the Constitution of India. This course aims to introduce the constitutional law of India to students from all walks of life and help them understand the constitutional principles as applied and understood in everyday life. The objective of making the Constitution of India, familiar to all students, and not only to law students, this course aims and objectifies legal understanding in the simplest of forms. This course is designed to cater to Constitutional Studies.								
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using 'PARTICIPATIVE LEARNING' techniques								
Course Outcomes	On successful completion of the course the students shall be able to: <table border="0"> <tr> <td>CO1: Describe the basic understanding of the Indian Constitution and the concepts and issues relevant to day-to-day life of the nation and equip citizens with the zeal for capacity building.</td><td>Level 2 – Understand</td></tr> <tr> <td>CO2: Recognize and identify the values of the Constitution of India.</td><td>Level 1 – Remember</td></tr> <tr> <td>CO3: Explain citizen-centric awareness of the rights and responsibilities of the State.</td><td>Level 2 – Understand</td></tr> <tr> <td>CO4: Explain the role of State actors in building India.</td><td>Level 2 – Understand</td></tr> </table>	CO1: Describe the basic understanding of the Indian Constitution and the concepts and issues relevant to day-to-day life of the nation and equip citizens with the zeal for capacity building.	Level 2 – Understand	CO2: Recognize and identify the values of the Constitution of India.	Level 1 – Remember	CO3: Explain citizen-centric awareness of the rights and responsibilities of the State.	Level 2 – Understand	CO4: Explain the role of State actors in building India.	Level 2 – Understand
CO1: Describe the basic understanding of the Indian Constitution and the concepts and issues relevant to day-to-day life of the nation and equip citizens with the zeal for capacity building.	Level 2 – Understand								
CO2: Recognize and identify the values of the Constitution of India.	Level 1 – Remember								
CO3: Explain citizen-centric awareness of the rights and responsibilities of the State.	Level 2 – Understand								
CO4: Explain the role of State actors in building India.	Level 2 – Understand								
Course Content:									
Module 1	Understanding the Making of the Constitution: The Constituent Assembly & The Constitution of India								
Topics: Historical Context of Constituent Assembly - Compositions & Functions of Constituent Assembly What is a Constitution? – Why have a Constitution? – Constitutional Change - Features of Indian Constitution – Preamble of Indian Constitution									
Module 2	Citizen's Fundamental Rights and State's Responsibilities (Directive Principles)								
Topics: Introduction to Fundamental Rights - Right to Equality – Facets of Right to Equality - Right to Freedom - Constitutional Position of Some Democratic Rights - Right Against Exploitation - Right to Freedom of Religion - Right to Constitutional Remedies Directive Principles of the State Policy									
Module 3	Organs Of the Government								
Topics: Executive: The President of India - Powers and Functions of President of India - Emergency Powers and the Position of the President									

Legislature: Union Council of Ministers - Prime Minister - The Rajya Sabha - The Lok Sabha - Relation between the Lok Sabha & Rajya Sabha - Office of the Speaker – Important Parliamentary Committees Judiciary: The Structure and Organization of the Judiciary & the High Court - The Supreme Court - Role of The Supreme Court - Judicial Activism in India - Basic Structure Doctrine & PIL	
Module 4	Federalism & Decentralization
Topics: What is Federalism? - Centre-State Legislative Relations - Centre-State Administrative Relations - Centre-State Financial Relations The 5th & 6th Schedules - Municipality- (History of Indian Municipality, Organization & Functions) – Panchayat 1 (Idea of Panchayat, Organization and Powers of Panchayats in India)	
Targeted Application & Tools that can be used: Application areas to familiarize students with fundamentals of Indian Constitutional concepts. Tools: Online Tools – NPTEL and Swayam.	
Project work/Assignment:	
Assessment Type Online end term exam will be conducted as notified by the Presidency University.	
Online Link*: 1) Prof. Amitabha Ray, SWAYAM Course: “Constitutional Government & Democracy in India” https://onlinecourses.swayam2.ac.in/cec19_hs13/preview * Other source links are available in below Resources link. Text Book 1. Durga Das Basu --- Introduction to the Constitution of India, 23rd Edition (Gurgaon; LexisNexis, 2018). 2. MP Jain’s Constitutional Law of India, Lexis Nexis 3. V.N Shukla’s Indian Constitutional Law, M.P Singh 13th Edition 4. MV Pylee’s Constitution of India 5. J.C.Johari -- The Constitution of India: A Politico-Legal Study (Greater Noida: Sterling Publishers Pvt. Ltd. 2013). 6. Himangshu Roy and M.P.Singh – Indian Political System, 4th Edition (Bengaluru; Pearson Education, 2018) 7. Vidya Bhushan & Vishnool Bhagwan--- Indian Administration (S. Chand, 2011) 8. S.R.Maheswari --- Indian Administration (Orient Blackswan, 2001) 9. Dr. A.Avasthi & A.P. Avasthi --- Indian Administration (L.N. Agarwal Educational Publishing, 2017). 10. B. L. Fadia --- Indian Government and Politics (Sahitya a. Bhawan, 13th Revised Edition, 2017). 11. P.M.Bakshi – The Constitution of India (Prayagraj, UP; a. Universal Law Publishing, January, 2018)	
Reference Books 1. HM Seervai, Constitutional Law of India, 4th Ed. Vol I, II, & III 2. Uday Raj Rai, Constitutional Law-I 3. Democracy and Constitutionalism in India, Oxford University Press 2009 Resources: 1. https://onlinecourses.nptel.ac.in/noc20_lw03/course?&force_user=true 2. https://onlinecourses.swayam2.ac.in/cec19_hs13/course?&force_user=true 3. https://nptel.ac.in/courses/129106003 4. https://nptel.ac.in/courses/129106411	

5. <https://nptel.ac.in/courses/129105608>
 6. <https://nptel.ac.in/courses/129106002>

Topics relevant to Skill Development:

1. An attitude of inquiry.
2. Write reports

The topics related to Constitutional Studies and its application :

All topics in theory component are relevant to Indian Constitution.

