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.) in COMPUTER SCIENCE AND ENGINEERING BLOCK CHAIN (CBC)

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. *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 B.Tech. Computer Science and Engineering

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

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

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

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

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

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

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

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

10. B. Tech. Computer Science and Information Technology

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

12. B. Tech. Computer Engineering

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

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

15. B.Tech. - Artificial Intelligence and Data Science

16. B.Tech. - Robotics and Artificial Intelligence

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. 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 Mechanical Engineer with innovative skills and moral and ethical values.

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

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

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

8.1 Programme Outcomes (PO)

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

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

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

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

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

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

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

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

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

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

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

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

8.2 Program Specific Outcomes (PSOs):

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

PSO 01: Showcase a deep understanding of computer science fundamentals to explain key concepts, analyze computational problems using theoretical knowledge, and design reliable and optimized computing systems..

PSO 02: Utilize programming skills, software development experience, and knowledge in blockchain technology to build practical, real-world applications, that help them prepare for a myriad of career opportunities in software development, pursue advanced studies, research or embark on entrepreneurial ventures.

PSO 03: Understand and utilize core principles of blockchain architecture, such as distributed ledgers, consensus mechanisms, smart contract development, while analyzing the cryptographic foundations, trust models, as well as decentralized control mechanisms to design blockchain-based solutions that ensure data integrity and resilience across various sectors.

9 Admission Criteria (as per the concerned Statutory Body)

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

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

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

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

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

9.5 Admissions are offered to Foreign Nationals and Indians living abroad in accordance with the

rules applicable for such admission, issued from time to time, by the Government of India.

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

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

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

10 Lateral Entry / Transfer Students requirements

10.1 Lateral Entry

The University admits students directly to the second year (3rd Semester) of the B.Tech. Degree program as per the provisions and/or regulations of the Government of Karnataka pertaining to the "Lateral Entry" scheme announced by the Government from time to time. Further, the general conditions and rules governing the provision of Lateral Entry to the B.Tech./B.E./B.S 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 Engineering-Block Chain) 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 Engineering-Block Chain) 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. 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 ssessment Components and Weightage

2	Credit Structure [L-T-P-C]	Percentage/ Marks	CA		Mid-Term		End-term		Project	Total	Exam Conducted by
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	

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

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

2.1 Minimum Performance Criteria:

2.1.1 Theory only Course and Lab/Practice Embedded Theory Course

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

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

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

2.1.2 Lab/Practice only Course and Project Based Courses

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

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

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

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

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

3.2 Students may earn credits from other Indian or foreign Universities/Institutions with which the University has an MOU, and that MOU shall have specific provisions, rules and guidelines

for transfer of credits. These transferred credits shall be counted towards the minimum credit requirements for the award of the degree.

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

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

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

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

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

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

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

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

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

Table 2: Durations and Credit Equivalence for Transfer of Credits from SWAYAM-NPTEL/ other approved MOOC Courses		
Sl. No.	**Course Duration**	**Credit Equivalence**
1	4 Weeks	1 Credit
2	8 Weeks	2 Credits
3	12 Weeks	3 Credits

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

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

3.4 The maximum number of credits that can be transferred by a student shall be limited to forty percent (40%) of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum for the award of the concerned Degree. However, the grades obtained in the Courses transferred from other Institutions/MOOCs, as mentioned in this Section (13.0) 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 Engineering-Block Chain) Program Structure (2025-2029) totalling 160 credits. Table 3.0 summarizes the type of baskets, number of courses under each basket and the associated credits that are mandatorily required for the completion of the Degree.

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

Table 3: B.Tech. (Computer Science and Engineering Block Chain) 2025-2029: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets		
Sl. No.	**Baskets**	**Credit Contribution**
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 57% out of the total credits of 160 for B.Tech. (Computer Science and Engineering Block Chain) program of four years' duration.

15. Minimum Total Credit Requirements of Award of Degree

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

PART C – CURRICULUM STRUCTURE

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

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

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

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

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

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

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

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

Table 3.1: List of Humanities and Social Sciences including Management Courses (HSMC)									
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of skill**	**Pre-Requisite**
1	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	FIN1002	Essentials of Finance	3	0	0	3	3	S	
6	APT4005	Aptitude for Employability	0	0	2	1	2	S	
		Total No. of Credits				**12**			

Table 3.2: List of Basic Science Courses (BSC)									
Sl.No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	**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**	4	0	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)									
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	MEC1006	Engineering Graphics	2	0	0	2	2	S	-
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)									
Sl.No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Contact Hours**	**Type of Skill**	**Pre-requisite**
1	CSS2200	Problem Solving using C	2	0	0	2	2	S	-
2	CSS2201	Problem Solving using C Lab	0	0	4	2	4	S	-
3	CSS2251	Data Communication and Computer Networks	3	0	0	3	3	S	-
4	CSS2252	Data Communication and Computer Networks Lab	0	0	2	1	2	S	-
5	CSS2253	Data Structures	3	0	0	3	3	S	-
6	CSS2254	Data Structures Lab	0	0	2	1	2	S	-
7	CSS2255	Object Oriented Programming Using Java	3	0	0	3	3	S/EM	-
8	CSS2256	Object Oriented Programming Using Java Lab	0	0	2	1	2	S/EM	-
9	CSS2257	Computer Organization and Architecture	3	0	0	3	3	S	-
10	CBC2000	Foundations of Blockchain Technology	3	0	0	3	3	S/EM	-

11	CSS2271	Software Design and Development	3	0	0	3	3	S	-
12	CSS2258	Web Technologies	3	0	0	3	3	S/EM	-
13	CSS2259	Web Technologies Lab	0	0	2	1	2	S/EM	-
14	CSS2260	Database Management Systems	3	0	0	3	3	S	-
15	CSS2261	Database Management Systems Lab	0	0	2	1	2	S	-
16	CSS2262	Analysis of Algorithms	3	1	0	4	4	S	-
17	CSS2263	Analysis of Algorithms Lab	0	0	2	1	2	S	-
18	CSS2266	Theory of Computation	3	0	0	3	3	S	-
19	CSS2502	Cryptography and Network Security	3	0	0	3	3	S	CSE2051
20	CSS2269	Operating Systems	3	0	0	3	3	S	Nil
21	CSS2270	Operating Systems Lab	0	0	2	1	2	S/EM	Nil
22	CBC2500	Smart Contract and Solidity	3	0	0	3	3	S/EM	CBC2000
23	CBC2501	Smart Contract and Solidity Lab	0	0	2	1	2	S/EM	CBC2000
24	CBC2502	Distributed Ledger Technology	3	0	0	3	3	S/EM	CBC2000
25	CBC2503	Distributed Ledger Technology Lab	0	0	2	1	2	S/EM	CBC2000
26	CBC2504	Blockchain Security and Performance	3	0	0	3	3	S/EM	CBC2000
27	CBC2505	Blockchain Security and Performance Lab	0	0	4	2	4	S/EM	CBC2000
28	CBC2506	Blockchain Architecture Design	3	0	0	3	3	S/EM	CBC2000
		Total No. of Credits				**64**			

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

Sl. No	**Course code**	**Course Name**	**L**	**T**	**P**	**C**
1	CSS7300	Capstone Project	0	0	0	10
2	CSS7000	Internship	0	0	0	2
3	CSS7100	Mini Project	0	0	0	4
		Total No. of Credits				**16**

Table 3.6: List of Mandatory Courses (MAC)						
Sl. 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) 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 project work in an Industry / Company or academic / research institution of her / his choice subject to the above mentioned condition (Sub-Clause 18.2.1). Provided further, that the Industry / Company or academic / research institution offering such project work confirms to the University that the project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.3 Capstone Project

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

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

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

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

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

18.3.1.5 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

Sl. No	New Course Code	Course Name	L	T	P	C	Type of Skill	Prerequisite	Type of course
1	CBC3400	Cryptography and Security in Blockchain	3	0	0	3	S/ EM	Nil	PEC
2	CSS3426	Front End Full Stack Development	2	0	2	3	S/ EM	CSE1504	PEC
3	CSS3427	Java Full Stack Development	2	0	2	3	S/ EM	CSE1514	PEC
	CSS3428	.Net Full Stack Development	2	0	2	3	S/ EM	CSE1504	PEC
4	CBC3401	Crypto Trading Strategies & Risk Management	3	0	0	3	S/EM	Nil	PEC
	CBC3402	Bitcoin and Ethereum Protocols	3	0	0	3	S/EM	Nil	PEC
	CAI3427	Language Models for Text Mining	2	0	2	3	S/ EM	CSE2064	PEC
5	CBC3403	Blockchain for Digital Identity Management	3	0	0	3	S	Nil	PEC
	CBC3404	Cryptocurrency Wallet Development	3	0	0	3	S	Nil	PEC
	CAI3428	Practical Deep Learning with TensorFlow	2	0	2	3	S/ EM	CSE2064	PEC
6	CBC3405	Blockchain Security & Ethical Hacking	3	0	0	3	S/EM	Nil	PEC
	CAI3429	Deep Learning for Computer Vision	2	0	2	3	S/ EM	**MAT1212**	PEC
Sl. No	**New Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Type of Skill**	**Prerequisite**	**Type of course**
1	CBC3406	Introduction to Artificial Intelligence in Block Chain	3	0	0	3	S/ EM	Nil	PEC
2	CSS3426	Front End Full Stack Development	2	0	2	3	S/ EM	CSE1504	PEC
3	CSS3427	Java Full Stack Development	2	0	2	3	S/ EM	CSE1514	PEC
	CSS3428	.Net Full Stack Development	2	0	2	3	S/ EM	CSE1504	PEC

4	CBC3407	Machine Learning for Cyber Threat Detection	3	0	0	3	S/EM	Nil	PEC
	CBC3408	AI-Powered Fraud Detection in Blockchain	3	0	0	3	S/EM	Nil	PEC
	CAI3427	Language Models for Text Mining	2	0	2	3	S/ EM	CSE2064	PEC
5	CBC3409	Optimizing Blockchain Networks with AI	3	0	0	3	S	Nil	PEC
	CBC3410	Generative AI for Blockchain Applications	3	0	0	3	S	Nil	PEC
	CAI3428	Practical Deep Learning with TensorFlow	2	0	2	3	S/ EM	CSE2064	PEC
6	CBC3411	Quantum Computing & Blockchain-AI Security	3	0	0	3	S/EM	Nil	PEC
	CAI3429	Deep Learning for Computer Vision	2	0	2	3	S/ EM	**MAT1212**	PEC
Sl.No	**New Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Type of Skill**	**Prerequisite**	**Type of course**
1	CBC3412	Introduction to Decentralized Finance (DeFi)	3	0	0	3	S/ EM	Nil	PEC
2	CSS3426	Front End Full Stack Development	2	0	2	3	S/ EM	CSE1504	PEC
3	CSS3427	Java Full Stack Development	2	0	2	3	S/ EM	CSE1514	PEC
	CSS3428	.Net Full Stack Development	2	0	2	3	S/ EM	CSE1504	PEC
4	CBC3413	Blockchain in Financial Services	3	0	0	3	S/EM	Nil	PEC
	CBC3414	Building Decentralized Applications (DApps) for Finance	3	0	0	3	S/EM	Nil	PEC
	CAI3427	Language Models for Text Mining	2	0	2	3	S/ EM	CSE2064	PEC
5	CBC3415	Smart Contracts for Financial Products	3	0	0	3	S	Nil	PEC
	CBC3416	Decentralized Autonomous Organizations and	3	0	0	3	S	Nil	PEC

		Risk Management in Finance							
	CAI3428	Practical Deep Learning with TensorFlow	2	0	2	3	S/ EM	CSE2064	PEC
6	CBC3417	Regulatory and Compliance Challenges in DeFi	3	0	0	3	S/EM	Nil	PEC
	CAI3429	Deep Learning for Computer Vision	2	0	2	3	S/ EM	**MAT1212**	PEC
Sl. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**	**Type of Skill**	**Prerequisite**	
1	LAW1007	Indian Constitution and Professional Ethics for Engineers	1	0	0	0	F	-	
2	CHE7601	Environmental Studies **	1	0	0	0	F	-	
3	XXXXXXX	Universal Human Values **	0	0	0	0	F	-	
Paytm Curriculum Structure									
1	XXXXXXX	Distributed Ledger Architectures and Smart Contract Systems	2	0	2	3			
2	XXXXXXX	Scalable DApp Engineering and Web3 System Integration	2	0	2	3			
3	XXXXXXX	Decentralized Finance (DeFi) Protocols on Enterprise Blockchain Applications	2	0	2	3			
4	XXXXXXX1	Blockchain Data Engineering and Regulatory Intelligence	2	0	2	3			
5	XXXXXXX	Interoperability Engineering and Cross-Chain System Design - I	2	0	2	3			

6	XXXXXXX	Interoperability Engineering and Cross-Chain System Design - II	2	0	2	3			
7	XXXXXXX	Mini Project - Blockchain Sprint	0	0	0	4			
8	XXXXXXX	Capstone Project - Blockchain FinTech Platform	0	0	0	10			

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**	**Type of Skill/ Focus**	**Course Caters to**	**Prerequisites/ Corequisites**	**Antirequisites**	**Future Courses that need this as a Prerequisite**
Chemistry Basket											
1	CHE1003	Fundamentals of Sensors	3	0	0	3	S	ES	-	-	-
2	CHE1004	Smart materials for IOT	3	0	0	3	S	ES	-	-	-
3	CHE1005	Computational Chemistry	2	0	0	2	S	ES	-	-	-
4	CHE1006	Introduction to Nano technology	3	0	0	3	S	ES	-	-	-
5	CHE1007	Biodegradable electronics	2	0	0	2	S	ES	-	-	-
6	CHE1008	Energy and Sustainability	2	0	0	2	S	ES	-	-	-
7	CHE1009	3D printing with Polymers	2	0	0	2	S	ES	-	-	-
8	CHE1010	Bioinformatics and Healthcare IT	2	0	0	2	S	ES	-	-	-
9	CHE1011	Chemical and Petrochemical catalysts	3	0	0	3	S	ES	-	-	-
10	CHE1012	Introduction to Composite materials	2	0	0	2	S	ES	-	-	-
11	CHE1013	Chemistry for Engineers	3	0	0	3	S	ES	-	-	-
12	CHE1014	Surface and Coatings technology	3	0	0	3	S	ES	-	-	-
13	CHE1015	Waste to Fuels	2	0	0	2	S	ES	-	-	-
14	CHE1016	Forensic Science	3	0	0	3	S	ES	-	-	-
Civil Engineering Basket											
1	CIV1001	Disaster mitigation and management	3	0	0	3	S	-	-	-	-
2	CIV1002	Environment Science and Disaster Management	3	0	0	3	FC	-	-	-	-
3	CIV2001	Sustainability Concepts in Engineering	3	0	0	3	S	-	-	-	-
4	CIV2002	Occupational Health and Safety	3	0	0	3	S	-	-	-	-
5	CIV2003	Sustainable Materials and Green Buildings	3	0	0	3	EM	-	-	-	-
6	CIV2004	Integrated Project Management	3	0	0	3	EN	-	-	-	-
7	CIV2005	Environmental Impact Assessment	3	0	0	3	EN	-	-	-	-
8	CIV2006	Infrastructure Systems for Smart Cities	3	0	0	3	EN	-	-	-	-
9	CIV2044	Geospatial Applications for Engineers	2	0	2	3	EM	-	-	-	-
10	CIV2045	Environmental Meteorology	3	0	0	3	S	-	-	-	-
11	CIV3046	Project Problem Based Learning	3	0	0	3	S	-	-	-	-
12	CIV3059	Sustainability for Professional Practice	3	0	0	3	EN	-	-	-	-
Commerce Basket											
1	COM2001	Introduction to Human Resource Management	2	0	0	2	F	HP/GS	-	-	-
2	COM2002	Finance for Non Finance	2	0	0	2	S	-	-	-	-
3	COM2003	Contemporary Management	2	0	0	2	F	-	-	-	-
4	COM2004	Introduction to Banking	2	0	0	2	F	-	-	-	-
5	COM2005	Introduction to Insurance	2	0	0	2	F	-	-	-	-
6	COM2006	Fundamentals of Management	2	0	0	2	F	-	-	-	-
7	COM2007	Basics of Accounting	3	0	0	3	F	-	-	-	-
Computer Science Basket											
1	CSS2002	Programming in Java	2	0	2	3	S/EM	-	-	-	-
2	CSS2003	Social Network Analytics	3	0	0	3	S	GS	-	-	-
3	CSS2004	Python Application Programming	2	0	2	3	S/ EM	-	-	-	-
4	CSS2005	Web design fundamentals	2	0	2	3	S/ EM/EN	-	-	-	-
5	CSS3111	Artificial Intelligence : Search Methods For Problem Solving	3	0	0	3	S/ EM/EN	-	-	-	-

6	CSS3112	Privacy And Security In Online Social Media	3	0	0	3	S/ EM/EN	-	-	-	-
7	CSS3113	Computational Complexity	3	0	0	3	S/ EM/EN	-	-	-	-
8	CSS3114	Deep Learning for Computer Vision	3	0	0	3	S/ EM/EN	-	-	-	-
9	CSS3115	Learning Analytics Tools	3	0	0	3	S/ EM/EN	-	-	-	-
Design Basket											
1	DES1001	Sketching and Painting	0	0	2	1	S	-	-	-	-
2	DES1002	Innovation and Creativity	2	0	0	2	F	-	-	-	-
3	DES1121	Introduction to UX design	1	0	2	2	S	-	-	-	-
4	DES1122	Introduction to Jewellery Making	1	0	2	2	S	-	-	-	-
5	DES1124	Spatial Stories	1	0	2	2	S	-	-	-	-
6	DES1125	Polymer Clay	1	0	2	2	S	-	-	-	-
7	DES2001	Design Thinking	3	0	0	3	S	-	-	-	-
8	DES1003	Servicability of Fashion Products	1	0	2	2	F	ES	-	-	-
9	DES1004	Choices in Virtual Fashion	1	0	2	2	F	ES, GS, HP	-	-	-
10	DES1005	Fashion Lifestyle and Product Diversity	1	0	2	2	F	ES, GS, HP	-	-	-
11	DES1006	Colour in Everyday Life	1	0	2	2	F	ES	-	-	-
12	DES2080	Art of Design Language	3	0	0	3	S	-	-	-	-
13	DES2081	Brand Building in Design	3	0	0	3	S	-	-	-	-
14	DES2085	Web Design Techniques	3	0	0	3	S	-	-	-	-
15	DES2089	3D Modeling for Professionals	1	0	4	3	S	-	-	-	-
16	DES2090	Creative Thinking for Professionals	3	0	0	3	S	-	-	-	-
17	DES2091	Idea Formulation	3	0	0	3	S	-	-	-	-
Electrical and Electronics Basket											
1	EEE1002	IoT based Smart Building Technology	3	0	0	3	S	-	-	-	-
2	EEE1003	Basic Circuit Analysis	3	0	0	3	S	-	-	-	-
3	EEE1004	Fundamentals of Industrial Automation	3	0	0	3	S	-	-	-	-
4	EEE1005	Electric Vehicles & Battery Technology	3	0	0	3	S	-	-	-	-
5	EEE1006	Smart Sensors for Engineering Applications	3	0	0	3	S	-	-	-	-
Electronics and Communication Basket											
1	ECE1003	Fundamentals of Electronics	3	0	0	3	F	-	-	-	-
2	ECE1004	Microprocessor based systems	3	0	0	3	F	-	-	-	-
3	ECE3089	Artificial Neural Networks	3	0	0	3	S	-	-	-	-
4	ECE3097	Smart Electronics in Agriculture	3	0	0	3	F/EM	-	-	-	-
5	ECE3098	Environment Monitoring Systems	3	0	0	3	F/EM	-	-	-	-
6	ECE3102	Consumer Electronics	3	0	0	3	F/EM	-	-	-	-
7	ECE3103	Product Design of Electronic Equipment	3	0	0	3	S/F/ EM / EN	-	-	-	-
8	ECE3106	Introduction to Data Analytics	3	0	0	3	F/EM	-	-	-	-
9	ECE3107	Machine Vision for Robotics	3	0	0	3	F/EM	-	-	-	-
English Basket											
1	ENG1008	Indian Literature	2	0	0	2	-	GS/ HP	-	-	-
2	ENG1009	Reading Advertisement	3	0	0	3	S	-	-	-	-
3	ENG1010	Verbal Aptitude for Placement	2	0	2	3	S	-	-	-	-
4	ENG1011	English for Career Development	3	0	0	3	S	-	-	-	-
5	ENG1012	Gender and Society in India	2	0	0	2	-	GS/ HP	-	-	-
6	ENG1013	Indian English Drama	3	0	0	3	-	-	-	-	-
7	ENG1014	Logic and Art of Negotiation	2	0	2	3	-	-	-	-	-
8	ENG1015	Professional Communication Skills for Engineers	1	0	0	1	-	-	-	-	-
DSA Basket											