Course Code: CDV2508	Course Name: Continuous Integration and Continuous Delivery (CI/CD) Pipelines Type of Course: Theory	L- T-P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	CDV2501 – DevOps Tools and Internals		
Anti-requisites	NIL		
Course Description	This course provides in-depth knowledge of CI/CD principles, tools, and practices used in modern software development. It covers the entire pipeline from code commit to deployment, enabling faster, reliable, and automated software delivery.		
Course Objective	• Understand the principles and lifecycle of CI/CD in DevOps • Learn to design, implement, and monitor CI/CD pipelines • Explore popular CI/CD tools such as Jenkins, GitLab CI, GitHub Actions, and others • Implement build automation, test integration, and secure deployments • Integrate code quality checks, containerization, and orchestration into pipelines		
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Explain the role and benefits of CI/CD in modern DevOps practices (Understand) CO2: Construct automated CI/CD pipelines using industry-standard tools (Apply) CO3: Evaluate pipeline performance and optimize for speed and reliability (Analyze) CO4: Integrate testing, security, and monitoring in deployment pipelines (Apply)		
Course Content:			
Module 1	CI/CD Fundamentals and Pipeline Architecture	Assignment	Conceptual Assignment 10 Sessions
Topics: Introduction to CI/CD, Benefits and challenges, CI/CD lifecycle, Stages in a pipeline: source, build, test, release, deploy, feedback loops, Pipeline architecture patterns.			

Module 2	Tools and Technologies for CI/CD	Assignment	Tool-Based Assignment	12 Sessions
<u>Topics:</u> Overview of Jenkins, GitLab CI, GitHub Actions, CircleCI, TravisCI, Azure DevOps, YAML pipelines, Webhooks and triggers, Version control integration (GitHub/GitLab/Bitbucket).				
Module 3	Pipeline Design, Automation, and Testing	Assignment	Practical Assignment	12 Sessions
<u>Topics:</u> Scripting build stages, Static code analysis tools (SonarQube), Automated testing: unit, integration, functional, Performance testing integration, Secrets management and credential handling.				
Module 4	Containerization, Security, and Monitoring in Pipelines	Assignment	Case Study / Mini Project	11 Sessions
<u>Topics:</u> Docker and Kubernetes integration, Deployment strategies (Blue/Green, Canary), Security practices in pipelines (SAST/DAST), Pipeline observability: logging, metrics, alerts, Case studies: GitOps, CI/CD in cloud-native environments.				
Targeted Application & Tools that can be used: Jenkins, GitLab CI, GitHub Actions, CircleCI, Travis CI, Azure DevOps, Docker, Kubernetes, SonarQube, Nexus Repository, and version control systems such as Git (GitHub, GitLab, Bitbucket). Professionally used software : Jenkins, GitHub Actions, GitLab CI/CD, Azure DevOps, Docker, Kubernetes, SonarQube, CircleCI, and Travis CI				
Text Book T1: Jez Humble & David Farley, <i>Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation</i> , Addison-Wesley T2: Nikhil Pathania, <i>CI/CD Pipeline with Jenkins</i> , Packt Publishing				
References R1: Mohamed Labouardy, <i>Hands-On Continuous Integration and Delivery</i> , Packt R2: Stephen Fleming, <i>Mastering Jenkins</i> , 2023 Edition R3: Viktor Farcic, <i>The DevOps Toolkit: Building the DevOps Pipeline</i> , DevOps Toolkit Series R4: Brent Laster, <i>Jenkins 2: Up and Running</i> , O'Reilly Web Resources W1. https://www.jenkins.io/doc W2. https://docs.gitlab.com/ee/ci W3. https://docs.github.com/actions W4. https://circleci.com/docs W5. https://azure.microsoft.com/en-us/services/devops				

Topics relevant to “SKILL DEVELOPMENT”: Develop practical skills in designing, automating, and managing CI/CD pipelines using modern DevOps tools for efficient and reliable software delivery.

Course Code: CDV2512	Course Name: Continuous Integration and Continuous Delivery (CI/CD) Pipelines Lab Type of Course: Laboratory	L- T-P- C	0-0-4-2
Version No.	2.0		
Course Pre-requisites	CDV2508 – Continuous Integration and Continuous Delivery (CI/CD) Pipelines		
Anti-requisites	NIL		
Course Description	This lab course offers hands-on experience in building, testing, and deploying software through CI/CD pipelines. It reinforces theoretical knowledge with practical implementations using tools like Jenkins, GitLab CI, GitHub Actions, and Docker. Students will learn to automate the software development lifecycle with a focus on continuous integration, automated testing, delivery pipelines, and deployment.		
Course Objective	- Set up and configure CI/CD pipelines using industry-standard tools - Automate build, test, and deployment stages - Integrate code quality, security scans, and monitoring into pipelines - Use containerization tools and orchestrators in the pipeline process - Troubleshoot and optimize pipeline performance		
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Configure and implement CI/CD pipelines using Jenkins, GitLab CI, or GitHub Actions (Apply) CO2: Automate build, test, and deployment processes with version control integration (Apply) CO3: Evaluate pipeline logs, identify bottlenecks, and ensure successful deployments (Analyze) CO4: Design and execute complete CI/CD workflows integrating containerization and cloud deployment tools (Create)		
Lab Experiments:			
1.Introduction to CI/CD tools and Git integration			
2.Install and configure Jenkins for pipeline setup			
3.Create a freestyle Jenkins job for a basic Java project			
4.Build a multistage pipeline using Jenkins (Build → Test → Deploy)			
5.Integrate static code analysis (SonarQube) with CI pipeline			

6. Automate unit and integration testing with JUnit and Maven 7. Containerize a sample application using Docker and deploy via Jenkins 8. Use GitLab CI/CD to build and deploy a web application 9. Implement CI/CD pipeline using GitHub Actions with YAML 10. Setup secrets and environment variables in CI/CD pipelines securely 11. Deploy to staging environment using Docker Compose/Kubernetes 12. Implement Blue/Green deployment strategy using Jenkins pipeline 13. Monitor pipeline status and generate build/test reports 14. Simulate failure and rollback strategies in CI/CD workflows 15. Mini project: Design a full pipeline from code commit to deployment (Cloud Optional)
Targeted Application & Tools that can be used: • Jenkins, GitHub Actions, GitLab CI • Docker, Docker Compose, Kubernetes • SonarQube, JUnit, Selenium • AWS/GCP/Azure (optional for cloud-based deployments) • VS Code, Git, Postman Professionally used software: Jenkins, GitHub Actions, GitLab CI/CD, Docker, Kubernetes, SonarQube, Maven, Selenium, and cloud platforms like AWS, Azure DevOps, and Google Cloud
Text Book T1: Jez Humble & David Farley, <i>Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation</i>, Addison-Wesley
References R1: Nikhil Pathania, <i>CI/CD Pipeline with Jenkins</i>, Packt Publishing R2: Brent Laster, <i>Jenkins 2: Up and Running</i>, O'Reilly R3: Viktor Farcic, <i>The DevOps Toolkit</i>, DevOps Toolkit Series R4: Mohamed Labouardy, <i>Hands-On Continuous Integration and Delivery</i>, Packt Web Resources W1. https://www.jenkins.io/doc/ W2. https://docs.gitlab.com/ee/ci/ W3. https://docs.github.com/en/actions W4. https://hub.docker.com/ W5. https://kubernetes.io/docs/

Topics relevant to “SKILL DEVELOPMENT”: Develop hands-on skills in building, automating, and managing CI/CD pipelines using modern DevOps tools, enabling efficient software integration, testing, and deployment in real-world environments.

Course Code: CDV2510	Course Name: Edge Computing with DevOps Type of Course: Theory	L- T-P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	CDV2000 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 Objective	- 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	On successful completion of the course the students shall be able to: CO1: Explain the fundamentals and significance of edge computing and DevOps in distributed systems (Understand) CO2: Evaluate architectural models and deployment strategies for edge computing using DevOps pipelines (Analyze) CO3: Implement edge-based applications using containerization and orchestration tools (Apply) CO4: Automate CI/CD pipelines and monitor edge-cloud systems using modern DevOps practices (Apply)		

Course Content:				
Module 1	Fundamentals of Edge Computing and DevOps	Assignment	Conceptual Assignment	10 Sessions
<u>Topics:</u> 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	Assignment	Tool-Based Assignment	12 Sessions
<u>Topics:</u> 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	Assignment	Practical Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Case Study /Mini Project	12 Sessions
<u>Topics:</u> 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.				
Targeted Application & Tools that can be used: AWS IoT Greengrass, Azure IoT Edge, EdgeX Foundry, Docker, Kubernetes (K3s, MicroK8s), Jenkins, GitHub Actions, GitLab CI, Helm, Prometheus, Grafana, and NVIDIA Jetson for edge AI deployments. Professionally used software: AWS IoT Greengrass, Azure IoT Edge, Docker, Kubernetes, Jenkins, GitHub Actions, Prometheus, Grafana, and EdgeX Foundry				
Text Book 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.				

References 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 Topics relevant to "SKILL DEVELOPMENT": Develop practical skills in designing, deploying, and automating edge-based applications using DevOps tools, enabling real-time processing, low-latency performance, and efficient edge-cloud integration.

Course Code: CSS2274	Course Name: Competitive Programming and Problem Solving Type of Course: Laboratory Course (ESC)	L- T-P- C	0-0-4-2
Version No.	1.0		
Course Pre- requisites	NIL		
Anti-requisites	NIL		
Course Description	This course equips students with efficient problem-solving skills for competitive programming and real-world computational challenges. Starting with brute-force approaches, students learn to optimize time and space complexity using advanced techniques such as greedy algorithms, dynamic programming, backtracking, graph algorithms, and heuristic optimization methods. Hands-on practice using platforms such as CodeChef and Codeforces helps students solve problems related to number theory, data structures, and algorithmic paradigms. The course also introduces intelligent problem-solving strategies, AI-assisted coding concepts, and optimization techniques for competitive programming. By understanding competitive programming constraints and developing a strategic problem-solving mindset, students gain confidence for coding competitions, technical interviews, and real-world applications.		