1	DSA2001	Spirituality for Health	2	0	0	2	F	HP	-	-	-
2	DSA2002	Yoga for Health	2	0	0	2	S	HP	-	-	-
3	DSA2003	Stress Management and Well Being	2	0	0	2	F	-	-	-	-
Kannada Basket											
1	KAN1001	Kali Kannada	1	0	0	1	S	-	-	-	-
2	KAN1003	Kannada Kaipidi	3	0	0	3	S	-	-	-	-
3	KAN2001	Thili Kannada	1	0	0	1	S	-	-	-	-
4	KAN2003	Pradharshana Kale	1	0	2	2	S	-	-	-	-
5	KAN2004	Sahithya Vimarshe	2	0	0	2	S	-	-	-	-
6	KAN2005	Anuvadha Kala Sahithya	3	0	0	3	S	-	-	-	-
7	KAN2006	Vichara Manthana	3	0	0	3	S	-	-	-	-
8	KAN2007	Katha Sahithya Sampada	3	0	0	3	S	-	-	-	-
9	KAN2008	Ranga Pradarshana Kala	3	0	0	3	S	-	-	-	-
Foreign Language Basket											
1	FRL1004	Introduction of French Language	2	0	0	2	S	S	-	-	-
2	FRL1005	Fundamentals of French	2	0	0	2	S	S	-	-	-
3	FRL1009	Mandarin Chinese for Beginners	3	0	0	3	S	S	-	-	-
Law Basket											
1	LAW1001	Introduction to Sociology	2	0	0	0	2	F	HP	-	-
2	LAW2001	Indian Heritage and Culture	2	0	0	0	2	F	HP/GS	-	-
3	LAW2002	Introduction to Law of Succession	2	0	0	0	2	F	HP/GS	-	-
4	LAW2003	Introduction to Company Law	2	0	0	0	2	F	HP	-	-
5	LAW2004	Introduction to Contracts	2	0	0	2	F	HP	-	-	-
6	LAW2005	Introduction to Copy Rights Law	2	0	0	2	F	HP	-	-	-
7	LAW2006	Introduction to Criminal Law	2	0	0	2	F	HP	-	-	-
8	LAW2007	Introduction to Insurance Law	2	0	0	2	F	HP	-	-	-
9	LAW2008	Introduction to Labour Law	2	0	0	2	F	HP	-	-	-
10	LAW2009	Introduction to Law of Marriages	2	0	0	2	F	HP/GS	-	-	-
11	LAW2010	Introduction to Patent Law	2	0	0	2	F	HP	-	-	-
12	LAW2011	Introduction to Personal Income Tax	2	0	0	2	F	HP	-	-	-
13	LAW2012	Introduction to Real Estate Law	2	0	0	2	F	HP	-	-	-
14	LAW2013	Introduction to Trademark Law	2	0	0	2	F	HP	-	-	-
15	LAW2014	Introduction to Competition Law	3	0	0	3	F	HP	-	-	-
16	LAW2015	Cyber Law	3	0	0	3	F	HP	-	-	-
17	LAW2016	Law on Sexual Harassment	2	0	0	2	F	HP/GS	-	-	-
18	LAW2017	Media Laws and Ethics	2	0	0	2	F	HP/GS	-	-	-
Mathematics Basket											
1	MAT2008	Mathematical Reasoning	3	0	0	3	S	-	-	-	-
2	MAT2014	Advanced Business Mathematics	3	0	0	3	S	-	-	-	-
3	MAT2041	Functions of Complex Variables	3	0	0	3	S	-	-	-	-
4	MAT2042	Probability and Random Processes	3	0	0	3	S	-	-	-	-
5	MAT2043	Elements of Number Theory	3	0	0	3	S	-	-	-	-
6	MAT2044	Mathematical Modelling and Applications	3	0	0	3	S	-	-	-	-
Mechanical Basket (not to be offered for Mechanical Department students)											
1	MEC1001	Fundamentals of Automobile Engineering	3	0	0	3	F	-	-	-	-
2	MEC1002	Introduction to Matlab and Simulink	3	0	0	3	S/EM	-	-	-	-
3	MEC1003	Engineering Drawing	1	0	4	3	S	-	-	-	-
4	MEC2001	Renewable Energy Systems	3	0	0	3	F	ES	-	-	-
5	MEC2002	Operations Research & Management	3	0	0	3	F	-	-	-	-
6	MEC2003	Supply Chain Management	3	0	0	3	S/ EM/ EN	-	-	-	-
7	MEC2004	Six Sigma for Professionals	3	0	0	3	S/EM	-	-	MEC 2008	-
8	MEC2005	Fundamentals of Aerospace Engineering	3	0	0	3	F	-	-	-	-
9	MEC2006	Safety Engineering	3	0	0	3	S/EM	ES	-	-	-
10	MEC2007	Additive Manufacturing	3	0	0	3	F/EM	-	-	-	-

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

21	MGT2018	Professional and Business Ethics	3	0	0	3	S/EM/ EN	HP	-	-	-
22	MGT2019	Sales Techniques	3	0	0	3	S/EM/ EN	HP	-	-	-
23	MGT2020	Marketing for Engineers	3	0	0	3	S/EM/ EN	HP	-	-	-
24	MGT2021	Finance for Engineers	3	0	0	3	S/EM/ EN	HP	-	-	-
25	MGT2022	Customer Relationship Management	3	0	0	3	S/EM/ EN	HP	-	-	-
Media Studies Basket											
1	BAJ3050	Corporate Filmmaking and Film Business	0	0	4	2	EM	HP	-	-	-
2	BAJ3051	Digital Photography	2	0	2	3	EM	HP	-	-	-
3	BAJ3055	Introduction to News Anchoring and News Management	0	0	2	1	EM	-	-	-	-

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 Engineering Block Chain

Table 3.9: 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 Engineering Block Chain

Table 3.10: 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 Cont¯rol	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

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Type of Skill	Pre-requisite	Basket
Semester 1 – 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

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

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 – 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

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Type of Skill	Pre-requisite	Basket
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

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Type of Skill	Pre-requisite	Basket
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	CBC2000	Foundations of Blockchain Technology	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

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Type of Skill	Pre-requisite	Basket
Semester 4			**19**	**2**	**10**	**24**	**31**			
1	CHE7601	Environmental Studies	1	0	0	0	1	F		MAC
2	**MAT2321**	**Discrete Mathematical Structures**	3	1	0	4	4	S/EM		PCC
3	CSS2258	Web Technologies	3	0	0	3	3	S/EM		PCC
4	CSS2259	Web Technologies Lab	0	0	2	1	2	S		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	CSS2262	Analysis of Algorithms	3	1	0	4	4	S		PCC
8	CSS2263	Analysis of Algorithms Lab	0	0	2	1	2	S/EM		PCC
9	CSS2264	Essentials of AI	3	0	0	3	3	S/EM		ESC
10	CSS2265	Essentials of AI Lab	0	0	2	1	2	S		ESC
11	FIN1002	Essentials of Finance	3	0	0	3	3	S		HSMC
12	APT4004	Aptitude Training - Intermediate	0	0	2	0	2	S		MAC
Semester 5			**18**	**0**	**8**	**23**	**26**			
1	CSS2266	Theory of Computation	3	0	0	3	3	S		PCC
2	CSS2502	Cryptography and Network Security	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	CBC2500	Smart Contract and Solidity	3	0	0	3	3	S/EM		PCC
6	CBC2501	Smart Contract and Solidity Lab	0	0	2	1	2	S/EM		PCC
7	CBC2502	Distributed Ledger Technology	3	0	0	3	3	S/EM		PCC
8	CBC2503	Distributed Ledger Technology Lab	0	0	2	1	2	S/EM		PCC
9	CSSXXXX	Professional Elective – I	3	0	0	3	3	S/EM		PEC
10	CSS7000	Internship	-	-	-	2	0	S/EM		PRW
11	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			MAC
2	CBC2504	Blockchain Security and Performance	3	0	0	3	3	S/EM		PCC
3	CBC2505	Blockchain Security and Performance Lab	0	0	4	2	4	S/EM		PCC
4	CBC2506	Blockchain Architecture Design	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	XXXXXX	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	0	0	0	10	0	S/EM		PRW
			117	**4**	**62**	**160**	**183**			

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: 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. Belozerova, and M. Khenner, *Ordinary and Partial Differential Equations*. Boca Raton, FL, USA: CRC Press, 2013.

E-resources/ Web links:

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

5. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_134719 6. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=TEXTBOOK_LIBRARY01_06082022_174
Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of differential calculus, differential equations and statistical methods with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem solving. This is attained through the assessment component mentioned in course handout.

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:** Understand the concepts of electrical conducting properties of metal, semiconductor, and superconductivity. — Level 2 – Understand **CO2:** Understand the principles of quantum mechanics. — Level 2 – Understand **CO3:** Discuss the quantum concepts used in quantum computers. — Level 2 – Understand **CO4:** Explain the applications of lasers and optical fibers in various technological fields. — Level 2 – Understand		
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 I|0> and I|1>, Pauli Matrices and its operations on 0 and 1 states, Mention of Conjugate and Transpose, Unitary Matrix U, Examples: 2x2 Matrices and their multiplication (Inner Product), Probability, Orthogonality

Module 3	Quantum Computing	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

Engineering Physics by Avadhanalu, Revised edition, S. Chand Publications,2024.

Quantum Computation and Quantum Information, Michael A. Nielsen & Isaac L. Chuang, Cambridge Universities Press, 2010 Edition
References: 1. Elementary Solid state Physics: Principles and Applications by M.A. Omar, 1st Edition, Pearson Publications, 2002. 2. Principles of Quantum Mechanics by R Shankar, 2nd edition, springer Publications, 2011. 3. Optoelectronics: An Introduction by John Wilson and John Hawkes, 3rd edition, Pearson Publications, 2017. 4. Engineering Physics by Gaur and Gupta, Dhanpat Rai Publications, 2012. 5. Introduction to Quantum Mechanics, David J Griffiths, Cambridge University Press, 2019
E-Resourses: 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:** 1. **Understand** the principles of interpersonal, intercultural, and professional communication in academic, social, and technical team contexts. 2. **Demonstrate** audience-centered spoken communication in individual presentations, group discussions, and simulated professional scenarios through vocal control, adaptive register, and collaborative engagement. 3. **Interpret** written, multimodal, and AI-generated texts to identify underlying assumptions, rhetorical strategies, logical errors, and credibility markers across diverse media and professional contexts. 4. **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. 5. **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 *(including cross-cultural dimensions)*

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 *(patterns of weak argumentation in everyday contexts)*
- 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 *(includes India's Digital Personal Data Protection Act, 2023)*
- 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:

1. **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.
2. **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.
3. **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.
4. **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.

5. **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.
6. **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.
7. **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). *Technical Communication* (13th ed.). Bedford/St. Martin's.

Reference Book(s):

R1. Carnegie, D. (1936/1998). *How to Win Friends and Influence People*. Simon & Schuster.

R2. Orwell, G. (1946). *Politics and the English Language*.

R3. Strunk, W., & White, E. B. (1999). *The Elements of Style* (4th ed.). Longman.

R4. Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

R5. O'Neil, C. (2016). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishers.

R6. Adler, R. B., Rodman, G., & DuPré, A. (2019). *Understanding Human Communication* (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**

Topics:

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

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

Topics:

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

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

Topics:

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

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

Topics:

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

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

Files and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter's age validation, Marks range validation (0-100), 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 Name: Digital Design Type of Course: Theory Course	L- T-P- C	2-0-0-2
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	ECE2052		
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 Objective	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 Outcomes	On successful completion of this course the students shall be able to: CO1. Explain the fundamental concepts of Number system, Boolean algebra and Logic gates CO2. Determine the minimum literal Boolean Expression using K-Map minimization techniques CO3. Make use of relevant logic gates for constructing the combinational circuit. CO4. Produce standard and real world combinational logic circuits using minimum number of logic gates CO5. Demonstrate the working principles of flip-flops and latches within the context of sequential logic circuits. CO6. Design synchronous and asynchronous logic circuit using Flip-flops		
Course Content:			
Module 1	Number System, Logic Gates & Boolean function simplification (Comprehension Level)	Assessment	10 Sessions
Topics: 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: (Application Level)	Assessment	10 Sessions
Topics: 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 (Application Level)	Quiz	10 Sessions
Topics:			

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.

Text Book

T1: Thomas L Flyod, "Digital Fundamentals", Eleventh Edition, Pearson Education. ISBN-10: 132737965. (2014).eBook-[PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org

T2: Mano, M. Morris and Ciletti Michael D., "Digital Design", Pearson Education, 6th edition ., [PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download (studymaterialz.in)

References

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

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

E-Resources/Web Resources

1. https://openFullText.html?DP=https://nptel.ac.in/courses/108105132
2. https://openFullText.html?DP=https://ocw.mit.edu/courses/6-111-introductory-digital-systems-laboratory-fall-2006/
3. https://openFullText.html?DP=https://www.javatpoint.com/digital-electronics
4. https://openFullText.html?DP=https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/
5. https://openFullText.html?DP=https://circuitverse.org

Course Code: ECE1512	Course Title: Design Thinking in Practice	L- T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	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		

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:** Understand electrical and optical properties of materials. Level 2 – Understand **CO2:** Interpret the results of various experiments to verify the concepts used in optoelectronics and advanced devices. Level 2 – Understand
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 (Audited Course)** **Type of Course: HSMC**	**L- T - P- C**	0-0-2-0
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course is designed to enable students to set SMART goals, form Professional & personal ethics for success and learn various email writing techniques. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.		
Course Objective	The objective of the course is to **familiarize the learners with the concepts of "Employability for Young Professionals" and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.**		
Course Out Comes	**On successful completion of this course the students shall be able to:** CO 1 **Define** their career goals CO 2 **Practice ethical habits** for better career success CO3 **Demonstrate** effective email writing techniques		
Course Content			

Module 1	**Goal Setting & Grooming**	Classroom activities	**10 Sessions**
Topics: SMART Goals, formal grooming through self-introduction activity **Activity:** Real world scenarios			
Module 2	**Habit Formation**	Role plays	**10 Sessions**
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 Sessions**
Topics: Types of prompts to generate effective or desired results for email etiquettes **Activity:** Individual student presenting various search prompts			
Faculty: L&D			

Targeted Application & Tools that can be used: 1. TED Talks 2. You Tube Links 3. Activities
Assignment proposed for this course Assignment 1: SMART Goal Assignment 2: AI tools for prompt search
Continuous Individual Assessment Module 1: Presentation Module 2: Activity based assessment Module 3: Class assessment
The topics related to skill development: Students acquire knowledge on SMART goals, implement grooming standards, practice ethical behavior in class and campus, acquire hands-on experience to use AI tools to get search prompts for desired email etiquettes.

Course Code: ECE2052	Course Name: Digital Design Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	ECE2022		
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 Objective	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 Outcomes	On successful completion of the course the students shall be able to CO1. Interpret Boolean laws and the logic gates functionality using available circuit elements in simulator platforms. CO2. Apply the fundamental laws and rules of Boolean algebra for simplified circuit simulation using any simulator tool. CO3. Develop combinational logic circuits by connecting offered gates for basic arithmetic operation CO4. Make use of circuit elements for ensuring the digital logic circuit's truth table perfection. CO5. Examine the performance of sequential logic circuit simulation using flip-flop and latch elements CO6. Demonstrate the operation of clocked and edge-triggered sequential components in an open-source simulation environment		

Course Content:

List of Laboratory Tasks:

- **Experiment 1: Logic Gates**.
  Study and verify the operation of basic logic gates such as AND, OR, NOT, NAND, NOR, XOR, and XNOR.
- **Experiment 2: Boolean Algebra**

  Implement and verify Boolean algebra laws and simplification techniques using logic circuits.
- **Experiment 3: Adder**

  Design and analyze half adder and full adder circuits for binary addition.
- **Experiment 4: Subtractor**

  Design and verify half subtractor and full subtractor circuits for binary subtraction.
- **Experiment 5: Magnitude Comparator**

  Design a circuit to compare two binary numbers and determine equality or greater/less conditions.
- **Experiment 6: Design of Combinational Circuits**

  Design and implement combinational logic circuits using Boolean expressions and K-maps.
- **Experiment 7: Multiplexer, Demultiplexer & Applications of MUX**

  Study the working of multiplexers and demultiplexers and implement logic functions using MUX.
- **Experiment 8: Flip-Flops**

  Implement and verify the operation of SR, JK, D, and T flip-flops.
- **Experiment 9: Shift Registers**

  Design and analyze serial and parallel shift registers for data storage and transfer.
- **Experiment 10: Counter Design**

  Design and implement synchronous and asynchronous counters using flip-flops.

Targeted Applications & Tools:

CIRCUIT VERSE	An online digital circuit simulation platform used for designing and testing logic circuits and combinational/sequential circuits.
HDL	Hardware Description Language used to model, design, and simulate digital electronic systems and circuits.
VHDL	Very High-Speed Integrated Circuit Hardware Description Language used for designing, simulating, and implementing digital systems in FPGA and ASIC design.

Skill-Building Activities through Experiential Learning:

- Design and simulation of digital circuits using CircuitVerse/HDL tools.
- Hands-on implementation of combinational and sequential circuits.
- Mini exercises on logic simplification and circuit optimization.
- Team-based circuit design and debugging activities.
- Real-time verification of flip-flops, counters, and shift registers.

Text Book

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

E-Resources/Web Resources

1. https://circuitverse.org/
2. https://ocw.mit.edu/courses/6-111-introductory-digital-systems-laboratory-fall-2006/
3. https://nptel.ac.in/courses/108105132
4. https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/
5. https://www.tutorialspoint.com/digital_electronics/index.htm
6. https://github.com/logisim-evolution/logisim-evolution
7. https://www.javatpoint.com/digital-electronics

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.					

<table>
<tr><td>Course Outcomes</td><td colspan="2">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)</td></tr>
<tr><td>Course Content:</td><td colspan="2"></td></tr>
<tr><td>Module 1</td><td>Integral Calculus and Special Functions</td><td>(8L+3T=11 Sessions)</td></tr>
<tr><td colspan="3">Definite and indefinite integrals, Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Surface areas, volumes, Sterlin approximation theorem.</td></tr>
<tr><td>Module 2</td><td>Vector Spaces and Inner Product Spaces</td><td>(12L+4T=16 Sessions)</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td>Module 3</td><td>Linear Transformations and Matrix Decompositions</td><td>(10L+3T=13 Sessions)</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td>Module 4</td><td>Probability and Random Variables for machine learning</td><td>(15L+5T =20 Sessions)</td></tr>
<tr><td colspan="3">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</td></tr>
<tr><td colspan="3">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.</td></tr>
</table>

- 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

1. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.
2. B. S. Grewal, *Higher Engineering Mathematics*, 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:

1. G. Strang, *Introduction to Linear Algebra*, 6th ed. Wellesley, MA, USA: Wellesley-Cambridge Press, 2023.
2. D. C. Lay, S. R. Lay, and J. J. McDonald, *Linear Algebra and Its Applications*, 6th ed. Harlow, U.K.: Pearson, 2021.
3. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.
4. B. S. Grewal, *Higher Engineering Mathematics*, 45th ed. New Delhi, India: Khanna Publishers, 2017.
5. C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, 2nd ed. Philadelphia, PA, USA: SIAM, 2023.

E-resources/ Web links:

1. https://open.umn.edu/opentextbooks/textbooks/24
2. https://open.umn.edu/opentextbooks/textbooks/introduction-to-probability
3. 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	

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

CIV1200</td><td colspan="2">Course Name: Foundations of Integrated Engineering

Type of Course: Theory Course</td><td>L- T-P- C</td><td>2-0-0-2</td></tr>
<tr><td>Version No.</td><td colspan="4">1.0</td></tr>
<tr><td>NCrF Level-</td><td colspan="4">5.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Co-requisites</td><td colspan="4"></td></tr>
<tr><td>Anti-requisites</td><td colspan="4">CSEXXXX</td></tr>
<tr><td>Course Description</td><td colspan="4">This interdisciplinary course introduces first-year engineering students to foundational principles and practices across key engineering domains, emphasizing real-world problem-solving, sustainability, and ethical innovation. Students explore how civil, mechanical, electrical, and IT systems intersect with emerging technologies like IoT, AI, and geomatics to address global challenges. Through case studies, learners gain deeper understanding of smart infrastructure, prototyping mechanical/electronic systems, and securing IT solutions. Topics include bioinformatics for environmental monitoring, GIS-enabled urban planning, renewable energy integration, and cybersecurity fundamentals. The course cultivates a holistic understanding of engineering's role in sustainable development, safety, and ethical decision-making, preparing students to contribute meaningfully to multidisciplinary projects in a technology-driven world.</td></tr>
<tr><td>Course Objective</td><td colspan="4">The objective of the course is skill development of student by using Participative Learning techniques.</td></tr>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of the course the students shall be able to:

CO1: Recall key principles of Agile, DevOps, and bioinformatics used in interdisciplinary engineering contexts. L1– Remember
CO2: Explain the role of GIS, LiDAR, and sustainable materials in designing smart infrastructure and disaster management systems. L2– Understand
CO3: Describe core components of mechanical systems and their real-world applications. L3 – Apply
CO4: Describe the functionality of IoT-enabled wearable devices, embedded systems, and renewable energy integration in smart grids. L4 – Analyze
CO5: List foundational IT concepts such as cloud computing architectures, cybersecurity threats, and blockchain applications. L5 – Evaluate</td></tr>
<tr><td colspan="5">Course Content:</td></tr>
<tr><td>Module 1</td><td>Foundations of Engineering Practice</td><td>Assignment</td><td>Case studies</td><td>6 Sessions</td></tr>
<tr><td colspan="5">Real-world problem-solving using data logic and practical applications, Collaboration and Innovation through multi-domain project, Engineering Ethics & Environmental Impact
Emerging Fields: Automation, and Introduction to bioinformatics and its application
Sustainability & Safety: Circular economy principles, carbon footprint analysis.</td></tr>
<tr><td>Module 2</td><td>Civil Engineering & Geomatics</td><td>Assignment</td><td>Article Review</td><td>6 Sessions</td></tr>
<tr><td colspan="5">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.</td></tr>
</table>

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:

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

References

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

Web-resources:

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

Course Code: 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.

Self-learning topics: 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.

Photovoltaics: Solar cells - Construction and working principle; types- Inorganic, Organic and quantum dot sensitized (QDSSC's).

Self -learning topics: Battery technology for e-mobility, Green hydrogen

Targeted Application & Tools that can be used:

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

Tools: Molview, chemdraw, excel etc

Project work/Assignment:

Assessment Type

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

Text Book

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

Reference Books

1. Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press,
2. ISBN: 978-036-727-510-5.
3. E-waste recycling and management: present scenarios and environmental issues, Khan, Anish, and
4. Abdullah M. Asiri. 2019, Springer, Vol. 33. ISBN: 978-3-030-14186-8.
5. Essentials of computational chemistry: theories and models, Christopher J Cramer, 2013, John Wiley &
6. Sons. ISBN: 978-0-470-09182-1.

7. Energy storage and conversion devices: Supercapacitors, batteries and hydroelectric cells, Anurag Gaur,
8. L. Sharma, Anil Arya. 2021, CRC press, 1st edition, ISBN: 978-1-003-14176-1.
9. Fundamentals of analytical chemistry: An introduction, Douglas A. Skooget etal., 2004 Thomson Asia pte
10. Ltd., 8th, ISBN: 978-0-495-55828-6
11. Functional and smart materials, Chander Prakash, Sunpreet Singh, J. Paulo Davim, 2020, CRC Press,
12. ISBN: 978-036-727-510-5.
13. Electrical and electronic devices, circuits and materials: Technological challenges and solutions. Tripathi,
14. L., Alvi, P. A., & Subramaniam, U, 2021, John Wiley & Sons, ISBN: 978-0367564261.
15. F. Jensen, Introduction to Computational Chemistry, 3rd edition, Wiley, 2017.