Course Objective	The objective of the course is to familiarize learners with the concepts of Competitive Programming and Problem Solving, enabling them to develop efficient coding, optimization, and analytical skills through experiential learning techniques and intelligent problem-solving approaches.	
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Understand the challenges of online competitive programming platforms, interpret problem constraints, and understand brute-force solution approaches used in competitive programming. [Understand] CO2: Analyze the time and space complexity of number-theoretic solutions to determine the need for optimization under competitive programming constraints. [Analyze] CO3: Apply appropriate algorithmic paradigms and data structures to solve problems within defined time and memory constraints. [Apply] CO4: Analyze execution failures and identify algorithmic inefficiencies in solutions involving advanced data structure concepts to redesign optimized solutions within given time and memory constraints. [Analyze] CO5: Analyze advanced problem-solving and intelligent optimization techniques using dynamic programming, greedy methods, backtracking, and heuristic approaches for large-scale competitive programming problems. [Analyze]	
Course Content:		
Module 1	Introduction to Competitive Programming	NA
Topics: 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. 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.		
Module 2:	Number Theory for Problem-Solving	NA
Topics: 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	NA
Topics: 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 & Problem Solving using Advanced Topics	NA
Topics: 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. 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.		
Module 5:	Advanced Competitive Programming and Intelligent Problem Solving	NA
Topics: Disjoint Sets and Efficient Representation, Greedy Algorithms, Backtracking Techniques, Dynamic Programming using Bottom-up Approach, Advanced Graph Algorithms for Competitive Programming, AI-assisted Problem Solving Concepts, Heuristic and Optimization Techniques, Intelligent Coding Strategies for Time and Space Optimization.		
List of Practical Tasks		60 Sessions
WEEK 1 Lab Sheet 1: Sorting & Ranking Level 1: Write a program to find the runner who finished in the third position from the given finishing times of N runners. Level 2: Write a program to find the top 10 runners along with their bib numbers efficiently for large datasets. Lab Sheet 2: Searching Techniques Level 1: Write a program to check whether a given book ID exists in a library database. Level 2: Design and implement a system to search books efficiently by title using an appropriate data structure. WEEK 2 Lab Sheet 3: Cost Calculation Level 1: Write a program to calculate the total cost of products in a shopping cart. Level 2: Design an algorithm to apply multiple discount rules and compute the final bill. Lab Sheet 4: Modular Arithmetic Level 1: Write a program to compute $(a^m \mod p)$. Level 2: Implement binary exponentiation for efficient computation of large powers modulo p.		

WEEK 3

Lab Sheet 5: Combinatorics

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

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

Lab Sheet 6: Graph Connectivity

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

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

WEEK 4

Lab Sheet 7: Array Processing

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

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

Lab Sheet 8: String Matching

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

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

WEEK 5

Lab Sheet 9: Priority Queue

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

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

Lab Sheet 10: Shortest Path

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

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

WEEK 6

Lab Sheet 11: Tree Traversal

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

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

Lab Sheet 12: Tree Applications

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

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

WEEK 7

Lab Sheet 13: Disjoint Set

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

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

Lab Sheet 14: Greedy Algorithms

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

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

WEEK 8

Lab Sheet 15: BFS Applications

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

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

Lab Sheet 16: Game Algorithms

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

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

WEEK 9

Lab Sheet 17: Sorting Algorithms

Level 1: Write a program to implement insertion sort.

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

Lab Sheet 18: Searching Algorithms

Level 1: Write a program to perform linear search.

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

WEEK 10

Lab Sheet 19: Algorithm Analysis

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

Level 2: Optimize a given algorithm to improve performance.

Lab Sheet 20: Recursion

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

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

WEEK 11

Lab Sheet 21: Graph Representation

Level 1: Represent a graph using adjacency list.

Level 2: Represent a graph using adjacency matrix.

Lab Sheet 22: Graph Traversal

Level 1: Implement Depth First Search (DFS).

Level 2: Implement Breadth First Search (BFS).

WEEK 12

Lab Sheet 23: Data Structure Integration

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

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

Lab Sheet 24: Real-world Application

Level 1: Design a simple recommendation system.

Level 2: Optimize it for large-scale data.

WEEK 13

Lab Sheet 25: String Algorithms

Level 1: Write a program for basic string matching.

Level 2: Implement efficient pattern matching algorithm.

Lab Sheet 26: Network Algorithms

Level 1: Check connectivity in a network.

Level 2: Detect anomalies using data comparison techniques.

WEEK 14

Lab Sheet 27: Algorithm Design

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

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

Lab Sheet 28: Performance Evaluation

Level 1: Analyze runtime of a program.

Level 2: Compare different algorithmic approaches.

WEEK 15: Intelligent Optimization and Competitive Programming

Lab Sheet 29: Advanced Problem Solving

Level 1: Solve optimization problems using greedy or dynamic programming techniques.
Level 2: Implement backtracking or graph-based solutions for competitive programming problems.

Lab Sheet 30: Intelligent Algorithm Analysis

Level 1: Analyze time and space complexity of advanced algorithms.
Level 2: Compare heuristic, dynamic programming, and brute-force approaches for performance optimization.

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>

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

Assessment Type

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

Course Code: APT4005		Course Title: Aptitude For Employability Type of Course: Practical Only			L- T-P- C	0-0-2-1
Version No.			1.0			
Course Pre-requisites			Students should have the basic concepts of Quantitative aptitude, Verbal ability along with its applications in real life problems.			
Anti-requisites			Nil			
Course Description			This course is designed to enable the students to enhance their skills in quantitative aptitude and verbal ability skills.			
Course Objective			The objective of the course is to familiarize the learners with concepts in Quantitative Aptitude and Verbal ability through problem solving techniques suitable for their career development.			
Course Outcomes			On successful completion of the course the students shall be able to: CO1] Recall all the basic mathematical concepts CO2] Identify the principle concept needed in a question CO3] Solve the quantitative and logical ability questions with the appropriate concept.			
Course Content:						
Module 1		Quantitative Ability	Lab-10hrs		Platform Assessment-10hrs	20 Hours
	Topics: Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss, Time Speed and Distance, Simple Interest and Compound Interest, Probability, Permutation and Combination.					
Module 2		Verbal Ability	Lab-5hrs		Platform Assessment-5hrs	10 Hours
	Topics: - Parts of Speech, Subject Verb Agreement, Spotting Error, Cloze Test, Verbal Analogies, Reading Comprehension, Idioms & Phrases, Para Jumbles					

	Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS
Evaluation	Continuous Evaluation • Topic wise evaluation
	1. Text Book Fast track objective by Rajesh Verma 2. R S Aggarwal 3. S.P Bakshi
	References 1. www.indiabix.com 2. www.testbook.com 3. www.youtube.com/c/TheAptitudeGuy/videos
	Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.