E resources

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

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,[Level 2 – Understand] pseudo code, and flowcharts for developing programs		

	CO2: Apply fundamental C programming constructs such as data types, operators, decision-making, and looping statements to develop simple programs. [Level 3 – Apply] **CO3:** Develop programs using arrays and strings to solve computational problems effectively [Level 3 – Apply] **CO4:** Implement modular and reusable programs using functions, recursion, and pointers. [Level 3 – Apply] **CO5:** Design applications using structures, unions, file handling concepts, and demonstrate the role of C programming in emerging technologies. [Level 3 – Apply]		
Course Content:			
Module 1	**Introduction to C Language**	**Assignment (Algorithms, Flowcharts, and Basic C Programs)**	**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**	**Programming Assignment (Arrays, Sorting, Searching, and String Handling in C)**	**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**	**Programming Assignment (Functions, Recursion, and Pointer Implementation in C)**	**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**	**Assignment (Structures, Unions, and File Operations in C)**	**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**	**Mini Project (C Programming Applications in Cyber Security, Blockchain, and AI)**	**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	*Communication in Practice* 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 *what* communication is, but *how* 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:** 6. **Explain** the principles governing professional communication identity, communicator ethics, and context-sensitive self-presentation across workplace, hierarchical, and digital contexts. 7. **Demonstrate** advanced vocal, rhetorical, and adaptive strategies across high-stakes spoken communication genres, including interviews, pitches, negotiations, and leadership scenarios. 8. **Demonstrate** advanced vocal, rhetorical, and adaptive strategies across high-stakes spoken communication genres, including interviews, pitches, negotiations, and leadership scenarios. 9. **Execute** the production of professional-grade written artifacts across high-demand genres—including formal reports, business proposals, data communications, and advocacy briefs—meeting real-world standards of clarity, structure, and audience appropriateness. 10. **Implement** AI-assisted communication workflows in professional contexts, including systematic prompt design for high-stakes outputs, validation protocols for synthetic materials, and the application of ethical frameworks at individual and organizational levels.					

Course Content: Communication in Practice				
Module 1	**Words that Work: Vocabulary, Register, and Precision**	**CA1**	**Register Switching Activity**	**09 Sessions**

The topics will be explored through vocabulary games, role-play activities, communication challenges, case-based tasks, and workplace simulations:

- Why Words Matter in Professional and Technical Communication
- Choosing the Right Words for Different Situations
- Understanding Meaning: Positive, Negative, and Neutral Word Choices
- Commonly Confused Words and Miscommunication in Professional Contexts
- Using Clear, Precise, and Concise Language
- Explaining Technical Terms to Non-Technical Audiences
- Building and Using Personal/Professional Vocabulary

Module 2	**Listening, Processing, and Structured Response Writing**	**CA2**	**Rapid Listening and Response**	**12 Sessions**

The topics will be explored through podcasts, interviews, workplace recordings, collaborative listening tasks, discussions, and real-time writing activities:

- Identifying Key Information from Spoken Inputs
- Organizing Information from Discussions and Conversations
- Converting Spoken Communication into Clear and Structured Written Responses
- Writing Summaries from Audio and Discussion Inputs
- Writing Follow-Up Responses Based on Spoken Communication
- Writing Action Notes, Instructions, and Task Updates from Listening Inputs
- Converting Meetings and Discussions into Structured Written Outputs

Module 3	**High-Stakes Spoken Communication**	**CA**	**Mock Professional Interview, Pitching an Idea Pitch**	**12 Sessions**

- Speaking as a professional: identity, presence, and the first impression
- Vocal authority, physical presence, and communication across hierarchies
- Interview communication: structure, composure, and the STAR framework
- Pitching ideas: the 3-minute and 10-minute pitch architecture
- 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
- 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 Name: Basics of Electrical and Electronics Engineering. Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	2.0		

Course Pre-requisites	NIL
Co-requisites	EEE1251
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 the course the students shall be able to:** CO1: Apply basic laws of Electrical Engineering to compute voltage, currents and other parameters in the circuits. [Level 3 - Apply] CO2: Discuss various fundamental parameters appearing in the characteristics of semiconductor devices and their applications. [Level2– Understand] CO3: Summarize the operations of different biasing configurations of BJTs and amplifiers. [Level2– Understand] CO4 :Discuss the performance characteristics and applications of various electrical Machines. [Level2– Understand]] CO5: Apply basic laws of Electrical Engineering to compute voltage, currents and other parameters in the circuits. [Level 3 – Apply]

Course Content:

Module 1	Introduction to Electrical Circuits	Assignment/ Quiz	10 Sessions

Topics:

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	Assessment : Quiz + Assignment	10 Sessions

Topics:

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	Assessment : Assignment/ Quiz	10 Sessions

Topics:

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	Assessment : Assignment/ Quiz	10 Sessions

Topics:

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 Applications & Tools:

Multisim	An electronic circuit simulation software used for designing, testing, and analyzing analog and digital circuits.
PSpice	A simulation tool used to analyze the behavior and performance of electronic circuits.
Multimeter	An electronic measuring instrument used to measure voltage, current, and resistance in circuits.
Function Generator	A device used to generate different electrical waveforms for circuit testing and analysis.
Power Supplies	Equipment used to provide regulated electrical power to electronic circuits and devices.

Skill-Building Activities through Experiential Learning:

- Hands-on experiments on electrical circuits and electronic components.
- Measurement and analysis of voltage, current, resistance, and waveform characteristics using electrical instruments.
- Simulation and testing of electronic circuits using software tools such as Multisim and PSpice.
- Practical study of semiconductor devices, BJTs, amplifiers, and electrical machines.
- Team-based troubleshooting and circuit debugging activities to enhance problem-solving skills.

Text Book

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: J. Millman, C. C. Halkias and C. D. Parikh, "Millman's Integrated Electronics", McGraw Hill Education, 2nd Edition.
Basics of Electrical & Electronics Laboratory Manual.

R1: Re 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.

E-Resources/Web Resources

1. https://www.digimat.in/nptel/courses/video/108105112/L01
2. https://nptel.ac.in/courses/108/105/108105153/
3. https://nptel.ac.in/courses/108/102/108102095

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 complete knowledge of C language. Students will be able to develop logics which will help them to create programs and applications in C. ACAlso by learning the basic programming constructs they can easily switch over to any other language in future.					
Course 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:** 1. Describe algorithms with flowcharts for solving problems [Understand] 2. Demonstrate the application of C programming constructs in developing simple programs [Apply] 3. Implement applications using arrays and strings [Apply] 4. Apply functional programming principles to construct modular and reusable software components for solving complex problem [Apply] 5. Develop applications in C using structures and unions [Apply] 6. Differentiate between Sequential and Random Access File Processing through applications. [Analyze]					
Course Content:						

List of Practicals: **60 Sessions**

Lab Sheet 1: 10 Sessions

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

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

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

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

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

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

Text Book(s):

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

Reference Book(s):

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

Web Links and Video Lectures:

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

Course Code: 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			
Anti-requisites			
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 used in chemistry laboratories for quantitative analysis. CO2: estimate the presence of acids and metal ions in domestic and industrial waste using laboratory techniques. CO3: review the experimental results and demonstrate improved experimental skills through hands-on laboratory experience. CO4: classify laboratory techniques such as experimental 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 (Audited Course)** Type of Course: Practical Only Course	L- T - P- C	0	0	2	0
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	NIL					
Course Description	This course is designed to enable students learn styles of communication, team building and use empathy in leadership. The course will benefit learners in preparing themselves effectively through various activities and learning methodologies.					
Course Objective	The objective of the course is to **familiarize the learners with the concepts of "Industry 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 Sessions**
Topics: Practice effective communication skills (Verbal, Non-verbal, Written and Visual) **Activity:** Use social media prompts to prepare self-introduction videos			
Module 2	**Team Building**	Group Activity	**10 Sessions**
Topics: Skills of an effective team player **Activity:** Student group activity to build class networking			
Module 3	**Leadership**	Case study	**10 Sessions**
Topics: Types of leadership, using empathy in leadership **Activity:** Individual presentation by students on corporate leaders.			
Faculty : L&D			
Targeted Application & Tools that can be used: 1. TED Talks 2. You Tube Links 3. Activities			
Assignment proposed for this course Assignment 1: 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: **Experiment No 1:** Familiarization of passive elements. Level 1: Identify and understand the function and ratings of basic passive elements. Level 2: Study the behavior of passive elements in simple electrical circuits by measuring voltage, current, and impedance. **Experiment No 2:** Experimental Verification of Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL). Level 1: Study and Verify KVL and KCL for the given electrical Circuit. **Experiment No 3:** Determination of Power Factor, and Impedance for R and RL Loads. Level 1: Determine the power factor and impedance of R and RL loads under steady-state AC supply conditions. Level 2: Analyze the effect of inductance on power factor, impedance, and phase angle for RL loads under varying load conditions.

	Experiment No 4: Modeling and Simulation of effective resistances in different circuit connections Using Simulink. Level 1: Model and simulate series and parallel resistor circuits in Simulink and determine their effective resistance. Level 2: Analyze the effect of different resistor combinations on total resistance, current distribution, and voltage drop. **Experiment No 5:** Simulation and analysis of line and phase voltages in Star-connected and Delta- connected supply systems. Level 1: Model and simulate star-connected and delta-connected three-phase supply systems and observe line and phase voltages. Level 2: Analyze and compare the relationship between line and phase voltages in star and delta connections. **Experiment 6:** Determination of transformation ratio of a Single-Phase Transformer Under different tapping conditions. Level 1: Determine the transformation ratio of a single-phase transformer under no-load condition for various tap settings. Level 2: Analyze the effect of different tapping conditions on secondary voltage, regulation, and transformer performance. **Experiment No. 7:** Study of the speed-torque characteristics of a Three-Phase Induction Motor. Level 1: Study and analyze the speed–torque characteristics of a three-phase induction motor. Level 2: Evaluate the effect of supply voltage and current on motor performance parameters such as slip, and torque. **Experiment 8:** Performance Analysis of Half-Wave and Full-Wave Rectifier Circuits Using Simulink. Level 1: Model and simulate half-wave and full-wave rectifier circuits in Simulink and analyze their output waveforms. Level 2: Evaluate and compare rectifier performance parameters such as ripple factor, efficiency, and output voltage. **Experiment 9:** Study of the characteristics of a Zener Diode. Level 1: Study the forward and reverse V–I characteristics of a Zener diode using DC analysis. Level 2: Analyze the Zener breakdown characteristics and determine the Zener voltage and regulation behavior. **Experiment 10:** Study of the Output and Transfer Characteristics of a MOSFET. Level 1: Study, analyze and implement the output characteristics of a MOSFET under different gate-source voltage conditions and observe their results. Level 2: Analyze the transfer characteristics of a MOSFET and evaluate key parameters such as threshold voltage and transconductance.

Targeted Application & Tools that can be used:

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

Professionally Used Software: MATLAB SIMULINK

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

Course Material

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: 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: CO1: Understand the concepts of human values, self-exploration, and holistic development in personal and social contexts. [Level 2 – Understand] CO2: Apply principles of human values and ethical conduct in personal, social, and professional life situations. [Level 3 – Apply] CO3: Apply the concepts of harmony and ethical reasoning to address issues in family, society, and professional environments. [Level 3 – Apply] CO4: Analyze the relationship between self and body, and the role of values in maintaining harmony within individuals and society. [Level 4 – Analyze] CO5: Evaluate personal beliefs, behaviors, and professional practices based on universal human values and ethical principles. [Level 5 – Evaluate]			
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*:

1) UHV II - https://www.youtube.com/watch?v=NhFBzn5qKIM&list=PLWDeKF97v9SO8vvjC1KyqteziTbTjN1So&pp=0gcJCWMEOCosWNin
2) Lecture by Dr. Kumar Sambhav, NPTEL course: Universal Human Values, https://onlinecourses.swayam2.ac.in/aic22_ge23/preview
3) Lecture by Dr. Padmavati, Dr Narendran Thiruthy, NPTEL Course: Biodiversity Protection, Farmers and Breeders Rights, https://nptel.ac.in/courses/129105008, 2024.

* Other source links are available in below Resources link.

Text Book

3. 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
4. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2019.
5. Premvir Kapoor, Professional Ethics and Human Values, Khanna Book Publishing, New Delhi, 2022.

Reference Books

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

Resources:

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

Topics relevant to Skill Development:

1. An attitude of enquiry.
2. Write reports

The topics related to Human values and Professional ethics:

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

Course Code: MAT2311	**Course Title: Vector Calculus, Numerical Methods and Sampling Distributions [Type of Course:1] School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**

Version No.	1.0	
Course Pre-requisites	NIL	
Anti-requisites	NIL	
Course Description	This course introduces the fundamental concepts and methods of ordinary differential equations, numerical techniques, and statistical inference used in engineering and scientific applications. The course covers analytical methods for solving differential equations, numerical methods for solving linear and nonlinear equations, interpolation, numerical differentiation and integration, and numerical solutions of ordinary differential equations. It also introduces sampling distributions and hypothesis testing for statistical decision-making. The course enables students to apply mathematical and statistical techniques to solve computational and real-world engineering problems.	
Course Objective	The objectives of this course are to develop the ability to solve ordinary differential equations arising in scientific and engineering applications. Introduce numerical techniques for solving systems of linear equations and nonlinear algebraic equations. Apply interpolation, differentiation, and integration methods for approximating mathematical functions and evaluating definite integrals. Employ numerical methods for obtaining approximate solutions of ordinary differential equations. Apply concepts of sampling distributions and statistical inference to estimate population parameters and support decision-making. Apply hypothesis testing techniques to real-world problems involving means, proportions, variances, and associations among variables.	
Course Outcomes	After successful completion of the course, students will be able to: **CO1:** Employ vector differential operators to solve problems involving scalar and vector fields and classify them as solenoidal or irrotational (L3) **CO2:** Apply vector integration techniques and integral theorems to a solve engineering problems. (L3) **CO3:** Use numerical methods to solve systems of linear and nonlinear equations. (L3) **CO4:** Implement numerical techniques for interpolation, differentiation, integration, and ordinary differential equations. (L3) **CO5:** Utilize sampling distributions and statistical inference techniques to solve statistical problems. (L3) **CO6:** Perform hypothesis testing for statistical decision-making. (L3)	
Course Content:		
Module 1	**Vector Calculus**	**(14L+5T=19 Sessions)**
Vector & Scalar Functions and Fields, Derivatives, Curve, Arc length, Curvature & Torsion, Gradient of Scalar Field, Directional Derivative, Divergence of a Vector Field, Curl of a Vector Field, Physical interpretation, Solenoidal and Irrotational Vector Fields. Problems. Line Integrals, Path Independence of Line Integrals, Surface and Volume Integrals, Green's Theorem in the plane, Divergence Theorem of Gauss, Stokes's Theorem. Problems on Image Edge Detection, Robot Path Planning and 3D Surface Modelling		
Module 2	**Solution of Linear Systems and non-linear of Equation**	**(9L+3T=12 Sessions)**
Introduction, Jacobi iteration method, Gauss-Seidel iteration method. Solution of Algebraic and Transcendental Equations: Bisection method, Regula-Falsi method, Newton-Raphson method, secant method, fixed point iteration method.		

Page Ranking, Network Analysis, Circuit Simulation, Root Finding, Load Balancing, and Computational Optimization Problems.

Module 3	**Numerical Interpolations and Differential Equations**	**(10L+3T=13 Sessions)**

Newton's method, Divided difference method, Lagrange's method, Numerical differentiation, Numerical integration: Trapezoidal rule, Simpson's one-third rule, Simpson's three-eighth rule, Gaussian quadrature rule. Solution of ordinary differential equations: Taylor series method, modified Euler's method, Runge-Kutta method for 4th order. Problems on Signal Reconstruction, Sensor Data Interpolation and Trajectory Prediction.

Module 4	**Sampling Distributions**	**(12L+4T=16 Sessions)**

Random sampling, sampling distributions, Standard Error, Type I & Type II errors, Testing of Hypothesis, Test of significance - Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations, Test for single mean, difference of means, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes. A/B Testing, Fraud Detection, Spam Classification Evaluation, and Network Traffic Analysis.

Targeted Applications & Mathematical Tools:

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

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

Assignment:

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

Text Book

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

References:

1. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, 7th ed. New Delhi, India: New Age International Publishers, 2019.
2. S. S. Sastry, *Introductory Methods of Numerical Analysis*, 5th ed. New Delhi, India: PHI Learning Pvt. Ltd., 2012.
3. J. Kiusalaas, *Numerical Methods in Engineering with Python 3*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2025.

4. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 8th ed. Hoboken, NJ, USA: John Wiley & Sons, 2023.
5. D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications*, 7th ed. Belmont, CA, USA: Cengage Learning, 2008.
6. E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
7. R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*, 11th ed. Boston, MA, USA: Cengage Learning, 2021.
8. S. C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2018.

E-resources/ Web links:

1. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_53751
2. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=TEXTBOOK_LIBRARY01_06082022_117
3. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7313
4. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=CUSTOM_PACKAGE_EBSCO_29052023_228965
5. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_8838
6. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=SPRINGER_OPEN_ACCESS_BOOKS_10022025_1861
7. https://presiuniv.knimbus.com/portal/v2/default/viewDetail?searchResultType=ECATALOGUE_BASED&uniqueId=DOAB_1_06082022_7312

Topics relevant to SKILL DEVELOPMENT: Identification and solution of differential equations, application of numerical methods for algebraic and differential equations, implementation of interpolation and numerical integration techniques, statistical data analysis and hypothesis testing, and use of Python/MATLAB for engineering problem solving.

Catalogue prepared by	**Dr. V Ramalatha**
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

Course Code: CSS2251	Course Name: Data Communications and Computer Networks Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			

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 Objective	The objective of the course is to familiarize the learners with the concepts of Data Communications and Computer Networks and attain Employability through Problem Solving Methodologies.
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Illustrate The Basic Concepts Of Data Communication And Computer Networks. [Level 3 - Apply] **CO2**: Analyze the functionalities of the Data Link Layer. [Level 4 – Analyze] **CO3:** Apply the Knowledge of IP Addressing and Routing Mechanisms in Computer Networks. [Level 3 – Apply] **CO4:** Demonstrate the working principles of the Transport layer and Application Layer. [Level 3 – Apply]

Course Content:

Module 1	Introduction and Physical layer	Assessment: Assignment (Problem Solving)	9 Sessions

Topics:

Introduction to Computer Networks and Data communications, Network Components – Topologies, Transmission Media –Reference Models - TCP/IP Suite, OSI Model .

Physical Layer -Analog and Digital Signals – Digital and Analog Signals – Transmission - Multiplexing and Spread Spectrum.

Module 2	Data Link Layer	Assessment: Assignment (Problem Solving)	12 Sessions

Topics:

Data Link Layer - Error Detection and Correction– Parity, CRC, Hamming Distance Flow Control and Error Control, Stop and Wait, Multiple Access Protocols, CSMA/CD,CSMA/CA

Module 3	Network Layer	Assessment: Assignment (Problem Solving)	13 Sessions

Topics:

Network Layer Services - Network Layer Services, Switching Techniques, IP Addressing methods- IPv4 IPV6 – Subnetting. Routing, - Distance Vector Routing, Link State Routing, RIP, OSPF, BGPV4.

Module 4	Transport and Application Layer	Assessment: Assignment (Problem Solving)	11 Sessions

Topics:

Transport Layers - Connection management – Flow control-Sliding Window, Go-Back N ARQ, Selective Repeat ARQ, UDP, TCP, congestion control, Congestion avoidance The Application Layer: Domain Name System (DNS), Domain Name Space, FTP, Electronic Mail (SMTP), HTTP.

Targeted Applications & Tools: Cisco Packet Tracer, Wireshark, and NS2 Simulator.

Skill-Building Activities through Experiential Learning:

Text Book **T1:** T1. Behrouz A. Forouzan, "Data Communications and Networking with TCP/IP Protocol Suite", 6th Edition, Tata McGraw-Hill, 2022. **T2:** Andrew S Tanenbaum, Nick Feamster & David J Wetherall, "Computer Networks" Sixth Edition, Pearson Publication, 2022.
References **R1:** "Computer Networking: A Top-Down Approach", Eighth Edition, James F. Kurose, Keith W. Ross, Pearson publication, 2021. **R2:** William Stallings, Data and Computer Communication, 8th Edition, Pearson Education, 2007. **R3:** Behrouz A. Forouzan, "Data Communications and Networking 5E", 5th Edition, Tata McGraw-Hill, 2012
E-Resources/Web Resources 1. https://www.geeksforgeeks.org/what-is-spread-spectrum/ 2. https://www.geeksforgeeks.org/difference-between-fdma-tdma-and-cdma/ 3. https://archive.nptel.ac.in/courses/106/105/106105183/ 4. http://www.nptelvideos.com/course.php?id=393 5. https://www.digimat.in/keyword/106.htmlhttps://puniversity.informaticsglobal.com/login

Course Code: CSS2252	Course Name: Data Communications and Computer Networks Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			
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	The objective of this lab course is to provide students with practical, hands-on experience in the configuration, operation, and troubleshooting of data communication systems and computer networks. Through guided experiments and real-world scenarios, students will reinforce theoretical knowledge, develop essential technical skills, and gain a deeper understanding of networking concepts, protocols, and devices used in modern communication systems.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Design and configure basic network topologies using routers, switches, and end devices to meet specified requirements [Level 3 – Apply] . **CO2:** Analyze and troubleshoot network connectivity and performance issues using tools such as Wireshark and network simulators. [Level 4 – Analyze]		

	CO3: Demonstrate understanding of key networking protocols (e.g., TCP/IP, ARP, ICMP, DHCP) through practical implementation and observation. [Level 3 – Apply] **CO4:** Apply IP addressing and subnetting techniques to efficiently allocate and manage network resources in various networking scenarios. [Level 3 – Apply]

Course Content:

List of Laboratory Tasks:

- **Lab sheet -1, M-1, 3 [2 Hours]**
- **Experiment No 1:**
  Level 1: Study of basic network commands and network configuration commands.

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

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

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

  Lab sheet – 5, M-3 [2 Hours]
- **Experiment No. 1:**
  Level 2 - Configure the static routing in the Cisco packet tracer.
- **Experiment No. 2:**
  Level 2 - Configure the dynamic routing protocol in the Cisco packet tracer.

  Lab sheet – 6, M-4 [2 Hours]
- **Experiment No. 1:** Configuration of DNS Server with Recursive & Integrative approach in Cisco packet tracer.

  Lab sheet – 7, M-4 [2 Hours]
- **Experiment No. 1:**
  Configure the telnet protocol in the router using the Cisco packet tracer.

  Lab sheet – 8, M-4[2 Hours]
- **Experiment No. 1:**
  Level1- Introduction to NS2 and basic TCL program.

  Lab sheet – 9, M-4 [2 Hours]
- **Experiment No. 1:**
  Level 1: Simulate three node Point to point network using UDP in NS2.
- **Experiment No. 2:**
  Simulate transmission of Ping message using NS2.

  Lab sheet – 10, M-4[2 Hours]
- **Experiment No. 1:**
  Simulate Ethernet LAN using N-node in NS2.

- **Experiment No. 2:**
  Simulate Ethernet LAN using N-node using multiple traffic in NS2

  Lab sheet –11, M-3,4 [2 Hours]
- **Experiment No. 1:**
  Level 1- Introduction to Wire Shark.
- **Experiment No. 2:**
  Level 2- Demonstration of packet analysis using wire shark.

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

Targeted Applications & Tools:

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.

Skill-Building Activities through Experiential Learning:

Text Book

T1: Behrouz A. Forouzan, "Data Communications and Networking 5E", 5 th Edition, Tata McGraw-Hill, 2017.

T2: Andrew S Tanenbaum, Nick Feamster & David J Wetherall, "Computer Networks" Sixth Edition, Pearson Publication, 2022

References

R1: "Computer Networking: A Top-Down Approach", Eighth Edition, James F. Kurose, Keith W. Ross, Pearson publication, 2021.
R2: William Stallings, Data and Computer Communication, 8th Edition, Pearson Education, 2007.
R3: Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.

E-Resources/Web Resources

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

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

3.https://www.youtube.com/watch?v=3DZLItfbqtQ

4.https://www.youtube.com/watch?v=_fIdQ4yfsfM

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

6. **https://puniversity.informaticsglobal.com/login**

Course Code: CSS2253	Course Title: Data Structures Type of Course: Theory Course (PCC)	L-T- P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		

Anti-requisites	NIL
Course Description	This course introduces the fundamental concepts of data structures and 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: CO1: Explain the concepts of basic data structures such as arrays, stacks, and queues and their operations. — Level 2 – Understand CO2: Apply linked list data structures to solve real-world problems. — Level 3 – Apply CO3: Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios. — Level 3 – Apply CO4: Implement and demonstrate various searching and sorting algorithms and analyze the efficiency of data structures and algorithms in terms of time and space complexity. — Level 3 – Apply CO5: Apply data structure concepts in cybersecurity, AI/ML, blockchain, and intelligent computing applications. — Level 3 – 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 Serach 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 etal., Ane books publishers, 2019. Web resources: 1. For theory: https://onlinecourses.nptel.ac.in/noc20_cs85/preview 2. https://puniversity.informaticsglobal.com/login		
Topics relevant to development of "Skill Development": Linked list and stacks Topics relevant to development of "Environment and sustainability: Queues		

Course Code:: CSS2254	Course Title: Data Structures Lab Type of Course:Laboratory Course (PCC)	L-T- P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		

Course Description	This course introduces the fundamental concepts of data structures and to emphasize the importance of choosing an appropriate data structure and technique for program development. This course has theory and lab component which emphasizes on understanding the implementation and applications of data structures using Java programming language. With a good knowledge in the fundamental concepts of data structures and practical experience in implementing them, the student can be an effective designer, developer for new software applications.
Course Objective	To develop programming and problem-solving skills by enabling students to implement various data structures, analyze their performance through experimentation and apply structured approaches to solve computational problems effectively.
Course Outcomes	On successful completion of this course, students will be able to: CO1: Explain the concepts of fundamental data structures such as arrays, stacks, and queues and their operations. — Understand CO2: Implement linear data structures such as linked lists to solve real-world problems. — Apply CO3: Utilize appropriate non-linear data structures such as trees and graphs for given problem scenarios. — Apply CO4: Implement and demonstrate 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: 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 etal., Ane books publishers, 2019. Web resources: 1. For theory: https://onlinecourses.nptel.ac.in/noc20_cs85/preview 2. https://puniversity.informaticsglobal.com/login
Topics relevant to development of "Skill Development": Linked list and stacks Topics relevant to development of "Environment and sustainability: Queues

Course Code: CSS2255	Course Title: 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. — L2 – Understand CO2: Apply the concept of classes, objects, and methods to solve problems. — L3 – Apply CO3: Apply the concept of arrays and strings. — L3 – Apply CO4: Implement inheritance and polymorphism for building secure applications. — L3 – Apply CO5: Apply Java I/O concepts for intelligent data processing, secure file handling, and AI/ML-based application development. — L3 – 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 Title: Object Oriented Programming Using Java Lab Type of Course: Laboratory Course (PCC)	L-T- P- C	0	0	2	1
Version No.	2.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces the core concepts of Object-Oriented Programming. 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. — L3 – Apply CO2: Apply the concepts of classes, objects, and methods to solve problems. — L3 – Apply CO3: Apply the concepts of arrays and strings. — L3 – Apply CO4: Implement inheritance and polymorphism to develop secure applications. — L3 – Apply CO5: Apply Java I/O concepts for intelligent data processing, secure file handling, and AI/ML-based application development. — L3 – 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 Title: Computer Organization and Architecture Type of Course: Theory Course (PCC)	L-T-P- C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the core principles of computer architecture and organization from basic to intermediate level. This theory-based course emphasizes 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		

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. https://nptel.ac.in/courses/106105163

3. https://nptel.ac.in/courses/106106092
4. https://puniversity.informaticsglobal.com:2229/login.aspx

Course Code: CBC2000	Course Name: Foundations of Blockchain Technology Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	2.0		
NCrF Level-			
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the foundational principles of blockchain technology, its architecture, components, and applications. It explores the working of distributed ledgers, cryptographic fundamentals, consensus algorithms, and the structure of cryptocurrencies and smart contracts.		
Course Objective	• Understand the fundamentals of distributed ledger technology • Explore cryptographic primitives used in blockchain systems • Explain consensus algorithms and their roles in decentralized networks • Analyze the architecture and components of blockchain platforms • Examine blockchain applications across industries		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Describe the architecture and components of blockchain technology **(Level 2: Understand)** **CO2:** Explain cryptographic techniques and consensus mechanisms in blockchain **(Level 2: Understand)** **CO3:** Demonstrate the creation and validation of transactions and blocks **(Level 3: Apply)** **CO4:** Compare various blockchain platforms and their real-world applications **(Level 4: Analyze)**		
Course Content:			
Module 1	Introduction to Blockchain Technology	Assessment: Conceptual Analysis	10 Sessions
Topics: History of blockchain Evolution from Bitcoin to Web3 Distributed ledger technology			

Key characteristics: immutability, transparency, trust

Use cases and applications

Module 2	Cryptography and Blockchain	Assessment : Case Study	12 Sessions

Topics:

Hash functions (SHA-256)

Digital signatures

Merkle trees

Public and private key cryptography,

Wallets and addresses

Transaction lifecycle

Module 3	Consensus Mechanisms	Assessment : Case Study	11 Sessions

Topics:

Consensus overview

Proof-of-Work (PoW)

Proof-of-Stake (PoS)

Practical Byzantine Fault Tolerance (PBFT)

Delegated Proof of Stake (DPoS)

Comparison of consensus algorithms

Module 4	Blockchain Platforms and Applications	Assessment : Experimental	12 Sessions

Topics:

Bitcoin overview

Ethereum overview

Permissioned vs permissionless blockchains

Hyperledger Fabric basics

Smart contracts

Blockchain in finance healthcare, and supply chain

Targeted Applications & Tools:

Bitcoin Blockchain Explorer	Used for exploring blockchain transactions, wallet addresses, and blocks.
Ethereum Blockchain Platform	Used for smart contract execution and decentralized application development.
Remix IDE	Used for developing, compiling, and testing smart contracts.
MetaMask Wallet	Used for blockchain wallet management and interaction with decentralized applications.
Hyperledger Fabric	Used for permissioned blockchain network development.
Ganache	Used for local blockchain simulation and smart contract testing.

Skill-Building Activities through Experiential Learning:

- Analyze blockchain architectures and decentralized systems.
- Understand and validate blockchain transactions and consensus mechanisms.
- Explore blockchain platforms and smart contract environments.
- Evaluate real-world blockchain applications across domains.

Text Book

T1: Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press

T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 3rd Edition, 2020

References

R1: Antonopoulos, Mastering Bitcoin, O'Reilly Media

R2: Arvind Narayanan et al., Introduction to Cryptography and Blockchain, Princeton Press

R3: Don Tapscott & Alex Tapscott, Blockchain Revolution, Portfolio

R4: Josh Thompson, Blockchain Basics, CreateSpace

E-Resources/Web Resources

1. https://blockgeeks.com
2. https://ethereum.org
3. https://bitcoin.org
4. https://hyperledger.org
5. https://web3.foundation

Course Code: CSS2271	Course Title: 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. — Level 2 – Understand CO2: Identify software requirements and develop design models, including Software Requirement Specification (SRS) documents. — Level 3 – Apply CO3: Apply Agile development practices and DevOps principles in software development. — Level 3 – Apply CO4: Apply software testing strategies, configuration management, and maintenance techniques. — Level 3 – Apply CO5: Evaluate secure and AI-driven software engineering practices to ensure software quality, reliability, scalability, and intelligent automation. — Level 5 – 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/TestRogor
Text Books: T1. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", VII Edition, McGraw-Hill, 2017. T2. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", VI Edition, McGraw-Hill, 2018.
References: R1. Rajib Mall, "Fundamentals of Software Engineering", VI Edition, PHI learning private limited, 2015. R2. Ian Sommerville, "Software Engineering", IX Edition, Pearson Education Asia, 2011. R3. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002 Journals / Magazines • IEEE Software • ACM Transactions on Software Engineering and Methodology • Journal of Systems and Software SWAYAM / NPTEL / MOOCs • NPTEL – Software Engineering (Prof. Rajib Mall, IIT KGP) • Coursera – Agile Development (University of Virginia) • Udemy – DevOps Masterclass • EdX – Software Testing Fundamentals (TUM)
Topics Relevant to "Skill Development": Black box Testing, White box Testing, AI tools for design, modelling, testing and document Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout

Course Code: APT4002	**Course Title: Introduction to Aptitude (Audited)**	**L-T- P- C**	0-2-0-0
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: CO1] **Recall** all the basic mathematical concepts they learnt in high school. CO2] **Identify** the principle concept needed in a question. CO3] **Solve** the quantitative and logical ability questions with the appropriate concept. CO4] **Analyze** the data given in complex problems. CO5] **Rearrange** the information to simplify the question			
Course Content:				
Module 1	Quantitative Ability	Assignment	Bloom's Level : Application	**12 Sessions**
Topics: Introduction to Aptitude, working of Tables, Squares, Cubes				
Module 2	Logical Reasoning	Assignment	Bloom's Level : Application	**18 Sessions**
Topics: Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Ordering and Ranking, Clocks and Calendars, Number Series, Wrong number series, Visual Reasoning				
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS				
Text Book 1. Quantitative Aptitude by R S Aggarwal 2. Verbal & Non-Verbal Reasoning by R S Aggarwal				
References 1. www.indiabix.com 2. www.youtube.com/c/TheAptitudeGuy/videos				
Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.				

Course Code: 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		

<table>
<tr><td>Course Description</td><td>This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamental environmental concepts and their relevance to business operations, preparing them to address forthcoming sustainability challenges. It is designed to equip students with the knowledge and skills needed to make decisions that account for environmental consequences, fostering environmentally sensitive and responsible future managers.
This course is designed to cater to Environment and Sustainability</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is ‘SKILL DEVELOPMENT’ of the student by using ‘PARTICIPATIVE LEARNING’ techniques</td></tr>
<tr><td>Course Outcomes</td><td>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></td></tr>
<tr><td colspan="2">Course Content:</td></tr>
<tr><td>Module 1</td><td>Understanding Environment, Natural Resources, and Sustainability</td></tr>
<tr><td colspan="2">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.</td></tr>
<tr><td>Module 2</td><td>Ecosystems, Biodiversity, and Sustainable Practices</td></tr>
<tr><td colspan="2">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.</td></tr>
<tr><td>Module 3</td><td>Environmental Pollution, Waste Management, and Sustainable Development</td></tr>
<tr><td colspan="2">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.</td></tr>
<tr><td>Module 4</td><td>Social Issues, Legislation, and Practical Applications</td></tr>
<tr><td colspan="2">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)</td></tr>
</table>

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, *Discrete Mathematics and Its Applications*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2019.
7. F. Harary, *Graph Theory*. Reading, MA, USA: Addison-Wesley Publishing Company, 1969.

References:

1. S. Epp, *Discrete Mathematics with Applications*, 5th ed. Boston, MA, USA: Cengage Learning, 2019.
2. K. A. Ross and C. R. B. Wright, *Discrete Mathematics*, 6th ed. New Delhi, India: Pearson Education, 2018.
3. L. Mott, A. Kandel, and T. P. Baker, *Discrete Mathematics for Computer Scientists and Mathematicians*, 3rd ed. New Delhi, India: PHI Learning, 2016.

E-resources/ Web links:

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. — Level 2 – Understand CO2: Apply CSS and advanced styling techniques to design responsive and interactive web pages. — Level 3 – Apply CO3: Develop dynamic web applications using JavaScript, DOM manipulation, and event handling techniques. — Level 3 – Apply		

	CO4: Build server-side web applications using PHP and database connectivity concepts.	Level 3 – Apply
	CO5: Apply web technologies to develop secure and intelligent web applications using AI concepts.	Level 3 – 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. — Level 2 – Understand CO2: Apply CSS and advanced styling techniques to design responsive and interactive web pages. — Level 3 – Apply CO3: Develop dynamic web applications using JavaScript, DOM manipulation, and event handling techniques. — Level 3 – Apply CO4: Build server-side web applications using PHP and database connectivity concepts. — Level 3 – Apply CO5: Apply web technologies to develop secure and intelligent web applications using AI concepts. — Level 3 – 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: CSS2260	Course Name: Database Management Systems Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			
Anti-requisites	NIL		
Course Description	This course introduces the foundational principles of database management systems, including data models, schemas, and architecture. This course provides a solid foundation on the relational model of data and the use of relational algebra. It develops skills in SQL for data definition, manipulation, and control, enabling students to construct and execute complex queries. The course also introduces the concept of object oriented and object relational databases and modern database technologies like NoSQL. The also course allows the students to gain insights into data storage structures and indexing strategies for optimizing query performance.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain Employability through Problem Solving Methodologies.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Describe the fundamental elements of relational database management systems. [Level 2 - Understand] **CO2**: Examine databases using SQL query processing and Optimization. [Level 3 – Apply] **CO3:** Design simple database systems applying the normalization constraints and demonstrate the database transaction processing, recovery, and security. [Level 3 – Apply] **CO4:** Interpret the concept of advanced databases and its applications. [Level 3 – Apply]		
Course Content:			
Module 1	Introduction to Database Modelling and Relational Algebra	Assessment: Assignment (Problem Solving)	10 Sessions
Topics: Introduction to Database: Schema, Instance, 3-shema architecture, physical and logical data independence, Data isolation problem in traditional file system, advantages of database over traditional file systems. Entity Relationship (ER) Model, ER Model to Relational Model, Examples on ER model. Relational Algebra with selection, projection, rename, set operations, Cartesian product, joins (inner and outer joins), and division operator. Examples on Relational Algebra Operations.			

Module 2	Fundamentals of SQL and Query Optimization	Assessment : Assignment (Programming)	11 Sessions

Topics:

SQL Database Querying, DDL, DML, Constraints, Operators, Set Operators, Aggregate Functions, Joins, Views, Procedures, Functions and Triggers.

Database programming issues and techniques: Embedded SQL, Dynamic SQL; SQL / PSM and NoSQL.

Query Optimization: Purpose, transformation of relational expressions, estimating cost and statistics of expression, choosing evaluation plans, linear and bushy plans, dynamic programming algorithms.

Module 3	Relational Database Design & Transaction Management	Assessment: Assignment (Problem Solving)	12 Sessions

Topics:

Relational database design: Problems in schema design, redundancy and anomalies, Normal Forms based on Primary Keys-(1NF,2NF, 3NF), Boyce-Codd Normal Form, Multi valued Dependency (Fourth Normal Form), Join Dependencies (Fifth Normal Form), lossy and lossless decompositions, Database De-normalization.

Transaction Management: The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; The write-ahead log protocol; Check pointing; Recovering from a System Crash; Media Recovery; Other approaches and interaction with concurrency control.

Module 4	Advanced DBMS Topics	Assessment : Assignment (Case Study)	12 Sessions

Topics:

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

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

Targeted Applications & Tools:

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

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

1. Problem Solving: Constructing ER-Diagrams for a given real time requirements, Normalizing the databases, querying the databases using relational algebra.

2. Programming: Implementation of any given scenario using MySQL.

Skill-Building Activities through Experiential Learning:

Text Book

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.
E-Resources/Web Resources

Course Code: CSS2261	Course Name: Database Management Systems Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites			
Anti-requisites	NIL		
Course Description	The Database Management Systems (DBMS) Laboratory is designed to provide students with hands-on experience in database design, implementation, and management using SQL and database management tools such as MySQL. The lab complements theoretical concepts learned in database courses by allowing students to practice database creation, querying, and optimization techniques. The DBMS Lab enables students to develop industry-relevant skills in database management, preparing them for careers in software development, data engineering, and database administration.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain Employability through Problem Solving Methodologies.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Demonstrate the database concepts, practice, and SQL queries. [Level 3 – Apply] **CO2:** Design and implement database schemas while applying normalization techniques to optimize structure. [Level 3 – Apply] **CO3:** Develop and implement stored procedures, triggers, and views for automation and efficiency. [Level 3 – Apply] **CO4:** Design and build database applications for real world problems. [Level 3 – Apply]		

Course Content:

List of Laboratory Tasks:

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

Labsheet-1 [3 Practical Sessions]

Experiment No 1: [1 Session]

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

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

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

Experiment No. 2: [2 Sessions]

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

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

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

Labsheet-2 [3 Practical Sessions]

Experiment No. 3: [1 Session]

3. mplement complex queries in SQL.

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

Experiment No. 4: [2 Session]

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

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

scenario. [Airline Database]

Labsheet-3 [2 Practical Sessions]

Experiment No. 5: [2 sessions]

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

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

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

Labsheet-4 [2 Practical Sessions]

Experiment No. 6: [2 Sessions]

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

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

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

Labsheet-5 [2 Practical Sessions]

Experiment No. 7: [2 Sessions]

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

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

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

Labsheet-6 [4 Practical Sessions]

Experiment No. 8: [2 Sessions]

8. To implement the concept of forms and reports.

Level 1: Implement the concept of forms and reports.

Level 2: Examine the schema relationship.

Experiment No. 9: [2 Sessions]

9. Create the database using the given schema. (Flight Management)

Level 1: Implement a relational database based on the provided schema for the Flight Management system, including the creation of tables, relationships, and constraints.

Level 2: Demonstrate schema relationships by defining primary and foreign keys to ensure data integrity within the Flight Management database.

Labsheet-7 [4 Practical Sessions]

Experiment No. 10: [2 Sessions]

10. Create the database using the given schema. (Company database)

Level 1: Implement the database schema by defining tables, relationships, and constraints according to the given Company Database schema.

Level 2: Demonstrate the schema's relationships and data integrity by creating and linking tables as per the specified requirements.

Experiment No. 11: [2 Sessions]

11. Create the database using the given schema. (Student Library)

Level 1: Implement forms and reports based on the provided Student Library database schema, ensuring effective data entry and reporting mechanisms.

Level 2: Demonstrate the schema relationships within the Student Library database, demonstrating how these relationships influence the creation and functionality of forms and reports.

Labsheet-8 [1 Sessions]

12. Design a mini project based on the databases such as Inventory Management System, University Management System, Hospital Management System, etc.

Level 1: Implement the real time database.

Level 2: Analyze the working of database in real time.

Targeted Applications & Tools:
Skill-Building Activities through Experiential Learning:
Text Book **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.
E-Resources/Web Resources

Course Code: CSS2262	**Course Title: Analysis of Algorithms** **Type of Course: Theory Course (PCC)**	**L- T-P- C**	**3-1-0-4**
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	Nil		
Course Description	This course introduces techniques for the design and analysis of efficient algorithms and methods of 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:		

CO1: Explain the efficiency of algorithms in terms of time and space complexity.	Understand
CO2: Apply divide-and-conquer and decrease-and-conquer techniques to solve searching and sorting problems.	Apply
CO3: Apply dynamic programming and greedy techniques to solve computational problems.	Apply
CO4: Apply time–space trade-off techniques and analyze the limitations of algorithms for problem solving.	Apply
CO5: Analyze problems using backtracking, 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-Rhy12fMwZOf

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

Course Code:	Course Title: Analysis of Algorithms Lab		
CSS2263	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 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: CO1: Compute the efficiency of algorithms in terms of time and space complexity. — Level 3 – Apply CO2: Apply divide-and-conquer techniques to solve searching and sorting problems. — Level 3 – Apply CO3: Apply dynamic programming techniques to solve computational problems. — Level 3 – Apply CO4: Apply greedy techniques for solving optimization problems. — Level 3 – Apply CO5: Analyze problems using backtracking, branch-and-bound, and AI-based optimization techniques to develop optimal solutions. — Level 4 – 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 1and 3 Pearson.

Web Based Resources and E-books:

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

W2. Coursera: Analysis of Algorithms by Princeton University

W3. Algorithms Specialization in Coursera by Stanford University(Group of 4 courses).