Course Code: CSS7100	Course Title: Mini Project Type of Course: PRW	L- T-P- C	0-0-0-4
Version No.	1.0		
Course Pre- requisites	Knowledge and Skills related to all the courses studied in previous semesters.		
Anti-requisites	NIL		
Course Description	Students observe science and technology in action, develop an awareness of the methods of scientific experimentation, and often get opportunities to see, study, and operate sophisticated and costly equipment. They also get opportunities to work with advanced software tools and modern technologies used in industries and research environments. Students use tools and technologies from emerging fields such as Artificial Intelligence (AI), Machine Learning (ML), Cyber Security, Blockchain, and Data Science to develop solutions for real-world and industry-related problems. They also learn about the implementation of management principles taught in the classroom by observing multidisciplinary teams of experts from engineering, science, economics, operations research, and management dealing with techno-economic problems at both micro and macro levels. The course further enables students to develop and refine their language, communication, teamwork, and interpersonal skills through various activities such as seminars, group discussions, project presentations, and report preparation. The broad-based core education, supported by strong foundations in mathematics, science, and analytical tools, provides the necessary background for students to properly understand and solve practical real-life problems. Students have the		

	flexibility to pursue this course either as Project Work and Dissertation within the university, as Project Work in an Industry, Company, or Research Laboratory, or through Internship Programs in industries and organizations.
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, students shall be able to: CO1: Identify engineering problems related to local, regional, national, and global needs. [Understand] CO2: Apply appropriate techniques and modern tools to solve the intended problems. [Apply] CO3: Design experiments in accordance with standards and specifications. [Analyze] CO4: Interpret events and results to derive meaningful conclusions. [Evaluate] CO5: Appraise project findings and communicate effectively through scholarly publications. [Create]

Course Code: PPS3018	Course Title: Preparedness for Interview Type of Course: Practical Only Course	L- T-P- C	0-0-2-1
Version No.		1.0	
Course Pre-requisites		Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.	
Anti-requisites		NIL	
Course Description		This course is designed to enable students to understand soft skills concepts to be corporate ready. The modules are set to improve self-confidence, communicate effectively and Prepare for the Interview to assist in employability. It helps the students to get a glimpse of the acceptable corporate readiness and equip them with the fundamental necessities of being able to confidently deal with the highly competitive corporate environment and helps in crafting different types of resumes. The pedagogy used will be group discussions, flipped classrooms, continuous feedback, role-play and mentoring.	

Course Objective		The objective of the course is to familiarize the learners with the concepts of “Preparing for Interview” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.			
Course Out Comes		On successful completion of this course the students shall be able to: CO1: Develop professional Resumes CO2: Illustrate Resumes effectively CO3: Apply skills and knowledge learnt for active and effective Group Discussions and Interview			
Course Content:					
Module 1	Resume Building	Classroom activity			10 Hours
	Topics: Resume structure, use of templates, Do's and Don'ts, ATS methods, Cover Letter and Video Resume Activity: Real world scenarios				

Course Code: CCS7300	Course Title: Capstone Project Type of Course: PRW	L- T-P- C	0-0-0-10
Version No.	1.0		
Course Pre- requisites	Knowledge and Skills related to all the courses studied in previous semesters.		
Anti-requisites	NIL		
Course Description	Students learn by observing science and technology in real-life applications and gain knowledge through practical experimentation and hands-on activities. They get opportunities to study and work with advanced equipment, software tools, and modern technologies. Students use tools and technologies from emerging fields such as Artificial Intelligence (AI), Machine Learning (ML), Cyber Security, Blockchain, and Data Science to develop solutions for real-world and industry-related problems. The course helps students understand how engineering, science, economics, operations research, and management work together to solve technical and business-related challenges. By working on projects, seminars, group discussions, and report preparation, students improve their communication, teamwork, presentation, and interpersonal skills. The strong foundation in mathematics, science, and analytical thinking helps students understand and solve practical problems effectively. Students can complete the course		

	through Project Work and Dissertation within the university, by doing projects in industries or research laboratories, or through internship programs in companies and organizations.
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, students shall be able to: CO1: Identify engineering problems at various societal levels (Understand) CO2: Apply tools and techniques for problem solving. (Apply) CO3: Design and plan experiments per industry standards. (Apply) CO4: Interpret the events and results for meaningful conclusions. (Evaluate) CO5: Appraise project findings and communicate effectively through scholarly publications. (Create)