W4. Algorithms Coding Contest Links maintained by Prof Gerth Stølting Brodal of Aarhus University

Topics relevant to "EMPLOYABILITY SKILLS": The lab experiments and assessments enable the student to acquire Skill Development through Experiential Learning techniques

Course Code: 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. . [Level 2 – Understand] **CO2:** Apply basic prompt engineering techniques for generating effective AI responses. [Level 3 – Apply] **CO3:** Evaluate and compare AI tools based on suitability, functionality, ethical considerations, privacy, and security requirements.. [Level 4 – Analyze] **CO4:** Understand the social, ethical, and economic impacts of AI technologies and their influence on society, industries, and employment. [Level 2 – Understand] **CO5:** Analyze advanced AI technologies and their real-world applications through case studies and practical scenarios. [Level 4 – Analyze]

Course Content:

Module 1	**Introduction to Artificial Intelligence**	**Assignment (Fundamentals and Applications of 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**	**Practical Assignment (AI Tools and Prompt Engineering 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**	**Quiz + Assignment (Evaluation and Ethical Use 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**	**Assignment (Social, Ethical, and Economic Impact Analysis 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 case study	Mini Project / Case Study (AI Technologies and Real-World Applications)	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:

1. Implementation of AI-based content generation and chatbot interaction using prompt engineering techniques.
   Tools: ChatGPT, Google Gemini, Hugging Face
2. Implementation of image generation and creative AI applications using generative AI tools.
   Tools: Canva AI, DALL·E, Adobe Firefly
3. Implementation of predictive analytics and machine learning models for simple real-world datasets.
   Tools: Google Colab, TensorFlow, Kaggle
4. Implementation of NLP-based applications such as sentiment analysis, text summarization, and language translation.
   Tools: NLTK, spaCy, Hugging Face Transformers
5. 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

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

Textbook(s):

1. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson Education, 2022.
2. Elaine Rich and Kevin Knight, *Artificial Intelligence*, 3rd Edition, McGraw Hill Education, 2009.
3. Tom Taulli, *Artificial Intelligence Basics: A Non-Technical Introduction*, 1st Edition, Apress, 2019.

References:

1. Ethem Alpaydin, *Introduction to Machine Learning*, 4th Edition, MIT Press, 2020.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, 1st Edition, MIT Press, 2016.
3. Melanie Mitchell, *Artificial Intelligence: A Guide for Thinking Humans*, 1st Edition, Farrar, Straus and Giroux, 2019.
4. Pedro Domingos, *The Master Algorithm*, 1st Edition, Basic Books, 2015.
5. Kai-Fu Lee, *AI Superpowers*, 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 (ESC)	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.. [Level 3 – Apply] **CO2:** Implement AI-based applications for text generation, image generation, and chatbot development. . [Level 3 – Apply] **CO3:** Apply Natural Language Processing and basic Machine Learning techniques for solving simple real-world problems.. [Level 3 – Apply] **CO4:** Analyze datasets and AI model outputs using preprocessing, visualization, and evaluation techniques. [Level 4 – Analyze] **CO5:** Analyze ethical issues, bias, and the impact of AI systems in practical applications [Level 4 – 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-Jupyer 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:** 1. **Understand the basic concepts of finance and financial markets and organizations.** 2. **Apply and interpret financial information for business decision making.** 3. **Identify various heads of income and deduction under Income Tax Act, 1961.**					
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):

1. Dr. Vinod K. Singhania & Dr. Monica Singhania. (Latest Assessment Year Edition). *Students' Guide to Income Tax including GST*. Taxmann Publications.
2. Pandey, I. M. (2025). *Financial Management*. Vikas Publishing House.

Reference Book (s):

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

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

1. https://presidencyuniversity.linways.com
2. https://onlinecourses.nptel.ac.in/noc24_ec01/preview
3. https://www.incometax.gov.in/iec/foportal/

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

Course 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 Out Comes	**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**	Assignment	**16 Sessions**
Topics: Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss			
Module 2	Quantitative Ability 2	Assignment	**14 Sessions**
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 Assignment			

Text Book:

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

References:

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

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: CO1: Explain fundamental concepts of automata, formal languages, and finite state machines. — Level 2-Understand CO2: Design and analyze deterministic and nondeterministic finite automata and regular expressions for language recognition. — Level 3-Apply CO3: Construct and evaluate context-free grammar and pushdown automata, including ambiguity removal and normal forms. — Level 3-Apply CO4: Design Turing machines and interpret their role in solving computational problems and language recognition. — Level 3-Apply CO5: Explain the applications of Theory of Computation concepts in real-world domains such as AI/ML, NLP, cybersecurity, compilers, and robotics. — Level 2-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 Accepter, 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: CSS2502	**Course Title: Cryptography and Network Security** **Type of Course: Program Core & Theory only**	**L- T-P- C**	3	0	3
Version No.	1				
Course Pre-requisites	CSE1507 Data Communication and Computer Networks				
Anti-requisites	**NIL**				
Course Description	The Course covers the principles and practice of cryptography and network security, focusing in particular on the security aspects of the web and Internet. The cryptographic tools such as shared key encryption, public key encryption, key exchange, and digital signature are explored. The use and utilization of the internet protocols and applications such as SSL/ TLS, IPSEC, Kerberos, PGP, and S/ MIME, SET are reviewed. System security issues such as viruses, intrusion and firewalls are also explored.				
Course Objective	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: CO1: Extrapolate cryptography fundamentals and security architecture [Level 2 - Understand] CO2: Classify classical encryption techniques [Level 3 - Apply] CO3: Examine symmetric key cryptography algorithms [Level 3 - Apply] CO4: Determine number theory concepts in cryptography [Level 3 - Apply] CO5: Demonstrate public key cryptography and authentication methods [Level 3 - Apply] CO6: Compare network security protocols and applications [Level 4 - Analyze]				
Course Content:					
Module 1	**Introduction to Cryptography and types of Ciphers**	**Problem Solving**	Identify the basic concept of Cryptography	10 Sessions	

Topics: Introduction to Cryptography, Model of Network Security, OSI Security architecture, Security Attacks: Active attacks, Passive attacks, Services: Authentication, Access Control, Data Confidentiality, Data Integrity, Nonrepudiation, Substitution techniques: Caesar cipher, Monoalphabetic cipher, Polyalphabetic cipher, Play-fair and Hill Cipher, Transposition techniques: Rail fence, Basics of Steganography.				
Module 2	**Symmetric Key Cryptography and Number Theory**	**Problem Solving & Participative Learning**	Apply the concepts of cryptographic algorithms	13 Sessions
Topics: Introduction to Block Cipher and Stream Cipher, Feistel Structure, Symmetric Encryption Algorithms: Data Encryption Standard, Introduction to Galois Field, Advanced Encryption Standard, Modular Arithmetic, Prime numbers, Fermat's little theorem, Primality testing: Miller-Rabin algorithm, Euclidean and Extended Euclidean Algorithm, Euler Totient Function, Chinese Remainder Theorem				

Course Code: CSS2269	Course Name: Operating Systems Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	CAI2501		
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 the course the students shall be able to:** **CO1:** Explain the fundamental concepts of operating systems and relevant case studies. [Level 2 - Understand] **CO2**: Apply various CPU scheduling algorithms for process management. [Level 3 – Apply] **CO3:** Apply synchronization techniques to handle concurrency problems. [Level 3 – Apply] **CO4:** Apply deadlock detection and recovery methods in operating systems. [Level 3 – Apply] **CO5:** Explain various memory management techniques used in operating systems. [Level 3 -Apply]		

Course Content:

Module 1	Introduction to Operating Systems	Assessment: Assignment (Programming using Keras/Sklearn)	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	Assessment : Quiz + Assignment (Sklearn/Statsmodels)	12 Sessions

Topics:

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

Module 3	**Process Synchronization and Deadlocks**	Assessment : Assignment (Programming using Keras/Sklearn)	12 Sessions

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	Assessment : Programming Assignment (Keras/TensorFlow)	12 Sessions

Topics:

Introduction to Memory Management, Basic hardware-Base and Limit Registers, Memory Management Unit(MMU), Dynamic loading and linking, Swapping, Contiguous and Non-Contiguous Memory Allocation, Segmentation, Paging - Structure of the Page Table – Virtual Memory and Demand Paging – Page Faults and Page Replacement Algorithms, Copy-on-write, Allocation of Frames, Thrashing

Introduction to File system management: File System Interface (access methods, directory structures), File system implementation.

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 Applications & Tools:

Use cases	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.
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.

Skill-Building Activities through Experiential Learning:

- Comparative Analysis of Ensemble Techniques
  Learners experiment with ensemble models like Random Forest, AdaBoost, and Gradient Boosting, conducting comparative studies on model variance, bias, and robustness. Through group assignments, students simulate real-world decision-making scenarios where ensemble strategies outperform individual models.

- Neural Network Construction Using Keras
  Using the Keras API, students build and train basic feedforward neural networks. They gain practical experience in selecting activation functions, configuring loss functions, tuning hyperparameters (e.g., learning rate, epochs), and applying backpropagation using optimizers like SGD and Adam.

- Unsupervised Learning for Pattern Discovery
  Through clustering algorithms such as K-Means, Hierarchical Clustering, DBSCAN, and Gaussian Mixture Models (GMM), learners segment real-world datasets and visualize clusters. They apply evaluation methods like the Elbow Method and Silhouette Score to validate clustering quality.

Text Book

T1: Silberschatz A, Galvin P B and Gagne G , "Silberschatz's Operating System Concepts", Paperback, Global Edition Wiley, 2019

References

R1: Silberschatz A, Galvin P B and Gagne G, "Operating System Concepts", 10th edition Wiley, 2018.

R2: William Stallings, "Operating Systems",Ninth Edition, By Pearson Paperback ,1 March 2018.
R3: Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, " Cracking the Operating System skills", Dreamtech, paperback, 2020.
R4: Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , "Operating Systems: Three Easy Pieces, Amazon digital Services", September 2018.

E-Resources/Web Resources

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: CSS2270	Course Name: **Operating Systems 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	CAI2500		
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 Objective	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 the course the students shall be able to:** **CO1:** Demonstrate system-level programming using system calls and operating system structures. [Level 3 – Apply] **CO2:** Simulate process scheduling and multithreading techniques. [Level 3 – Apply] **CO3:** Apply synchronization techniques using semaphores and shared memory to handle concurrency problems. [Level 3 – Apply] **CO4:** Demonstrate memory management and file system concepts using simulation or scripting tools. [Level 3 – Apply] **CO5:** Analyze the performance of operating system algorithms related to scheduling, synchronization and memory management. . [Level 4 – Analyze]

Course Content:

List of Laboratory Tasks:

- **Experiment 1: Introduction to OS**
  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.

  OS Structure

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

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

- **Experiment 2: System Parameters**

  Linkers & Loaders

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

  Level 2: Analyze object files and executable generation.

  System Utilities

  Level 1: Write simple shell commands using system calls.

  Level 2: Implement a mini shell program.

- **Experiment 3: Process Concept**

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

  Level 2: Demonstrate parent-child communication.

  Process Operations
  Level 1: Implement process states and transitions.

  Level 2: Simulate process control block (PCB).

- **Experiment 4: Inter Process Communication**
  Pipes:

  Level 1: Implement communication using pipes.

  Level 2: Implement bidirectional communication using pipes.

Sockets/RPC

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

Level 2: Simulate RPC communication.

- **Experiment 5: Threads**
  Thread Creation

  Level 1: Create threads using pthread library.

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

  Thread Issues

  Level 1: Demonstrate thread synchronization issues.

  Level 2: Solve race condition problem.

- **Experiment 6: CPU Scheduling**
  FCFS & SJF

  Level 1: Implement FCFS scheduling.

  Level 2: Implement SJF and compare results.

  Advanced Scheduling

  Level 1: Implement Round Robin scheduling.

  Level 2: Implement Priority scheduling and analyze performance.

- **Experiment 7: Synchronization**

  Critical Section

  Level 1: Implement Peterson's solution.

  Level 2: Analyze critical section problem.

  Semaphores

  Level 1: Implement semaphore operations

  Level 2: Solve producer-consumer problem using semaphores.

- **Experiment 8: Classical Problems**
  Reader-Writer

  Level 1: Implement Reader-Writer problem.

  Level 2: Analyze starvation issues.

  Dining Philosophers

  Level 1: Implement Dining Philosophers problem.

  Level 2: Avoid deadlock using strategies.

- **Experiment 9: Dead Locks**
  Deadlock Detection

  Level 1: Implement Resource Allocation Graph.

  Level 2: Detect deadlock using algorithm.

  Deadlock Avoidance

Level 1: Implement Banker's algorithm.

Level 2: Analyze safe and unsafe states.

- **Experiment 10: Memory Management Basics**

  Memory Allocation

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

  Level 2: Compare memory utilization.

  Paging

  Level 1: Simulate paging mechanism.

  Level 2: Analyze page table structure.

- **Experiment 11: Virtual Memory**

  Page Replacement

  Level 1: Implement FIFO page replacement.

  Level 2: Implement LRU and Optimal algorithms.

  Page Faults

  Level 1: Calculate page faults.

  Level 2: Analyze thrashing condition.

- **Experiment 12: Advanced Memory Management**

  Segmentation

  Level 1: Implement segmentation.

  Level 2: Compare segmentation vs paging.

  Demand Paging

  Level 1: Simulate demand paging.

  Level 2: Analyze copy-on-write mechanism.

- **Experiment 13: File System**

  File Operations

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

  Level 2: Simulate directory structure.

  File Allocation

  Level 1: Implement file allocation methods.

  Level 2: Compare allocation techniques.

- **Experiment 14: File System Implementation**

  File System Design

  Level 1: Design simple file system.

  Level 2: Analyze file system performance.

Access Methods

Level 1: Implement sequential access.

Level 2: Implement indexed access.

- **Experiment 15: Integration & Simulation**

  OS Simulation

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

  Level 2: Integrate scheduling and memory management.

  Mini Project

  Level 1: Implement OS concept-based project.

  Level 2: Analyze performance and optimization.

Project Work/Assignments

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 Applications & Tools:

Use cases	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.
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.

Skill-Building Activities through Experiential Learning:

- **Data Preprocessing with Real-World Datasets**
  Learners perform hands-on preprocessing of real-world datasets (e.g., UCI, Kaggle) by handling missing values, scaling features, and encoding categorical variables. This foundational step helps them understand the impact of clean data on model performance.

- **Exploratory Data Analysis and Visualization**
  Using libraries such as Matplotlib and Seaborn, students conduct exploratory data analysis through plots like scatter plots, heatmaps, and pair plots. Visualizations are used to identify patterns, correlations, and anomalies before model building.

- **Model Building with Supervised Learning Algorithms**
  Students implement regression and classification algorithms—Linear Regression, Logistic Regression, Naïve Bayes, and SVM—on structured datasets. They analyze model predictions using decision boundaries, ROC curves, and learning curves.

Text Book
T1: Silberschatz A, Galvin P B and Gagne G , “Silberschatz's Operating System Concepts”, Paperback, Global Edition Wiley, 2019
References
R1: Silberschatz A, Galvin P B and Gagne G, “Operating System Concepts”, 10th edition Wiley, 2018. **R2:** William Stallings, “Operating Systems”,Ninth Edition, By Pearson Paperback ,1 March 2018. **R3:** Sundaram RMD, Shriram K V, Abhishek S N, B Chella Prabha, “ Cracking the Operating System skills”, Dreamtech, paperback, 2020. **R4:** Remzi H. Arpaci-Dusseau Andrea C. Arpaci-dusseau , “Operating Systems: Three Easy Pieces, Amazon digital Services”, September 2018.
E-Resources/Web Resources
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: CBC2500	Course Name: Smart Contract and Solidity Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	2.0		
NCrF Level-			
Course Pre-requisites	Basics of Mathematics and any Programming Language		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state. Solidity is a curly- bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript. The Ethereum Virtual Machine (EVM) and assembly (low level language), events and logging blockchain emissions, send vs transfer methods, scoping and more		
Course Objective	• To introduce the fundamental concepts of **blockchain technology, Ethereum architecture, and smart contracts**. • To develop the ability to **design and implement smart contracts using Solidity** programming **language**. • To understand the **working of Ethereum Virtual Machine (EVM) and contract execution mechanisms.** • To enable students to **develop, test, and deploy smart contracts using Remix IDE** and **related development tools**.		

Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Understand the fundamentals of computational elements of blockchain technology **(Level 2: Understand)** **CO2:** Implement user-defined operations of arbitrary complexity that are not possible through plain cryptocurrency protocols **(Level 3: Apply)** **CO3:** Exhibit best practices for designing solutions with smart contracts using Solidity and Remix IDE **(Level 3: Apply)**

Course Content:

Module 1	Introduction to Smart Contract	Assessment: Fundamentals of Smart Contract and Solidity	12 Sessions

Topics:

A Simple Smart Contract
Blockchain Basics
The Ethereum Virtual Machine
Versioning, Remix
npm / Node.js
Docker
Binary Packages
Building from Source
CMake options.

Module 2	Solidity in Depth	Assessment : Case Study	12 Sessions

Topics:

Layout of a Solidity Source File
Structure of a Contract
Types, Units and Globally Available Variables
Expressions and Control Structures,
Contracts
Solidity Assembly
Miscellaneous
Solidity v0.5.0 Breaking Changes

Module 3	Contract Metadata & Contract ABI Specification	Assessment : Implementing Applications	14 Sessions

Topics:

Encoding of the Metadata Hash in the Bytecode
Usage for Automatic Interface Generation and NatSpec
Usage for Source Code Verification
Basic Design, Function Selector
Argument Encoding
Types
Design Criteria for the Encoding

Formal Specification of the Encoding

Function Selector and Argument Encoding

Examples

Use of Dynamic Types

Events

JSON

Strict Encoding Mode

Non- standard Packed Mode

Targeted Applications & Tools:

Remix IDE	Used for writing, compiling, testing, and deploying smart contracts.
Solidity Compiler	Used for compiling Solidity smart contract code into executable bytecode.
MetaMask	Used for wallet management and interaction with Ethereum-based decentralized applications.

Skill-Building Activities through Experiential Learning:

- Design and implement smart contracts using Solidity.
- Test and deploy Ethereum smart contracts using Remix IDE.
- Analyze contract execution in Ethereum Virtual Machine (EVM).
- Apply best practices for secure smart contract development

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 et al., *Network Security: Private Communication in a Public World*, Prentice Hall, 2nd Edition, 2002

R4: Christof Paar and Jan Pelzl, *Understanding Cryptography*, Springer, 2010

E-Resources/Web Resources

1. https://cryptography.io

2. https://nvlpubs.nist.gov

3. https://www.owasp.org

4. https://www.tutorialspoint.com/cryptography/index.htm

5. https://www.coursera.org/learn/crypto

Course Code: CBC2501	Course Name: Smart Contract and Solidity Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	2.0		
NCrF Level-			
Course Pre-requisites	Basics of Mathematics and any Programming Language		

Co-requisites	NIL
Anti-requisites	NIL
Course Description	Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state. Solidity is a curly- bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript. The Ethereum Virtual Machine (EVM) and assembly (low level language), events and logging blockchain emissions, send vs transfer methods, scoping and more
Course Objective	• To introduce the fundamental concepts of **blockchain technology, Ethereum architecture, and smart contracts**. • To develop the ability to **design and implement smart contracts using Solidity programming language**. • To understand the **working of Ethereum Virtual Machine (EVM) and contract execution mechanisms**. • To enable students to **develop, test, and deploy smart contracts using Remix IDE and related development tools**.
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO 1:** Understand the fundamentals of computational Element of the Blockchain Technology (Level 2: Understand) **CO 2:** Implement user-defined operations of arbitrary complexity that are not possible through plain cryptocurrency protocols (Level 3: Apply) **CO 3:** Exhibit best practices for designing solutions with smart contracts using Solidity and Remix IDE (Level 3: Apply)

Course Content:

List of Lab Experiments:

Week	Lab Experiment Title	Tool/Technology Used
Week 1	Introduction to cryptographic tools and basic encryption techniques	CrypTool / Python (PyCryptodome)
Week 2	Implementation of Caesar and Monoalphabetic ciphers	Python / CrypTool
Week 3	Playfair and Hill Cipher encryption and decryption	Python / CrypTool
Week 4	DES algorithm implementation and file encryption	OpenSSL / Python (PyCryptodome)
Week 5	AES encryption in ECB and CBC modes	OpenSSL / Python (cryptography)
Week 6	RSA key generation, encryption, and decryption	OpenSSL / Python
Week 7	Diffie-Hellman key exchange simulation	Python
Week 8	Hashing and message digest using SHA, MD5	Python (hashlib) / OpenSSL
Week 9	Digital signature creation and verification	GnuPG / Python (cryptography)

Week 10	Setup secure communication using SSL/TLS	OpenSSL / Wireshark
Week 11	Steganography and cryptanalysis demo	Steghide / CrypTool
Week 12	Password hashing and brute force attack demo	John the Ripper / Hashcat
Week 13	Secure Email communication using GnuPG (PGP Simulation)	GnuPG
Week 14	Network traffic capture and analysis for secure protocols	Wireshark
Week 15	Mini project: End-to-end secure message exchange with key management	Python + GPG + OpenSSL

Targeted Applications & Tools:

CrypTool	Used for learning, analyzing, and demonstrating cryptographic algorithms and security techniques.
Python	Used for implementing cryptographic algorithms, security analysis, and automation of cybersecurity tasks.
OpenSSL	Used for encryption, SSL/TLS certificate generation, secure communication, and cryptographic operations.
GnuPG	Used for secure encryption, decryption, digital signatures, and secure file/email communicatio
Wireshark	Used for capturing, monitoring, and analyzing network traffic to identify security issues and communication patterns.

Skill-Building Activities through Experiential Learning:

Develop the ability to design, implement, test, and deploy decentralized applications using Solidity smart contracts and Ethereum development tools.

Text Book

T1: Andreas M. Antonopoulos & Gavin Wood, Mastering Ethereum, O'Reilly Media, 2018.

T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 3rd Edition, 2020.

References

R1: Narayan Prusty, Building Blockchain Projects, Packt Publishing, 2018.

R2: Arshdeep Bahga & Vijay Madisetti, Blockchain Applications: A Hands-On Approach, VPT, 2017.

R3: Melanie Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media, 2015.

R4: Daniel Drescher, Blockchain Basics, Apress, 2017.

E-Resources/Web Resources

1. https://ethereum.org/en/developers/
2. https://soliditylang.org
3. https://remix.ethereum.org
4. https://docs.hyperledger.org
5. https://developer.mozilla.org/en-US/docs/Web/Web3

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

CBC2502</td><td colspan="2">Course Name: Distributed Ledger Technology

Type of Course: Theory Course</td><td>L- T-P- C

3-0-0-3</td></tr>
<tr><td>Version No.</td><td colspan="3">2.0</td></tr>
<tr><td>NCrF Level-</td><td colspan="3"></td></tr>
<tr><td>Course Pre-requisites</td><td colspan="3">NIL</td></tr>
<tr><td>Co-requisites</td><td colspan="3">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="3">NIL</td></tr>
<tr><td>Course Description</td><td colspan="3">This course introduces the principles, architectures, and applications of Distributed Ledger Technology (DLT), with a special focus on blockchain systems. Students will explore consensus mechanisms, smart contracts, permissioned and permissionless ledgers, and enterprise blockchain frameworks like Hyperledger Fabric and Corda. Real-world use cases in supply chain, finance, and digital identity are emphasized.</td></tr>
<tr><td>Course Objective</td><td colspan="3">• Understand the fundamentals and types of distributed ledger technologies.
• Explore blockchain architecture and consensus mechanisms.
• Analyze security, scalability, and privacy in decentralized systems.
• Apply DLT in enterprise and public applications through smart contracts and platforms.</td></tr>
<tr><td>Course Outcomes</td><td colspan="3">On successful completion of the course the students shall be able to:

CO1: Explain the architecture and working principles of DLT and blockchain.

(Level 2: Understand)

CO2: Compare consensus mechanisms and evaluate DLT use cases and limitations.

(Level 4: Analyze)

CO3: Implement smart contracts using Ethereum or Hyperledger. (Level 3: Apply)

CO4: Develop secure and scalable DLT applications for real-world problems.

(Level 3: Apply)</td></tr>
<tr><td colspan="4">Course Content:</td></tr>
<tr><td>Module 1</td><td>Introduction to DLT and Blockchain</td><td>Assessment: Distributed Ledger System</td><td>11 Sessions</td></tr>
<tr><td colspan="4">Topics:

Distributed Ledger vs Centralized Systems
Blockchain overvie
Key features of DLT (immutability, transparency, decentralization)
Cryptographic primitives
Merkle trees
Hash functions
Public vs Private blockchains,</td></tr>
</table>

Peer-to-peer networking			
Module 2	Blockchain Architecture and Consensus Mechanisms	Assessment : Case Study	12 Sessions
Topics: Blockchain structure Blocks and transactions Consensus algorithms: Proof of Work, Proof of Stake, Delegated PoS Byzantine Fault Tolerance, Forks, Block finality Challenges in consensus Network latency, Scalability			
Module 3	Smart Contracts and DLT Frameworks	Assessment : Implementing Applications	11 Sessions
Topics: Introduction to smart contracts Ethereum architecture Solidity basics Hyperledger Fabric architecture and components Chain code in Go/Node.js Smart contract lifecycle Oracles and off-chain data access			
Module 4	Applications, Security, and Industry Use Cases	Assessment : DTL in supply chain	11 Sessions
Topics: DLT in supply chain digital identity, healthcare, fintech Security issues in blockchain: 51% attack, Sybil attacks, Double spending Data privacy and GDPR Case studies: IBM Food Trust, TradeLens, Digital Voting			

Targeted Applications & Tools:

Ethereum Blockchain Platform	Used for developing and deploying smart contracts, decentralized applications (DApps), cryptocurrencies, NFTs, and decentralized finance (DeFi) applications on a public blockchain network.
Hyperledger Fabric Framework	Used for building secure, permissioned enterprise blockchain applications with private transactions for domains such as supply chain, healthcare, banking, and business process automation.

Skill-Building Activities through Experiential Learning:

- **Smart Contract Development using Ethereum**
  Learners design and deploy smart contracts using Solidity on the Ethereum blockchain. Through hands-on coding exercises, students understand decentralized transaction execution, gas mechanisms, and blockchain-based automation.
- **Decentralized Application (DApp) Development**
  Students build simple DApps integrating frontend interfaces with Ethereum smart contracts using tools such as MetaMask and Web3 libraries. These activities provide practical exposure to blockchain-enabled application architecture.
- **Blockchain Network Setup using Hyperledger Fabric**
  Learners configure peers, orderers, channels, and certificate authorities in Hyperledger Fabric.

	Practical exercises help students understand permissioned blockchain architecture and enterprise-level distributed ledger management.
Text Book **T1:** Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020 **T2:** Melanie Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly Media, 2015	
References **R1:** Arshdeep Bahga and Vijay Madisetti, *Blockchain Applications: A Hands-On Approach*, VPT, 2017 **R2:** Andreas M. Antonopoulos and Gavin Wood, *Mastering Ethereum*, O'Reilly Media, 2018 **R3:** Narayan Prusty, *Building Blockchain Projects*, Packt Publishing, 2018 **R4:** Nitin Gaur et al., *Hands-On Blockchain for Python Developers*, Packt, 20	
E-Resources/Web Resources **1.** https://ethereum.org/en/developers/ **2.** https://hyperledger.org/use/fabric **3.** https://www.ibm.com/blockchain **4.** https://www.blockchain-council.org/ **5.** https://developer.algorand.org/	

Course Code: CBC2503	Course Name: Distributed Ledger Technology Laboratory Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	2.0		
NCrF Level-			
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This laboratory course provides hands-on experience in working with blockchain-based distributed ledger platforms such as Ethereum and Hyperledger. Students will learn to develop, deploy, and test smart contracts, create blockchain networks, and simulate real-world DLT use cases using open-source tools.		
Course Objective	• To provide practical understanding of blockchain platforms like Ethereum and Hyperledger. • To develop and test smart contracts using Solidity and Chain code. • To simulate peer-to-peer blockchain networks and transaction flows. • To integrate DLT with applications for real-world solutions.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Understand the setup and configuration of blockchain environments. (Level 2: Understand) **CO2:** Apply skills to write, deploy, and test smart contracts on Ethereum. (Level3: Apply) **CO3:** Implement and manage blockchain networks using Hyperledger Fabric. (Level 3: Apply)		

	CO4: Apply integration and debugging techniques to test and deploy decentralized applications. (Level 3: Apply)

Course Content:

List of Laboratory Tasks:
List of Lab Experiments:

Week	Lab Experiment Title	Tool/Technology Used
Week 1	Installation and setup of blockchain development environment	Node.js, Ganache, MetaMask, Remix IDE
Week 2	Creating a local Ethereum blockchain using Ganache CLI	Ganache CLI
Week 3	Creating Ethereum wallets and sending Ether between accounts	MetaMask, Web3.js
Week 4	Writing and deploying a basic smart contract in Solidity	Remix IDE
Week 5	Solidity programming: control structures, data types, modifiers	Remix IDE
Week 6	Deploying smart contracts on Ganache and interacting with Web3.js	Ganache, Web3.js, Node.js
Week 7	Smart contract functions: Events, state variables, and mappings	Remix IDE
Week 8	Testing smart contracts using Truffle framework	Truffle, Mocha/Chai for test cases
Week 9	Setup of a Hyperledger Fabric blockchain network using Docker	Hyperledger Fabric, Docker
Week 10	Writing and deploying Chaincode (smart contracts) on Fabric	Go or Node.js
Week 11	Invoking transactions and querying the ledger in Fabric	Fabric CLI or SDK
Week 12	Developing a basic asset transfer DApp on Ethereum	Solidity, Web3.js, HTML/JS frontend
Week 13	Smart contract security: Detecting vulnerabilities (e.g., reentrancy)	MythX, Slither
Week 14	End-to-end DApp project integration and deployment	Truffle/Hardhat + Web3.js + frontend
Week 15	Mini Project Demo: Real-world use case using DLT	Students choose Ethereum or Fabric for implementation

Targeted Applications & Tools:

Ethereum Blockchain Platform	Used for developing and deploying **smart contracts, decentralized applications (DApps), cryptocurrencies, and decentralized finance (DeFi) applications** on a public blockchain network.
Hyperledger Fabric Framework	Used for building **secure, permissioned enterprise blockchain applications** for supply chain management, healthcare, banking, and business process automation.

Skill-Building Activities through Experiential Learning:

Develop the ability to design, implement, test, and deploy distributed ledger applications using blockchain platforms such as Ethereum and Hyperledger, enabling students to build secure

decentralized systems for real-world applications.
Text Book **T1:** Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020 **T2:** Melanie Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly Media, 2015
References **R1:** Arshdeep Bahga and Vijay Madisetti, *Blockchain Applications: A Hands-On Approach*, VPT, 2017 **R2:** Andreas M. Antonopoulos and Gavin Wood, *Mastering Ethereum*, O'Reilly Media, 2018 **R3:** Narayan Prusty, *Building Blockchain Projects*, Packt Publishing, 2018 **R4:** Nitin Gaur et al., *Hands-On Blockchain for Python Developers*, Packt, 2020
E-Resources/Web Resources **1.** https://ethereum.org/en/developers/ **2.** https://hyperledger.org/use/fabric **3.** https://www.ibm.com/blockchain **4.** https://www.blockchain-council.org/ **5.** https://developer.algorand.org/

Course Code: CSS7000	**Course Title: Internship** **Type of Course: PRW**	**L- T-P- C**	**0**	**0**	**0**	**2**
Version No.	1.0					
Course Pre-requisites	Knowledge and Skills related to all the courses studied in previous semesters.					
Anti-requisites	NIL					
Course Description	Students observe science and technology in action, develop an awareness of the 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 Title: 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 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.					
	CO2] **Apply** the concepts in problem solving (Bloom's taxonomy Level 3)					
	CO3] **Analyze** and structure the reasoning techniques and spatial visualization skills					
Course Content:						
Module 1	Logical Thinking	Assignment		**16 Sessions**		
	Topics:					
	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 Sessions**		
	Topics:					
	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 Type of Course: Theory Course	L- T-P- C	1-0-0-0
Version No.			
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamentals of Indian Constitution concepts and their relevance to 75+ Years of Republic of India (https://constitution75.com/) as well as #AzaadiKaAmrutMahotsav / Azadi Ka Amrit Mahotsav (https://amritmahotsav.nic.in). It is designed to equip students with the knowledge about the Constitution of India. This course aims to introduce the constitutional law of India to students from all walks of life and help them understand the constitutional principles as applied and understood in everyday life. The objective of making the Constitution of India, familiar to all students, and not only to law students, this course aims and objectifies legal understanding in the simplest of forms. This course is designed to cater to Constitutional Studies.		
Course Objective	The objective of the course is 'SKILL DEVELOPMENT' of the student by using 'PARTICIPATIVE LEARNING' techniques		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Describe the basic understanding of the Indian Constitution and the concepts and issues relevant to day-to-day life of the nation and to equip the Citizen with the zeal of capacity building. Recognizing and identify the values of the Constitution of India. [Level 2 - Understand] **CO2**:Enabling the Citizen-centric Awareness of Rights and Responsibilities of the State. [Level 2 – Understand] **CO3:** 3. Explain the role of the State actors in building India. [Level 2 – Understand] **CO4: 4.** Understanding the Gandhian vision over the power of the LSG (Local Self-Governance) [Level 2 – Understand]		
Course Content:			
Module 1	Understanding the Making of the Constitution: The Constituent Assembly & The Constitution of India	Assessment: Quiz	5 sessions

Topics:

Historical Context of Constituent Assembly - Compositions & Functions of Constituent Assembly

What is a Constitution? – Why have a Constitution? – Constitutional Change - Features of Indian Constitution – Preamble of Indian Constitution

Module 2	Citizen's Fundamental Rights and State's Responsibilities (Directive Principles)	Assessment	5 Sessions

Topics:

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	Assessment : Presentation	5 Sessions

Topics:

Executive: The President of India - Powers and Functions of President of India - Emergency Powers and the Position of the President

Legislature: Union Council of Ministers - Prime Minister - The Rajya Sabha - The Lok Sabha - Relation between the Lok Sabha & Rajya Sabha - Office of the Speaker – Important Parliamentary Committees

Judiciary: The Structure and Organization of the Judiciary & the High Court - The Supreme Court - Role of The Supreme Court - Judicial Activism in India - Basic Structure Doctrine & PIL

Module 4	Federalism & Decentralization		6 sessions

Topics:

What is Federalism? - Centre-State Legislative Relations - Centre-State Administrative Relations - Centre-State Financial Relations

The 5th & 6th Schedules - Municipality- (History of Indian Municipality, Organization & Functions) – Panchayat 1 (Idea of Panchayat, Organization and Powers of Panchayats in India)

Text Book

T1: Durga Das Basu --- Introduction to the Constitution of India, 23rd Edition (Gurgaon; LexisNexis, 2018**).**

T2: MP Jain's Constitutional Law of India, Lexis Nexis

References

R1: Himangshu Roy and M.P.Singh – Indian Political System, 4th Edition (Bengaluru; Pearson Education, 2018)

R2: Vidya Bhushan & Vishnoo Bhagwan--- Indian Administration (S. Chand, 2011)

E-Resources/Web Resources

1. https://openFullText.html?DP=https://legislative.gov.in/constitution-of-india/
2. https://openFullText.html?DP=https://nptel.ac.in/courses/109104074
3. https://openFullText.html?DP=https://www.indiacode.nic.in/

Course Code: CBC2504	Course Name: Blockchain Security and Performance Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	2.0		
NCrF Level-			
Course Pre-requisites	CBC1700		
Co-requisites	CBC2504		
Anti-requisites	NIL		
Course Description	This course delves into the security and performance aspects of blockchain systems. It covers consensus attacks, smart contract vulnerabilities, performance bottlenecks, and secure protocol design. Emphasis is laid on techniques to improve resilience, scalability, and privacy while preserving decentralization.		
Course Objective	• Understand blockchain-specific threats and security principles • Analyze performance metrics in blockchain platforms • Explore security techniques for smart contracts and consensus algorithms • Evaluate design strategies to enhance scalability and privacy		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Identify and describe security challenges in blockchain platforms (Level-1: Understand) **CO2:** Evaluate consensus attacks and performance issues in different blockchain frameworks (Level2: Analyze) **CO3:** Implement security-aware smart contracts and mitigation techniques (Level3: Apply) **CO4:** Analyze and optimize blockchain performance under diverse network loads (Level3: Apply)		
Course Content:			
Module 1	Blockchain Security Foundations	Assessment: Programming Assignment	11 Sessions
Topics: Security principles in distributed systems Blockchain threat models Sybil attacks Eclipse attacks 51% attack Cryptographic assumptions Consensus vulnerabilities Double-spending			
Module 2	Smart Contract Security	Assessment : Case Study	11 Sessions
Topics:			

Ethereum vulnerabilities

DAO attack analysis

Reentrancy attacks

Arithmetic overflow/underflow

Front-running

Best practices in secure smart contract design

Tools for smart contract auditing

Module 3	Blockchain Performance Metrics	Assessment : Experimental Assignment	11 Sessions

Topics:

Performance benchmarks: latency, throughput, block size, transaction rate

Bottlenecks in scalability

Network propagation delay

Chain growth and quality

Trade-offs in performance vs security

Module 4	Enhancing Blockchain Performance and Privacy	Assessment : Experimental Assignment	12 Sessions

Topics:

Layer-2 scaling (State Channels, Rollups)

Sharding DAG-based architectures

Zero-Knowledge Proofs

Mixers and Privacy coins

Secure Multi-party Computation

Formal verification techniques

Targeted Applications & Tools:

Ethereum Blockchain Platform	Smart contract security testing and decentralized application development.
Hyperledger Fabric	Enterprise blockchain security and performance evaluation.
Solidity Programming Language	Development of secure smart contracts.
Ganache Local Blockchain	Local blockchain testing and simulation.
Hardhat / Truffle Framework	Smart contract deployment, testing, and debugging.
OpenZeppelin Libraries	Secure smart contract development using reusable components.

Skill-Building Activities through Experiential Learning:

- Analyze blockchain vulnerabilities and security threats.
- Develop and test secure smart contracts.
- Evaluate blockchain performance metrics and scalability issues.
- Apply blockchain benchmarking and privacy enhancement techniques using industry tools..

Text Book

T1: Joseph Bonneau et al., SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies, IEEE Security & Privacy, 2015

T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 3rd Edition, 2020

References

R1: Andreas M. Antonopoulos, Mastering Ethereum, O'Reilly Media
R2: David Mazieres, The Stellar Consensus Protocol – A Federated Model for Internet-Level Consensus
R3: Christian Cachin et al., Introduction to Reliable and Secure Distributed Programming
R4: Nakamoto, Satoshi. Bitcoin: A Peer-to-Peer Electronic Cash System (whitepaper)

E-Resources/Web Resources

1. https://consensys.net
2. https://ethereum.org/en/developers/docs/security/
3. https://chainsecurity.com/
4. https://cryptozombies.io
5. https://bitcoin.org/en/security

Course Code: CBC2505	Course Name: Blockchain Security and Performance Lab Type of Course: Laboratory Course	L- T-P- C	0-0-2-1
Version No.	2.0		
NCrF Level-			
Course Pre-requisites	CBC2000		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course delves into the security and performance aspects of blockchain systems. It covers consensus attacks, smart contract vulnerabilities, performance bottlenecks, and secure protocol design. Emphasis is laid on techniques to improve resilience, scalability, and privacy while preserving decentralization.		
Course Objective	• Understand blockchain-specific threats and security principles • Analyze performance metrics in blockchain platforms • Explore security techniques for smart contracts and consensus algorithms • Evaluate design strategies to enhance scalability and privacy		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Identify and describe security challenges in blockchain platforms **(Level 2: Understand)** **CO2:** Evaluate consensus attacks and performance issues in different blockchain frameworks **(Level 4: Analyze)** **CO3:** Implement security-aware smart contracts and mitigation techniques **(Level 3: Apply)** **CO4:** Analyze and optimize blockchain performance under varying network conditions using benchmarking tools. **(Level 4: Analyze)**		

Course Content:

Week	Lab Experiment Title	Tools/Platforms
1	Introduction to Blockchain Security Concepts and Tools	Ganache, MetaMask
2	Setup and Configuration of Private Ethereum Network	Geth, Truffle

3	Smart Contract Vulnerability Assessment (Reentrancy Attack)	Remix IDE, Solidity
4	Implementing Access Control in Smart Contracts	Solidity, Truffle
5	Role-based Authorization with OpenZeppelin Libraries	OpenZeppelin, Remix
6	Timestamp Manipulation Attack Analysis	Remix, MetaMask
7	Replay Attack Detection and Prevention	Hardhat, Wireshark
8	Gas Optimization Techniques for Smart Contracts	Remix IDE, Truffle
9	DoS (Denial of Service) Attack Simulation and Handling	Solidity, Ganache
10	Event Logging and Auditing in Blockchain Systems	Web3.js, Ethers.js
11	Zero Knowledge Proof (ZKP) based Smart Contract Implementation	ZoKrates, Hardhat
12	Analyzing Blockchain Performance – Throughput and Latency	Hyperledger Caliper
13	Implementing Multi-signature Wallets	Solidity, Gnosis Safe
14	Secure Oracle Integration in Blockchain	Chainlink, Hardhat
15	Mini Project Presentation – Secure & Performance-oriented Blockchain Application	Tools as per project

Targeted Applications & Tools:

Ethereum Blockchain Platform	Used for developing and deploying smart contracts and decentralized applications (DApps) on a public blockchain network
Hyperledger Fabric	Used for building secure, permissioned enterprise blockchain applications for business networks.
Solidity Programming Language	Used for writing and developing smart contracts on Ethereum-based blockchain platforms.
Ganache Local Blockchain	Used for simulating a local Ethereum blockchain environment for testing smart contracts.
Hardhat / Truffle Framework	Used for smart contract development, testing, debugging, and deployment.
Open Zeppelin Libraries	Used for implementing secure and reusable smart contract components such as tokens and access control

Skill-Building Activities through Experiential Learning:

Develop the ability to **analyze blockchain vulnerabilities, implement secure smart contracts, and evaluate blockchain performance using industry-standard security analysis and benchmarking tools.**

Text Book

T1: Joseph Bonneau et al., SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies, IEEE Security & Privacy, 2015

T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 3rd Edition, 2020

References

R1: Andreas M. Antonopoulos, *Mastering Ethereum*, O'Reilly Media

R2: David Mazieres, *The Stellar Consensus Protocol – A Federated Model for Internet-Level Consensus*

R3: Christian Cachin et al., *Introduction to Reliable and Secure Distributed Programming*

R4: Nakamoto, Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System (whitepaper)*

E-Resources/Web Resources

1. https://consensys.net
2. https://ethereum.org/en/developers/docs/security/
3. https://chainsecurity.com/
4. https://cryptozombies.io
5. https://bitcoin.org/en/security

Course Code: CBC2506	Course Name: Blockchain Architecture Design Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	2.0		
NCrF Level-			
Course Pre-requisites	CBC 2001 Introduction to Block Chain Platforms		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course introduces the core concepts and architecture of blockchain systems. It covers decentralized ledger structures, consensus algorithms, smart contracts, and cryptographic techniques. Students will analyze public and private blockchain platforms, including Ethereum and Hyperledger. The course emphasizes the design, scalability, and security of blockchain-based applications. Real-world case studies illustrate blockchain's transformative potential across industries.		
Course Objective	• Understand the fundamental concepts, architecture, and components of blockchain technology. • Analyze various consensus mechanisms and evaluate their role in securing blockchain networks. • Design and develop smart contracts and decentralized applications using suitable blockchain platforms. • Evaluate the scalability, security, and real-world applicability of blockchain solutions across domains.		
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Understand the fundamental architecture and core components of blockchain systems, including data structures, cryptographic principles, and peer-to-peer networks. **CO2:** Analyze the design and implementation of consensus mechanisms and assess their impact on decentralization, performance, and security. **CO3:** Compare and evaluate the architectural designs and scalability mechanisms of leading blockchain platforms such as Ethereum, Hyperledger Fabric, and Corda. **CO4:** Analyze the design patterns, security considerations, and architectural trade-offs in developing robust and scalable blockchain solutions.		
Course Content:			

Module 1	Fundamentals of Blockchain Architecture	Assessment: Design Exercise	10 Sessions
Topics: Introduction to blockchain Distributed ledger technology Blockchain structure Blocks and chaining Merkle trees Hash functions Digital signatures Public and private keys P2P networks Cryptographic fundamentals			
Module 2	Consensus Mechanisms and Protocol Design	Assessment : Case Study	12 Sessions
Topics: Consensus principles Proof of Work (PoW) Proof of Stake (PoS) Practical Byzantine Fault Tolerance (PBFT) Delegated PoS Proof of Authority Block propagation Forking and finality Impact on decentralization and security			
Module 3	**Comparative Blockchain Platform Architectures**	Assessment : Experimental Assignment	11 Sessions
Topics: Ethereum architecture Smart contract execution with EVM Hyperledger Fabric components Channel architecture and endorsement policy Corda design principles Platform use-case suitability Scalability and performance considerations			
Module 4	Design Patterns, Security, and Scalability in Blockchain Systems	Assessment : Experimental Assignment	12 Sessions
Topics: Blockchain solution design process Architectural trade-offs Privacy layers and off-chain solutions			

Sharding

Sidechains

Layer 2 solutions

Security threats and mitigations

Blockchain development patterns

Case studies and best practices

Targeted Applications & Tools:

Ethereum Blockchain Platform	Used for developing decentralized applications (DApps), smart contracts, decentralized finance (DeFi) systems, digital assets, and blockchain-based public applications.
Hyperledger Fabric Framework	Building secure and permissioned enterprise blockchain solutions such as supply chain management, healthcare systems, banking applications, and business process automation.
Corda Platform	Secure and privacy-focused business transaction management for enterprise systems.