ELECTIVES

Course Code: CDV3403	Course Title: Infrastructure as Code (IaC) with Terraform & Ansible Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
NCrF Level-	NIL		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	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 Objective	- 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 Out Comes	On successful completion of the course the students shall be able to: CO1: Describe the purpose, workflow, and tools of Infrastructure as Code (Understand) CO2: Compare declarative and imperative approaches for managing infrastructure (Analyze) CO3: Automate cloud provisioning using Terraform modules and workspaces (Apply) CO4: Use Ansible for system configuration, orchestration, and remote execution (Apply)			
Course Content:				
Module 1	Introduction to IaC and Cloud Infrastructure	Assignment	Conceptual Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Practical Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Tool-Based Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Case Study / Mini Project	12 Sessions
Topics: 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				
Targeted Application & Tools that can be used: Terraform, Ansible, AWS, Azure, Google Cloud Platform, HashiCorp Vault, AWS Secrets Manager, InSpec, Molecule, Docker, Kubernetes, Git, and CI/CD tools like Jenkins and GitHub Actions. Professionally used software: Terraform, Ansible, AWS CloudFormation, HashiCorp Vault, Jenkins, GitHub Actions, Docker, and Kubernetes.				
Skill-Building Activities through Experiential Learning: Develop practical skills in automating infrastructure provisioning and configuration using Terraform and Ansible, enabling scalable, consistent, and reliable deployment of cloud-based systems.				
Text Book T1: Yevgeniy Brikman, <i>Terraform: Up & Running</i> , O'Reilly Media, 2023 T2: Lorin Hochstein, <i>Ansible: Up and Running</i> , O'Reilly Media, 2023				
References R1: Russell Jones, <i>Learning Infrastructure as Code</i> , Packt Publishing, 2022 R2: Steve Smith, <i>Infrastructure as Code Handbook</i> , Leanpub, 2023 R3: James Turnbull, <i>The Terraform Book</i> , James Turnbull Publications, 2022 R4: Gourav Shah, <i>Mastering Ansible</i> , Packt Publishing, 2022				
E-Resources/Web Resources 1: https://www.terraform.io/docs 2: https://docs.ansible.com 3: https://learn.hashicorp.com 4: https://www.redhat.com/en/technologies/management/ansible 5: https://github.com/ansible/ansible-examples				

Course Code: CDV3400	Course Title: Fundamentals of Cloud Computing Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		

NCrF Level-	NIL			
Course Pre-requisites	CDV2000 – DevOps Foundations			
Co-requisites	NIL			
Anti-requisites	NIL			
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 Objective	• 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 Out Comes	On successful completion of the course the students shall be able to: C01: Explain cloud computing principles, architecture, and service models (Understand) C02: Compare deployment models and evaluate cost, scalability, and reliability (Analyze) C03: Use virtualization and containerization tools for cloud deployment (Apply) C04: Work with cloud platforms to provision, monitor, and scale services (Apply)			
Course Content:				
Module 1	Introduction to Cloud Computing	Assignment	Conceptual Analysis	11 Sessions
<u>Topics:</u> 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	Assignment	Test Case Design Exercise	11 Sessions
<u>Topics:</u> 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	Assignment	Automation Testing Assignment	11 Sessions

Topics:

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	Assignment	Performance Testing Case Study	12 Sessions
----------	-------------------------------	------------	--------------------------------	-------------

Topics:

Hands-on with AWS Console, Azure Portal, Google Cloud Console, Cloud-native application development, Real-world use cases in e-commerce, banking, healthcare, education

Targeted Application & Tools that can be used:

AWS (Amazon Web Services)	Cloud computing platform used for scalable infrastructure, storage, networking, and application deployment.
Microsoft Azure	Cloud platform used for hosting, managing, and deploying enterprise applications and services.
Google Cloud Platform (GCP)	Cloud computing platform used for infrastructure management, analytics, and scalable cloud deployments.
Docker	Containerization platform used for packaging, deploying, and running applications consistently across environments.
Kubernetes	Container orchestration platform used for automating deployment, scaling, and management of containerized applications.
VirtualBox / VMware	Virtualization platforms used for creating and managing virtual machines and testing environments.
Terraform	Infrastructure as Code (IaC) tool used for provisioning and managing cloud infrastructure resources.
CloudWatch / Azure Monitor	Cloud-native monitoring tools used for tracking infrastructure performance, logging, alerting, and operational analytics.

Skill-Building Activities through Experiential Learning:

Develop the ability to design, deploy, and manage scalable cloud-based applications using virtualization, containerization, and modern cloud platforms.

Text Book

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

References

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: CDV3407	Course Title: AI-Driven DevOps & IT Operations Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
NCrF Level-	NIL		
Course Pre-requisites	CDV1700 DevOps Foundations		
Co-requisites	NIL		
Anti-requisites	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 Objective	• 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 Out Comes	On successful completion of the course the students shall be able to: CO1: Explain the fundamentals, components, and benefits of AIOps (Understand) CO2: Correlate events and alerts using ML models and behavior analysis (Analyze) CO3: Build anomaly detection and automation pipelines using AIOps tools (Apply) CO4: Integrate AIOps with DevOps workflows for intelligent system management (Apply)		
Course Content:			

Module 1	Introduction to AIOps	Assignment	Conceptual Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Analytical Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Practical Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Case Study / Mini Project	12 Sessions
<u>Topics:</u> 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				
Targeted Application & Tools that can be used:				
Dynatrace, Moogsoft, Splunk ITSI, Elastic Stack (ELK), Datadog, Prometheus, Grafana, Apache Kafka, Fluentd, OpenTelemetry, Python (scikit-learn, pandas), and cloud platforms like AWS, Azure, and Google Cloud.		Dynatrace, Moogsoft, Splunk ITSI, Elastic Stack (ELK), Datadog, Prometheus, Grafana, Apache Kafka, Fluentd, OpenTelemetry, Python (scikit-learn, pandas), and cloud platforms like AWS, Azure, and Google Cloud.		
Professionally used software : Dynatrace, Splunk ITSI, Datadog, Elastic Stack, Moogsoft, Prometheus, Grafana, and cloud-native AIOps solutions from AWS and Azure		Professionally used software : Dynatrace, Splunk ITSI, Datadog, Elastic Stack, Moogsoft, Prometheus, Grafana, and cloud-native AIOps solutions from AWS and Azure		
Dynatrace, Moogsoft, Splunk ITSI, Elastic Stack (ELK), Datadog, Prometheus, Grafana, Apache Kafka, Fluentd, OpenTelemetry, Python (scikit-learn, pandas), and cloud platforms like AWS, Azure, and		Dynatrace, Moogsoft, Splunk ITSI, Elastic Stack (ELK), Datadog, Prometheus, Grafana, Apache Kafka, Fluentd, OpenTelemetry, Python (scikit-learn, pandas), and cloud platforms like AWS, Azure, and Google Cloud.		

Google Cloud.	
Professionally used software : Dynatrace, Splunk ITSI, Datadog, Elastic Stack, Moogsoft, Prometheus, Grafana, and cloud-native AIOps solutions from AWS and Azure	Professionally used software : Dynatrace, Splunk ITSI, Datadog, Elastic Stack, Moogsoft, Prometheus, Grafana, and cloud-native AIOps solutions from AWS and Azure
Dynatrace, Moogsoft, Splunk ITSI, Elastic Stack (ELK), Datadog, Prometheus, Grafana, Apache Kafka, Fluentd, OpenTelemetry, Python (scikit-learn, pandas), and cloud platforms like AWS, Azure, and Google Cloud.	Dynatrace, Moogsoft, Splunk ITSI, Elastic Stack (ELK), Datadog, Prometheus, Grafana, Apache Kafka, Fluentd, OpenTelemetry, Python (scikit-learn, pandas), and cloud platforms like AWS, Azure, and Google Cloud.
Skill-Building Activities through Experiential Learning:	
Develop the ability to design test cases, perform manual and automated testing, identify software defects, and implement testing frameworks to ensure software reliability and quality in real-world development environments.	
Text Book T1: Pranay Ahlawat, <i>AIOps: Real-time Analytics for IT Operations</i> , Packt Publishing, 2023 T2: Nilesh Barla, <i>Mastering AIOps</i> , BPB Publications, 2023	
References R1: Mitch Ashley, <i>AIOps for Dummies</i> , Wiley, 2022 R2: Ernest Lefner, <i>Site Reliability Engineering and AIOps</i> , Leanpub, 2022 R3: Ankur Shah, <i>Enterprise AIOps with IBM Watson</i> , IBM Redbooks, 2021 R4: Jason Bloomberg, <i>Data-Driven DevOps</i> , 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: CDV3409	Course Title: AI-Enabled Cloud Cost Optimization Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
NCrF Level-	NIL		
Course Pre-requisites	NIL		

Co-requisites	NIL			
Anti-requisites	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 Objective	- 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 Out Comes	On successful completion of the course the students shall be able to: CO1: Explain cloud cost structures and factors affecting pricing (Understand) CO2 Examine usage data to identify inefficiencies and cost trends (Analyze) CO3: Use AI/ML models to forecast usage and automate optimization (Apply) CO4 (Apply): Implement tools and strategies for continuous cost governance(Apply)			
Course Content:				
Module 1	Cloud Billing & Cost Management Basics	Assignment	Conceptual Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Practical Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Practical Assignment	11 Sessions
<u>Topics:</u>				

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	Assignment	Case Study / Mini Project	12 Sessions
----------	---------------------------	------------	---------------------------	-------------

Topics:

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

Targeted Application & Tools that can be used:

Targeted Application & Tools that can be used: AWS Cost Explorer, AWS Compute Optimizer, Azure Cost Management, Google Cloud Billing, Grafana, Prometheus, Python (Pandas, scikit-learn, Prophet), CloudHealth, Spot.io, Terraform, and cloud-native monitoring tools.	Targeted Application & Tools that can be used: AWS Cost Explorer, AWS Compute Optimizer, Azure Cost Management, Google Cloud Billing, Grafana, Prometheus, Python (Pandas, scikit-learn, Prophet), CloudHealth, Spot.io, Terraform, and cloud-native monitoring tools.
Professionally used software : AWS Cost Explorer, Azure Cost Management, Google Cloud Billing, CloudHealth, Spot.io, Grafana, Prometheus, and machine learning tools like Python (scikit-learn, Prophet)	Professionally used software : AWS Cost Explorer, Azure Cost Management, Google Cloud Billing, CloudHealth, Spot.io, Grafana, Prometheus, and machine learning tools like Python (scikit-learn, Prophet)
Targeted Application & Tools that can be used: AWS Cost Explorer, AWS Compute Optimizer, Azure Cost Management, Google Cloud Billing, Grafana, Prometheus, Python (Pandas, scikit-learn, Prophet), CloudHealth, Spot.io, Terraform, and cloud-native monitoring tools.	Targeted Application & Tools that can be used: AWS Cost Explorer, AWS Compute Optimizer, Azure Cost Management, Google Cloud Billing, Grafana, Prometheus, Python (Pandas, scikit-learn, Prophet), CloudHealth, Spot.io, Terraform, and cloud-native monitoring tools.
Professionally used software : AWS Cost Explorer, Azure Cost Management, Google Cloud Billing, CloudHealth, Spot.io, Grafana, Prometheus, and machine learning tools like Python (scikit-learn, Prophet)	Professionally used software : AWS Cost Explorer, Azure Cost Management, Google Cloud Billing, CloudHealth, Spot.io, Grafana, Prometheus, and machine learning tools like Python (scikit-learn, Prophet)
Targeted Application & Tools that can be used: AWS Cost Explorer, AWS Compute Optimizer, Azure Cost Management, Google Cloud Billing, Grafana, Prometheus, Python (Pandas, scikit-learn, Prophet), CloudHealth, Spot.io, Terraform, and cloud-native monitoring tools.	Targeted Application & Tools that can be used: AWS Cost Explorer, AWS Compute Optimizer, Azure Cost Management, Google Cloud Billing, Grafana, Prometheus, Python (Pandas, scikit-learn, Prophet), CloudHealth, Spot.io, Terraform, and cloud-native monitoring tools.