Skill-Building Activities through Experiential Learning:

- **Comparative Analysis of Blockchain Platforms**
  Learners perform comparative studies of leading blockchain platforms such as Ethereum, Hyperledger Fabric, and Corda by analyzing their architecture, consensus mechanisms, scalability models, transaction throughput, and permission management. Through case-based discussions and group activities, students evaluate platform suitability for enterprise and decentralized applications.
- **Smart Contract Development and Deployment**
  Using Solidity along with Remix IDE, Truffle, and Hardhat, students design, compile, test, and deploy smart contracts on local blockchain environments. Learners gain practical exposure to contract logic development, debugging, transaction execution, and decentralized application workflows.
- **Blockchain Security and Vulnerability Assessment**
  Students conduct hands-on security analysis by simulating attacks such as reentrancy attacks, replay attacks, timestamp manipulation, and denial-of-service attacks within blockchain systems. Using secure coding libraries such as OpenZeppelin, learners implement mitigation strategies and evaluate contract security against real-world threats.

Text Book

T1. Daniel Drescher, "Blockchain Basics: A Non-Technical Introduction in 25 Steps", Apress, 2017.
T2. Imran Bashir, "Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications", Packt Publishing, 2023 (4th Edition).

References

R1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, Princeton University Press, 2016.
R2. Sina Esfandyari, “Architecting Blockchain Solutions: A Guide to Design and Development”, Packt Publishing, 2021.
R3. Mouftah H.T., Al-Anbagi I. (Eds.), “Blockchain for Cybersecurity and Privacy: Architectures and Applications”, Springer, 2020.
R4. Rajeev Agrawal, Abhishek Kumar, Kamlesh Dutta, “Blockchain Technology: Architecture and Applications”, CRC Press, 2021.

E-Resources/Web Resources

1. https://www.ibm.com/blockchain/architecture
2. https://ethereum.org/en/whitepaper/
3. https://hyperledger-fabric.readthedocs.io/en/release-2.2/

4. https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

5. https://github.com/PacktPublishing/Mastering-Blockchain-Third-Edition

6. https://blockchainhub.net/

7. https://www.coindesk.com/learn/blockchain-101

Course Code: CSS2274	Course Title: 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. — **Level 2 – Understand** **CO2:** Analyze the time and space complexity of number-theoretic solutions to determine the need for optimization under competitive programming constraints. — **Level 4 – Analyze** **CO3:** Apply appropriate algorithmic paradigms and data structures to solve problems within defined time and memory constraints. — **Level 3 – 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. — **Level 4 – Analyze** **CO5:** Analyze advanced problem-solving and intelligent optimization techniques using dynamic programming, greedy methods, backtracking, and heuristic — **Level 4 – Analyze**		

	approaches for large-scale competitive programming problems.

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 Sessions**		
	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 Sessions**		
	Topics: - Parts of Speech, Subject Verb Agreement, Spotting Error, Cloze Test, Verbal Analogies, Reading Comprehension, Idioms & Phrases, Para Jumbles					
	Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS					

Evaluation	**Continuous Evaluation** • Topic wise evaluation
Text Book 1. Fast track objective by Rajesh Verma 2. R S Aggarwal 3. S.P Bakshi	
References 1. www.indiabix.com 2. www.testbook.com 3. www.youtube.com/c/TheAptitudeGuy/videos	
Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.	

Course Code: 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. [Level 2 - Understand] **CO2:** Apply appropriate techniques and modern tools to solve the intended problems. [Level 3 - Apply] **CO3:** Design experiments in accordance with standards and specifications. [Level 4- Analyze] **CO4:** Interpret events and results to derive meaningful conclusions. [Level 5- Evaluate] **CO5:** Appraise project findings and communicate effectively through scholarly publications. [Level 6 - 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 Sessions
	Topics: Resume structure, use of templates, Do's and Don'ts, ATS methods, Cover Letter and Video Resume **Activity:** Real world scenarios		
Module 2	**Group Discussion**	**Mock G D**	9 Sessions
	Topics: -Group discussion as a placement process, GD techniques like Keyword. SPELT & POV of affected parties. Do & Don't of GD, Case-lets and topics for GD, practice session and evaluation **Activity:-** Real world scenarios		
Module 3	**Personal Interview**	**Grooming checks + Evaluation + Mock Interview+ Role Play**	9 Sessions
	Topics: Placement process, Different interview rounds, HR interviews, Interview questions and desired answers, Different types of interviews, Do's and Don'ts. **Activity: -** Role Play & Real-world scenario		
Module 4	**Recap/Revision /Feedback Session**	**Practice sessions**	2 Sessions
	Targeted Application & Tools that can be used: 1. TED Talks 2. You Tube Links 3. Role Play activities		
	Project work/Assignment: Mention the Type of Project /Assignment proposed for this course		
	Continuous Individual Assessment		
	The Topics related to Skill Development: Art Of Presentation and Group Discussion for **Skill Development** through **Participative Learning** Tech- niques. This is attained through assessment Component mentioned in course handout.		

Course Code: CSS7300	**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 [Level 2 - Understand] **CO2:** Apply tools and techniques for problem solving. [Level 3 - Apply] **CO3:** Design and plan experiments per industry standards. [Level 3- Apply] **CO4:** Interpret the events and results for meaningful conclusions. [Level 5- Evaluate] **CO5:** Appraise project findings and communicate effectively through scholarly publications. [Level 6- Create]

ELECTIVES

Course Code: CBC3400	Course Name: Cryptography and Security in Blockchain Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course provides a deep dive into the cryptographic principles and security mechanisms that underpin blockchain technology. It explores key cryptographic algorithms, consensus protocols, wallet security, smart contract vulnerabilities, and advanced security features essential for safeguarding decentralized systems.		

Course Objective	Understand core cryptographic techniques used in blockchain ecosystems Explore consensus protocols and their security implications Analyze common attacks and vulnerabilities in blockchain networks Apply cryptographic methods for secure blockchain applications and smart contracts
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1 : Explain the role of cryptographic primitives in securing blockchain data and identity [Level 2 - Understand] CO2 (Analyze): Evaluate consensus protocols and blockchain attack surfaces [Level 2 – Analyze] CO3 (Apply): Implement and test cryptographic algorithms and secure wallet mechanisms [Level 3 – Apply] CO4 (Apply): Identify and mitigate security flaws in smart contracts and blockchain systems [Level 3 – Apply]

Course Content:

Module 1	Cryptographic Foundations	Assessment: Assignment	11 Sessions
Topics: Symmetric vs. asymmetric cryptography, Hash functions (SHA-256, Keccak), Digital signatures (ECDSA), Message integrity and non-repudiation, Merkle trees, Zero-knowledge proofs basics			
Module 2	Consensus and Blockchain Security	Assessment : Assignment	11 Sessions
Topics: Proof of Work (PoW), Proof of Stake (PoS), PBFT, Delegated PoS, Sybil attacks, 51% attack, Forking issues, Blockchain scalability vs. security trade-offs			
Module 3	Wallets and Transaction Security	Assessment : Assignment	11 Sessions
Topics: Topics: Public/private key generation, Wallet types (hot, cold, hardware), Multisig wallets, Transaction signing and verification, Secure key storage, Replay protection			
Module 4	Smart Contract and Platform Security	Assessment : Assignment	12 Sessions
Topics: Smart contract vulnerabilities (reentrancy, overflow), DAO case study, Formal verification, Auditing tools (Mythril, Slither), Secure coding standards, Layer-2 security (ZK-rollups, state channels), Privacy coins (ZCash, Monero)			

Text Book

T1 : Kelsey Hightower et al., Blockchain Security and Privacy, Wiley, 2023

T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 4th Edition, 2023

References

R1: Arvind Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press, 2022
R2: William Stallings, *Cryptography and Network Security*, Pearson, 7th Edition, 2022
R3: Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, 2022
R4: Nipun Jaswal, *Mastering Blockchain Security*, Packt Publishing, 2022

E-Resources/Web Resources

1.https://cryptozombies.io
2.https://soliditylang.org/security

3.https://ethereum.org/en/developers/docs/security
4.https://z.cash/technology
5.https://slither.readthedocs.io

Course Code: CBC3400	Course Name: Cryptography and Security in Blockchain Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course provides a deep dive into the cryptographic principles and security mechanisms that underpin blockchain technology. It explores key cryptographic algorithms, consensus protocols, wallet security, smart contract vulnerabilities, and advanced security features essential for safeguarding decentralized systems.		
Course Objective	Understand core cryptographic techniques used in blockchain ecosystems Explore consensus protocols and their security implications Analyze common attacks and vulnerabilities in blockchain networks Apply cryptographic methods for secure blockchain applications and smart contracts		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1 : Explain the role of cryptographic primitives in securing blockchain data and identity [Level 2 - Understand] CO2 (Analyze): Evaluate consensus protocols and blockchain attack surfaces [Level 2 – Analyze] CO3 (Apply): Implement and test cryptographic algorithms and secure wallet mechanisms [Level 3 – Apply] CO4 (Apply): Identify and mitigate security flaws in smart contracts and blockchain systems [Level 3 – Apply]		
Course Content:			
Module 1	Cryptographic Foundations	Assessment: Assignment	11 Sessions
Topics:			

Symmetric vs. asymmetric cryptography, Hash functions (SHA-256, Keccak), Digital signatures (ECDSA), Message integrity and non-repudiation, Merkle trees, Zero-knowledge proofs basics			
Module 2	Consensus and Blockchain Security	Assessment : Assignment	11 Sessions
Topics: Proof of Work (PoW), Proof of Stake (PoS), PBFT, Delegated PoS, Sybil attacks, 51% attack, Forking issues, Blockchain scalability vs. security trade-offs			
Module 3	Wallets and Transaction Security	Assessment : Assignment	11 Sessions
Topics: Topics: Public/private key generation, Wallet types (hot, cold, hardware), Multisig wallets, Transaction signing and verification, Secure key storage, Replay protection			
Module 4	Smart Contract and Platform Security	Assessment : Assignment	12 Sessions
Topics: Smart contract vulnerabilities (reentrancy, overflow), DAO case study, Formal verification, Auditing tools (Mythril, Slither), Secure coding standards, Layer-2 security (ZK-rollups, state channels), Privacy coins (ZCash, Monero)			
Text Book T1 : Kelsey Hightower et al., Blockchain Security and Privacy, Wiley, 2023 T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 4th Edition, 2023			
References R1: Arvind Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press, 2022 R2: William Stallings, *Cryptography and Network Security*, Pearson, 7th Edition, 2022 R3: Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, 2022 R4: Nipun Jaswal, *Mastering Blockchain Security*, Packt Publishing, 2022			
E-Resources/Web Resources 1.https://cryptozombies.io 2.https://soliditylang.org/security 3.https://ethereum.org/en/developers/docs/security 4.https://z.cash/technology 5.https://slither.readthedocs.io			

Course Code: CBC3401	Course Name: Crypto Trading Strategies & Risk Management Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		

Anti-requisites	NIL		
Course Description	This course provides an in-depth understanding of crypto trading mechanisms, investment strategies, and risk management techniques. It covers technical and fundamental analysis, trading indicators, automated strategies, and regulatory considerations. Emphasis is placed on market psychology, portfolio diversification, and risk mitigation in highly volatile crypto markets		
Course Objective	Understand cryptocurrency markets, exchanges, and trading instruments Learn technical and fundamental analysis techniques for crypto assets Explore risk assessment models and portfolio protection strategies Apply trading strategies and automation tools within regulatory frameworks _____		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1 : Explain cryptocurrency market structures, instruments, and volatility characteristics [Level 2 - Understand] CO2 : Evaluate and compare crypto trading strategies and their performance indicators [Level 2 – Analyze] CO3: Apply risk management techniques and portfolio diversification methods [Level 3 – Apply] CO4 : Use tools to implement, simulate, or automate crypto trading strategies [Level 3 – Apply]		
Course Content:			
Module 1	Cryptocurrency Markets & Exchanges	Assessment: Assignment	11 Sessions
Topics: Overview of crypto assets, Exchanges (CEX vs DEX), Trading pairs, Order types (market, limit, stop-loss), Market liquidity and volatility, Understanding spreads, slippage, and fees			
Module 2	Trading Strategies and Analytics	Assessment : Assignment	11 Sessions
Topics: Technical indicators (MACD, RSI, Bollinger Bands), Candlestick patterns, Moving averages, Breakout and momentum strategies, Arbitrage and swing trading, Volume analysis, On-chain analytics			
Module 3	Risk Management in Crypto Trading	Assessment : Assignment	11 Sessions
Topics: Topics: Position sizing, Risk-reward ratio, Stop-loss vs trailing stops, Leverage and margin risks, Portfolio diversification, Drawdown analysis, Hedging strategies			
Module 4	Automation, Tools & Regulations	Assessment : Assignment	12 Sessions
Topics: Trading bots and scripting basics (Python, Pionex), Backtesting tools (TradingView, CryptoCompare), Trading APIs (Binance, Coinbase), Regulatory risks, KYC/AML, Taxation, Real-world strategy simulation			
Text Book			

T1 : Glen Goodman, The Crypto Trader: How Anyone Can Make Money Trading Bitcoin and Other Cryptocurrencies, Harriman House, 2023
T2: Rolf Schlotmann, Crypto Trading: A Comprehensive Beginner's Guide, Independently Published, 2022

References

R1: Steve Burns, Technical Analysis for Crypto Traders, New Trader U, 2022
R2: Michael Radkay, Cryptocurrency Risk Management, Wiley, 2023
R3:Victor Lucas, The Cryptocurrency Trading Guide, Kindle Edition, 2022
R4: Nik Patel, An Altcoin Trader's Handbook, Amazon Publishing, 2021

E-Resources/Web Resources

1. https://www.tradingview.com
2. https://coinmarketcap.com
3. https://pionex.com
4. https://academy.binance.com
5. https://cryptoquant.com

Course Code: CBC3402	Course Name: Bitcoin and Ethereum Protocols Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course offers an in-depth exploration of the technical protocols behind Bitcoin and Ethereum, the two most prominent blockchain platforms. It focuses on consensus algorithms, cryptographic structures, transaction lifecycles, smart contract execution, protocol upgrades, and governance models. Students will gain foundational knowledge to build, secure, and innovate on these platforms		
Course Objective	Understand the architecture and protocols of Bitcoin and Ethereum blockchains Explore the transaction mechanisms and consensus protocols of both platforms Learn how Ethereum enables smart contracts and decentralized applications Analyze protocol limitations, upgrades, and scalability efforts		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1 : Describe the technical components and operation of Bitcoin and Ethereum [Level 2 - Understand] CO2: Compare consensus mechanisms and data structures used in both platforms [Level 2 – Analyze] CO3: Trace transaction flow and analyze smart contract execution [Level 3 – Apply] CO4: Evaluate scaling solutions, forks, and governance in decentralized networks [Level 3 – Apply]		

Course Content:			
Module 1	Bitcoin Protocol Internals	Assessment: Assignment	11 Sessions
Topics: Bitcoin architecture, UTXO model, SHA-256 and ECDSA cryptography, Merkle trees, Block structure and mining, Proof-of-Work (PoW), Difficulty adjustment, Nakamoto consensus			
Module 2	Ethereum Platform and EVM	Assessment : Assignment	11 Sessions
Topics: Ethereum accounts model (EOA vs Contract), Gas and transaction fees, Ethereum Virtual Machine (EVM), Solidity basics, State transitions, Keccak-256 hashing, Contract storage and execution			
Module 3	Transaction Lifecycle & Security	Assessment : Assignment	11 Sessions
Topics: Transaction creation, mempool, mining and validation, Transaction finality, Smart contract deployment, Reentrancy and gas-related vulnerabilities, Forking and transaction replay			
Module 4	Protocol Upgrades, Governance & Scaling	Assessment : Assignment	12 Sessions
Topics: Bitcoin forks (SegWit, Taproot), Ethereum forks (Byzantium to Shanghai), Ethereum 2.0 and Proof-of-Stake (PoS), Rollups and Layer 2 scaling, DAO governance models, Limitations and innovations			
Text Book T1: Andreas M. Antonopoulos, Mastering Bitcoin, O'Reilly Media, 2nd Edition, 2023 T2: Andreas M. Antonopoulos & Gavin Wood, Mastering Ethereum, O'Reilly Media, 2023			
References R1: Arvind Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press, 2022 R2: Imran Bashir, Mastering Blockchain, Packt Publishing, 2023 R3: Ritesh Modi, Introducing Ethereum and Solidity, Apress, 2022 R4: Roger Wattenhofer, The Science of the Blockchain, InTechOpen, 2021			
E-Resources/Web Resources 1. https://bitcoin.org/en/developer-documentation 2. https://ethereum.org/en/developers/docs 3. https://soliditylang.org 4. https://github.com/ethereum/EIPs 5. https://bitcoin.stackexchange.com			

Course Code: CBC3403	Course Name: Blockchain for Digital Identity Management Type of Course: Theory Course	L- T-P- C	3-0-0-3

Version No.	X.0		
NCrF Level-	5.5		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course explores blockchain-based digital identity systems, focusing on decentralized identifiers (DIDs), verifiable credentials (VCs), and self-sovereign identity (SSI). It addresses architectural models, privacy-preserving technologies, and compliance frameworks like GDPR. Emphasis is placed on hands-on implementation, interoperability, biometric linkage, and emerging trends in Web3, metaverse, and cross-border identity solutions.		
Course Objective	• Understand the challenges of traditional identity management systems • Explore blockchain-based identity frameworks and standards • Learn to design and implement decentralized digital identity models • Evaluate security, privacy, and compliance aspects in identity solutions		
Course Outcomes	On completion of this course, students will be able to: • CO1: Define key concepts related to digital identity, authentication, and authorization within traditional and blockchain-based systems. (Level 1 - Remember) • CO2: Explain the architecture, standards, and protocols used in decentralized identity systems, including DIDs and VCs. (Level 2 - Understand) • CO3: Distinguish between centralized and self-sovereign identity models and examine their security and privacy implications. (Level 4 - Analyze) • CO4: Construct verifiable credentials and DIDs using frameworks like Hyperledger Indy or uPort. (Level 6 - Create) • CO5: Demonstrate the application of blockchain-based identity solutions in sectors like e-governance, healthcare, and banking, concerning compliance frameworks like GDPR and KYC. (Level 3 - Apply)		
Course Content:			
Module 1	Digital Identity Foundations & Challenges	Assessment : MCQ, Case-Based Quiz	9 Sessions
Topics: Identity lifecycle: provisioning, authentication, revocation, Centralized & federated identity models (OAuth, SAML), Risks: data breaches, identity theft, vendor lock-in, Introduction to blockchain-based identity models, Biometric identity challenges and blockchain linkage.			
Module 2	Decentralized Identity Standards & Ecosystems	Assessment : Comparative Matrix & Role Play	9 Sessions
Topics: Decentralized Identifiers (DIDs), Verifiable Credentials (VCs), Self-Sovereign Identity (SSI) principles, W3C standards, DIF, ToIP Framework, Selective disclosure, Zero-Knowledge Proofs (zk-SNARKs), DIDComm and credential exchange protocols.			
Module 3	Blockchain Identity Platforms & Architectures	Assessment : Hands-on Identity Wallet Lab	9 Sessions
Topics: Hyperledger Indy, Aries, uPort, ION (Microsoft), Identity layers on blockchain (Layer 0 to 3), Wallets: Custodial vs. non-custodial, mobile ID wallets, Issuer–Holder–Verifier model, MetaMask + Ceramic + DIDs integration (hands-on tools).			
Module 4	Implementation, Interoperability & Security	Assessment : Use Case Evaluation	9 Sessions
Topics:			

Smart contracts for credential issuance, Cross-platform interoperability (e.g., Aries Frameworks across ledgers), SSI in cross-border identity, e-passports, educational credentials, Secure revocation, credential expiry & fraud prevention, Blockchain analytics & risk scoring in identity systems.

Module 5	Regulations, Ethics, and Emerging Trends	Assessment : Research Presentation	9 Sessions

Topics:

GDPR, eIDAS, KYC/AML frameworks, Ethics in decentralized identity (data ownership, surveillance concerns), AI-powered identity scoring and fraud detection, Biometric SSI models (voice, iris, fingerprint linked DIDs), Decentralized identity in Metaverse, IoT, and Web3 ecosystems.