Skill-Building Activities through Experiential Learning:	
Develop the ability to design test cases, perform manual and automated testing, identify software defects, and implement testing frameworks to ensure software reliability and quality in real-world development environments.	
Text Book T1: J.R. Storment & Mike Fuller, <i>Cloud FinOps: Collaborative Cloud Financial Management</i> , O'Reilly Media, 2023 T2: Aditya Garg, <i>Cloud Cost Optimization Handbook</i> , Packt Publishing, 2023	
References R1: Ajay Dholakia, <i>Cloud Optimization for Dummies</i> , Wiley, 2022 R2: Paul Zikopoulos, <i>AI for Cloud Operations</i> , IBM Redbooks, 2022 R3: Rajdeep Dua, <i>Practical Google Cloud Platform</i> , Apress, 2022 R4: Stephen Fleming, <i>The Art of Cost Optimization in AWS</i> , 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: CDV3411	Course Title: Reinforcement Learning for Self-Healing DevOps Pipelines Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
NCrF Level-	NIL		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	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 Objective	- 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 Out Comes	On successful completion of the course the students shall be able to: CO1: Explain the principles of Reinforcement Learning and its role in automation (Understand) CO2: Examine RL models for adaptive testing and pipeline optimization (Analyze) CO3: Implement RL agents for identifying and healing failures in CI/CD workflows (Apply) CO4: Integrate RL with DevOps tools for real-time decision making and system recovery (Apply)			
Course Content:				
Module 1	Reinforcement Learning Basics	Assignment	Conceptual Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Analytical Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Practical Assignment	11 Sessions
<u>Topics:</u> 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	Assignment	Case Study /Mini Project	12 Sessions
<u>Topics:</u> 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				

Targeted Application & Tools that can be used: Python (NumPy, Pandas), TensorFlow, PyTorch, OpenAI Gym, Stable-Baselines3, Jenkins, GitHub Actions, Kubernetes, Prometheus, Grafana, and log monitoring tools. Professionally used software: TensorFlow, PyTorch, OpenAI Gym, Stable-Baselines3, Jenkins, GitHub Actions, Kubernetes, Prometheus, and Grafana
Skill-Building Activities through Experiential Learning: Develop practical skills in designing reinforcement learning agents and integrating them with DevOps pipelines to enable self-healing, adaptive automation, and intelligent system optimization.
Text Book T1: Richard S. Sutton & Andrew G. Barto, <i>Reinforcement Learning: An Introduction</i>, MIT Press, 2nd Edition, 2023 T2: Rahul Raj, <i>Reinforcement Learning for DevOps Automation</i>, Packt Publishing, 2023
References R1: Maxim Lapan, <i>Deep Reinforcement Learning Hands-On</i>, Packt Publishing, 2022 R2: Daniel Whitenack, <i>Applied Machine Learning for DevOps Engineers</i>, O'Reilly Media, 2022 R3: Ian Goodfellow et al., <i>Deep Learning</i>, MIT Press, 2022 R4: Mark Smiley, <i>CI/CD for Machine Learning with Kubernetes</i>, O'Reilly, 2023
E-Resources/Web Resources https://spinningup.openai.com https://keras.io/examples/rl/ https://www.tensorflow.org/agents https://learn.microsoft.com/en-us/devops https://docs.github.com/en/actions

Course Code: CDV3416	Course Title: DevSecOps in Financial Services Type of Course: Theory	L-T- P- C	3-0-0-3
Version No.	2.0		
NCrF Level-	NIL		
Course Pre-requisites	NIL		
Co-requisites	NIL		

Anti-requisites	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 Objective	- 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 Out Comes	On successful completion of the course the students shall be able to: CO1: Describe the DevSecOps lifecycle and security needs of financial applications (Understand) CO2: Evaluate security threats and compliance risks in DevOps pipelines (Analyze) CO3: Implement automated security testing and compliance checks (Apply) CO4: Integrate policy-as-code and governance in financial DevOps systems (Apply)			
Course Content:				
Module 1	DevSecOps Foundations in Finance	Assignment	Conceptual Analysis	11 Sessions
<u>Topics:</u> 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	Assignment	Test Case Design Exercise	11 Sessions
<u>Topics:</u> 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	Assignment	Automation Testing Assignment	11 Sessions
<u>Topics:</u>				

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	Assignment	Performance Testing Case Study	12 Sessions
----------	------------------------------	------------	--------------------------------	-------------

Topics:

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)

Targeted Application & Tools that can be used:

SAST / DAST Tools (SonarQube, OWASP ZAP, Snyk)	Used for static and dynamic application security testing to identify vulnerabilities and security risks in software applications.
Container Security Tools (Trivy, Aqua Security)	Used for scanning container images, detecting vulnerabilities, and securing containerized environments.
Secrets Management Tools (HashiCorp Vault, GitGuardian)	Used for securely storing, managing, and monitoring sensitive credentials, API keys, and secrets.
Infrastructure Security Tools (Checkov, tfsec)	Used for analyzing Infrastructure as Code (IaC) configurations and detecting cloud security misconfigurations.
Policy-as-Code Tools (OPA – Open Policy Agent, Sentinel)	Used for defining, enforcing, and automating security and compliance policies in infrastructure and CI/CD workflows.
CI/CD Tools (Jenkins, GitHub Actions, GitLab CI)	Used for automating software build, testing, integration, deployment, and DevSecOps workflows.
SIEM Tools (Splunk, ELK Stack)	Used for centralized logging, security monitoring, threat detection, and security event analysis.

Skill-Building Activities through Experiential Learning:

Develop the ability to integrate security, compliance, and automation into DevOps pipelines to build secure, resilient, and audit-ready financial applications.

Text Book

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

References

- 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>