Targeted Applications & Tools:

Tools/Software/Platform	Application
Hyperledger Indy, uPort	Create a DID (Decentralized Identifier) using a blockchain-based identity framework
Veramo, Trinsic Studio	Design and issue a Verifiable Credential (VC) for academic certification
Aries Framework, Indy SDK	Simulate a credential verification process using issuer-holder-verifier architecture
IDunion ZKP Playground, Zokrates	Implement selective disclosure using Zero-Knowledge Proofs (ZKP)
Case study analysis with W3C DID Spec and GDPR guidelines	Analyze a real-world SSI system (e.g., Aadhaar, eIDAS) for decentralization and privacy gaps

Skill-Building Activities through Experiential Learning:

- Build Your Decentralized Identity (DID) Profile

  Participants create their own decentralized identity using platforms like uPort or Trinsic Studio. They generate cryptographic key pairs, register their DID on a blockchain test network, and associate attributes such as name, email, and photo. This activity builds practical understanding of self-sovereign identity creation and DID management.

- Issue and Verify a Verifiable Credential (VC)

  Learners simulate a real-world credential issuance process using the issuer-holder-verifier model. Teams use tools like Hyperledger Aries or Trinsic to issue, share, and verify digital credentials such as "Certified Blockchain Analyst." This helps participants understand decentralized credential workflows and their role in building trust without centralized intermediaries.

- Identity Risk Assessment in Centralized vs. Decentralized Systems

  This activity involves investigating real-world identity breaches, such as Aadhaar data leaks or Facebook privacy scandals. Learners analyze vulnerabilities in centralized identity systems and explore how decentralized, blockchain-based identity solutions could have mitigated or prevented such incidents, strengthening understanding of privacy, security, and data ownership principles.

Text Book

T1: Alex Preukschat & Drummond Reed, Self-Sovereign Identity, Manning Publications, 2023
T2: Mearian Lucas, Blockchain for Digital Identity, BPB Publications, 2022
T3: Information Security and Cryptology: 15th International Conference, Inscrypt 2019, Nanjing, China, December 6–8, 2019, Revised Selected Papers. (2020). Germany: Springer International Publishing.

References

R1: Imran Bashir, Mastering Blockchain, Packt Publishing, 2023
R2: Arvind Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press, 2022

R3: Phillip J. Windley, Digital Identity, O'Reilly Media, 2021 R4: Sovrin Foundation Whitepapers and ToIP Framework Documents
E-Resources/Web Resources 1. https://www.w3.org/TR/did-core/ 2. https://identity.foundation 3. https://trustoverip.org 4. https://www.ibm.com/solutions/blockchain-identity 5. https://www.nec.com/en/global/solutions/blockchain/blockchain-for-digital-identity.html

Course Code: CBC3404	Course Name: Cryptocurrency Wallet Development Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	X.0		
NCrF Level-	5.5		
Course Pre-requisites	CBC2000		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course delves into the architecture, development, and security of cryptocurrency wallets across blockchain ecosystems. It covers wallet types, cryptographic key management, transaction signing, SDK integration, user interface design, and DApp connectivity. Emphasis is placed on secure coding, compliance, and deploying feature-rich wallets for multi-chain and decentralized environments.		
Course Objective	• Understand the fundamentals of cryptocurrency wallets and key management • Learn to build and secure wallets for Bitcoin, Ethereum, and other blockchains • Explore wallet SDKs, libraries, and integration techniques • Apply secure coding practices to prevent theft, fraud, and misuse		
Course Outcomes	On completion of this course, students will be able to: • CO1: Define the types, components, and structures of cryptocurrency wallets and key generation mechanisms. (Level 1 - Remember) • CO2: Explain how wallet security mechanisms mitigate common threats such as phishing, key theft, and unauthorized access. (Level 2 - Understand) • CO3: Illustrate the role of SDKs and APIs in enabling wallet functionality and interaction with blockchain networks. (Level 3 - Apply) • CO4: Develop basic cryptocurrency wallet functionality, including key management and transaction signing using Web3 libraries. (Level 6 - Create) • CO5: Integrate wallet features with decentralized applications (DApps) using standard protocols like WalletConnect and MetaMask. (Level 3 – Apply)		
Course Content:			
Module 1	Wallet Architecture & Key Management	Assessment : Design Blueprint, Quiz	9 Sessions
Topics: Types of wallets: hot, cold, custodial, non-custodial, multisig, HD wallets (BIP32/39/44), seed phrases, key derivation paths, Address generation and formats (Bech32, Base58Check), Public/private key generation using secp256k1, Key encryption, local vs secure enclave storage, Backup & recovery: Shamir Secret Sharing, QR-based cold backups, Introduction to Smart Contract Wallets (e.g., Safe Wallet).			
Module 2	Wallet Security & Compliance	Assessment : Threat Modeling, Case Study	9 Sessions
Topics:			

Common attack vectors: phishing, malware, SIM swap, keylogging, Hardware Security Modules (HSMs), Secure Enclave (TEE), Multi-factor authentication (2FA, biometric unlock), KYC/AML integration practices in wallet apps, FATF Travel Rule, GDPR, and MiCA compliance overview, Transaction signing delegation and social recovery, Privacy-preserving features (Stealth addresses, CoinJoin, zkWallets)

Module 3	Wallet Development & SDK Integration	Assessment : Wallet App Prototype	9 Sessions

Topics:

Transaction lifecycle in Bitcoin and Ethereum, Wallet SDKs and libraries: bitcoinjs-lib, ethers.js, Web3.js, Transaction signing, broadcasting, nonce management, Multi-currency and token (ERC-20, BEP-20) wallet handling, UI/UX principles for mobile/web wallet apps, QR code, NFC integration for transactions, Error handling, confirmations, and fee estimation APIs.

Module 4	Advanced Wallet Features & DApp Integration	Assessment : Functional Wallet + DApp Link	9 Sessions

Topics:

Web3 wallet interaction: MetaMask, WalletConnect, RainbowKit, Smart contract interactions: send/receive tokens, stake/unstake, EIP-712 signing and EIP-4337 (Account Abstraction) overview, NFT wallet support: ERC-721, ERC-1155, Token swaps and bridges within wallets, Lightning Network wallets and micropayments, Integrating with Wallet APIs (Ledger, Trezor, Fireblocks).

Module 5	Wallet Testing, Deployment & Maintenance	Assessment : Project Demo, Peer Review	9 Sessions

Topics:

Unit and integration testing of wallet modules, Using Testnets (Goerli, Sepolia, Bitcoin Testnet), Wallet CI/CD pipelines and upgrade handling, Open-source wallet code audit (e.g., Electrum, MetaMask), Monitoring on-chain activity for anomalies (e.g., wallet draining), App publishing and app store compliance (Play Store, App Store), Incident response planning and user support design

Targeted Applications & Tools:

Tools/Software/Platform	Application
ethers.js, MetaMask, Remix IDE, Infura	ethers.js, MetaMask, Remix IDE, Infura
bitcoinjs-lib, qrcode, web3.js, Node.js	bitcoinjs-lib, qrcode, web3.js, Node.js
Ledger Live, Trezor Bridge, WalletConnect	Ledger Live, Trezor Bridge, WalletConnect
bip39, bip32, bip44, React, Redux	bip39, bip32, bip44, React, Redux
Jumio, Shyft, Civic, Chainalysis APIs	Jumio, Shyft, Civic, Chainalysis APIs

Skill-Building Activities through Experiential Learning:

- Develop & Deploy a Basic Ethereum Wallet

  Participants will create and deploy a simple Ethereum wallet on a test network. They will generate keys using a mnemonic phrase, design a basic interface for transactions, and sign and broadcast transactions using tools like ethers.js and MetaMask, gaining practical exposure to wallet creation and transaction handling.

- Implement QR Code Based Transaction Flow

  This activity focuses on building a QR code-based transaction feature for enhanced usability. Learners will generate a wallet address, convert it into a QR code, scan it on another device, and securely sign transactions, experiencing secure cross-device interaction for cryptocurrency transactions across mobile and web platforms.

- Simulate Hardware Wallet Integration

  Participants will simulate interaction with hardware wallets such as Ledger or Trezor using emulators and testnets. They will retrieve public keys, sign transactions, and broadcast them, understanding the importance of secure key storage, phishing protection, and hardware wallet integration in real-world cryptocurrency wallet systems.

Text Book
T1: Andreas M. Antonopoulos, Mastering Bitcoin, O'Reilly Media, 2nd Edition, 2023 T2: Chris Dannen, Introducing Ethereum and Solidity, Apress, 2022 T3: Cryptocurrency Wallet: A Guide To Secure Digital Asset Management in Decentralized Systems. (2025). (n.p.): One Billion Knowledgeable.
References
R1: Imran Bashir, Mastering Blockchain, Packt Publishing, 2023 R2: Tim Coulter, Blockchain Developer's Guide, Packt Publishing, 2022 R3: Joseph Bonneau et al., SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies, IEEE Security & Privacy, 2022 R4: Ledger Academy and Trezor Docs (for practical wallet development)
E-Resources/Web Resources
6. https://web3js.readthedocs.io 7. https://docs.ethers.org 8. https://bitcoin.org/en/developer-guide 9. https://developer.metamask.io 10. https://walletconnect.com 11. https://tokenminds.co/blog/knowledge-base/cryptocurrency-wallet-development-step-by-step-guide 12. https://www.designrush.com/agency/blockchain-development-companies/trends/cryptocurrency-wallet-development

Course Code: CBC3405	Course Name: Blockchain Security & Ethical Hacking Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	X.0		
NCrF Level-	5.5		
Course Pre-requisites	CCS2504		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course explores security risks and ethical hacking practices in blockchain ecosystems, covering vulnerabilities in smart contracts, consensus protocols, wallets, and networks. It emphasizes secure development, penetration testing, and auditing techniques. Practical exposure is provided through tools and case studies to detect, exploit, and mitigate threats in decentralized applications.		
Course Objective	• Understand blockchain security principles and ethical hacking frameworks • Explore attack vectors in consensus, smart contracts, wallets, and networks • Learn secure development practices and ethical exploitation techniques • Apply auditing tools and mitigation strategies to secure blockchain ecosystems		
Course Outcomes	On completion of this course, students will be able to: • CO1: Describe the principles of blockchain security, including system design considerations, attack surfaces, and the role of ethical hacking. (Level 2 - Understand) • CO2: Explain common vulnerabilities in smart contracts, cryptocurrency wallets, and blockchain consensus mechanisms. (Level 2 - Understand) • CO3: Identify potential threats in blockchain networks and categorize different types of attacks, such as Sybil attacks, 51% attacks, and smart contract exploits. (Level 2 - Understand) • CO4: Use appropriate auditing tools to inspect smart contracts and blockchain networks for security flaws and vulnerabilities. (Level 3 - Apply)		

	• CO5: Simulate ethical hacking scenarios in a controlled lab environment to demonstrate typical exploits and implement mitigation strategies. (Level 3 - Apply)

Course Content:

Module 1	Blockchain Threat Models & Risk Landscape	Assessment : Concept Quiz, Threat Map	9 Sessions

Topics:

Threat modeling for decentralized systems, Consensus-level attacks: 51%, selfish mining, long-range, Wallet vulnerabilities: seed phrase phishing, dusting, Smart contract risks: reentrancy, integer overflow, DoS, Ethical hacking principles, legal considerations, scope definition, Rug pulls and honeypot contract scams, Risk scoring with CVSS for smart contracts.

Module 2	Penetration Testing & Blockchain Exploits	Assessment : Practical Lab: Vulnerability Exploitation	9 Sessions

Topics:

Setting up a testnet for ethical hacking, Automated vulnerability detection in Solidity, Replay attacks, front-running, sandwich attacks, Flash loan and oracle manipulation exploits, Bug bounty lifecycle (reporting, triage, rewards), Simulated hacks on forked DeFi protocols.

Module 3	Secure Smart Contract & Wallet Development	Assessment : Secure Contract Build, Code Audit	9 Sessions

Topics:

Secure design patterns (checks-effects-interactions, circuit breaker), Solidity security: visibility, fallback, delegatecall issues, Wallet development hardening: MPC, biometric integration, Reentrancy guards, pausable contracts, time locks, OWASP Top 10 for Smart Contracts.

Module 4	Auditing Tools, Automation & DevSecOps	Assessment : Audit Report + Automation Demo	9 Sessions

Topics:

Security tools: Mythril, Slither, Echidna, Manticore, Hardhat Security, Formal verification and symbolic execution, CI/CD integration of smart contract security checks, GitHub Actions for contract testing and deployment, Secure RPC and node configuration (e.g., Geth, Hardhat).

Module 5	Real-World Exploits & Defense Mechanisms	Assessment : Case Analysis & Strategy Pitch	9 Sessions

Topics:

Case study: Poly Network exploit, Ronin Bridge attack, Curve pool bug, Understanding Maximal Extractable Value (MEV) and defenses, Decentralized bridge vulnerabilities and cross-chain threats, DApp security monitoring and anomaly detection (Forta, Tenderly), DeFi insurance protocols and security rating platforms (Certik, Hacken).

Targeted Applications & Tools:

Tools/Software/Platform	Application
Mythril, Slither, Remix IDE	Perform vulnerability analysis on a smart contract to detect reentrancy and integer overflows
Ganache, Hardhat, NetStats Dashboard	Simulate a 51% attack on a private blockchain testnet

Burp Suite, MetaMask, OWASP ZAP, Wireshark	Conduct penetration testing on a Web3 DApp for wallet and input validation vulnerabilities
Truffle, Ganache, Remix IDE, Ethers.js	Analyze the DAO hack and reproduce the exploit in a test environment
Solidity, Hardhat, OpenZeppelin Libraries	Develop a secure smart contract implementing fallback protection and reentrancy guards

Skill-Building Activities through Experiential Learning:

- Smart Contract Hackathon (Reentrancy Challenge)

  Participants engage in a controlled hackathon where they identify and exploit a reentrancy vulnerability in a deliberately insecure Solidity smart contract using Remix IDE and MetaMask. After demonstrating the exploit, they apply industry best practices to patch the flaw using reentrancy guards and preventive coding techniques.

- Blockchain Forensics and Exploit Tracing

  Learners trace real-world blockchain exploits such as the DAO or Poly Network hacks using tools like Etherscan or Chainalysis demo platforms. They analyze transaction flows, identify the exploit vector, and document findings in a report, enhancing their understanding of blockchain forensics and incident response.

- Penetration Test of a Web3 DApp

  In this activity, participants perform a penetration test on a deployed decentralized application (DApp) using tools like OWASP ZAP and Burp Suite. They identify vulnerabilities such as XSS or insecure wallet integration, simulate potential attacks, and provide documented security recommendations to improve the DApp's security posture.

Text Book

T1: Rajneesh Gupta, Blockchain Security and Ethical Hacking, BPB Publications, 2023
T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 4th Edition, 2023
T3: Maurushat, A. (2019). Ethical Hacking. Canada: University of Ottawa Press.

References

R1: Joseph Bonneau et al., SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies, IEEE, 2022
R2: Mudit Gupta, Smart Contract Hacking Handbook, Immunefi Community Docs, 2022
R3: Damilare Daramola, Ethereum Smart Contract Security, Apress, 2023
R4: Kevin Mitnick, The Art of Invisibility, Little Brown, 2022

E-Resources/Web Resources

13. https://consensys.github.io/smart-contract-best-practices/
14. https://cryptozombies.io
15. https://immunefi.com/learn
16. https://github.com/crytic/slither
17. https://www.coursera.org/learn/blockchain-security

Course Code: CBC3406	Course Name: Introduction to Artificial Intelligence in Blockchain Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		

Anti-requisites	NIL
Course Description	This course introduces how Artificial Intelligence (AI) can be integrated with blockchain technology to create intelligent, secure, and autonomous decentralized systems. Topics include AI-driven smart contracts, secure data sharing, AI model integrity on-chain, blockchain for AI auditability, and applications in finance, healthcare, and supply chain
Course Objective	Understand the synergy between AI and blockchain technologies Explore architectures that combine decentralized data with intelligent decision-making Learn use cases of AI-powered blockchain applications Analyze challenges in data privacy, trust, and model governance in AI-blockchain systems
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1 : Explain the fundamentals of AI and blockchain and their convergence [Level 2 - Understand] CO2 : Evaluate use cases and architectures of AI-powered blockchain systems [Level 2 – Analyze] CO3 : Integrate AI models into smart contracts and decentralized applications [Level 3 – Apply] CO4 : Examine the challenges and ethical concerns of deploying AI on the blockchain [Level 3 – Apply]

Course Content:

Module 1	Fundamentals of AI and Blockchain	Assessment: Assignment	11 Sessions
Topics: Overview of AI (ML, DL, NLP), Blockchain structure, consensus, and smart contracts, AI vs blockchain roles, Benefits of integration, Blockchain for AI audit trails and data provenance			
Module 2	Intelligent Smart Contracts & AI Models On-Chain	Assessment : Assignment	11 Sessions
Topics: Embedding decision-making into contracts, Oracle networks, AI-driven DApps, Federated learning and blockchain, Edge AI and decentralized AI agents			
Module 3	Use Cases in Industry	Assessment : Assignment	11 Sessions
Topics: AI in DeFi fraud detection, Healthcare diagnostics with secure sharing, Predictive analytics in blockchain logistics, Tokenizing AI models, Model training marketplaces, NFT + AI systems			
Module 4	Challenges, Ethics & Future Directions	Assessment : Assignment	12 Sessions
Topics: Data privacy, GDPR, Explainable AI (XAI) on blockchain, Resource constraints, Model verification and updates, Regulatory landscape, Ethical implications, Roadmap to autonomous decentralized AI			

Text Book T1: Arshdeep Bahga & Vijay Madisetti, Blockchain Applications: A Hands-On Approach, VPT, 2023 T2: Mohammad Rezaul Karim, AI and Blockchain for Beginners, Packt Publishing, 2023
References R1: Imran Bashir, Mastering Blockchain, Packt Publishing, 2023 R2: Kai-Fu Lee, AI Superpowers: China, Silicon Valley, and the New World Order, HMH, 2022 R3: Bhaskar Krishnamachari, Blockchain and AI for Internet of Things, Springer, 2022 R4: Jaspreet Bindra, The Tech Whisperer: On Digital Transformation and the Technologies That Enable It, Penguin, 2021
E-Resources/Web Resources 1. https://ai.google 2. https://ethereum.org/en/developers/docs/oracles 3. https://deeplearning.ai 4. https://oceanprotocol.com 5. https://h2020bonseyes.eu

Course Code: CBC3407	Course Name: Machine Learning for Cyber Threat Detection Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		
Course Description	This course focuses on the application of machine learning techniques for detecting and preventing cybersecurity threats. It explores supervised, unsupervised, and deep learning methods for anomaly detection, intrusion detection systems (IDS), malware analysis, and phishing detection. Real-time threat analytics, datasets, and evaluation metrics are emphasized		
Course Objective	Understand the fundamentals of cyber threats and threat detection strategies Learn to apply machine learning models to identify malicious activities Explore threat datasets, feature engineering, and evaluation metrics Analyze and compare ML techniques for cybersecurity use cases		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1 : Explain the types of cyber threats and the role of machine learning in threat detection [Level 2 - Understand] CO2 : Evaluate datasets and select appropriate ML models for different security tasks [Level 2 – Analyze] CO3 : Build and train ML models for intrusion detection and malware classification [Level 3 – Apply] CO4: Implement real-time threat detection and evaluate model performance [Level 3 – Apply]		

Course Content:			
Module 1	Introduction to Cybersecurity and Threat Types	Assessment: Assignment	11 Sessions
Topics: Cyber threat landscape, Attack vectors (phishing, malware, DoS, ransomware), Indicators of compromise (IOCs), Traditional IDS vs ML-based IDS, Need for intelligent threat detection			
Module 2	Machine Learning for Threat Modeling	Assessment : Assignment	11 Sessions
Topics: Supervised vs unsupervised learning, Feature selection from logs/packets, Preprocessing and encoding, Cyber threat datasets (NSL-KDD, CICIDS, CTU-13), Model selection strategies			
Module 3	Model Development and Application	Assessment : Assignment	11 Sessions
Topics: Logistic regression, Decision trees, SVM, Random Forests, K-means, Autoencoders, Neural networks for classification, Evaluation metrics (precision, recall, F1, ROC-AUC)			
Module 4	Advanced Detection and Real-World Use Cases	Assessment : Assignment	12 Sessions
Topics: Deep learning for APT and malware detection, Threat intelligence integration, Adversarial ML, Real-time IDS with streaming data, SOC and SIEM integration, Case studies: phishing, botnet, zero-day detection			
Text Book T1: Emmanuel Tsukerman, Machine Learning for Cybersecurity Cookbook, Packt Publishing, 2023 T2: Clarence Chio & David Freeman, Machine Learning and Security, O'Reilly Media, 2023			
References R1: Xiaofeng Chen, AI in Cybersecurity, Springer, 2022 R2: Richard MacDonald, Practical Machine Learning for Cybersecurity, Apress, 2021 R3: Mark Stamp, Information Security: Principles and Practice, Wiley, 2022 R4: Kim Crawley, Hacking AI: The Security Threat of Artificial Intelligence, Wiley, 2022			
E-Resources/Web Resources 1. https://www.kaggle.com/datasets 2. https://www.unb.ca/cic/datasets			

3. https://github.com/nu11secur1ty
4. https://scikit-learn.org/stable/
5. https://cybersecurity.att.comhttps://h2020bonseyes.eu

Course Code: CBC3408	Course Name: AI-Powered Fraud Detection in Blockchain Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	(new version to be added by PU)		
NCrF Level-	5.5		
Course Pre-requisites	CAI2500		
Co-requisites	NIL		
Anti-requisites			
Course Description	This course explores the use of Artificial Intelligence for detecting fraud in blockchain networks. It covers anomaly detection, behavioral analysis, and machine learning models for identifying suspicious patterns in transactions, smart contracts, and wallets. Emphasis is placed on real-time analytics, visualization, regulatory compliance, and contemporary DeFi fraud case studies.		
Course Objective	•		
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1: Define the types of fraud commonly occurring in blockchain and decentralized finance environments, including transaction-level and protocol-level exploits. (Level 1 - Remember) • CO2: Describe how AI and machine learning techniques are applied to detect and mitigate fraudulent activities in blockchain systems. (Level 2 - Understand) • CO3: Identify suspicious wallet behaviors, transaction patterns, and anomalies using AI-driven detection models. (Level 2 - Understand) • CO4: Apply supervised and unsupervised machine learning techniques to build basic fraud detection models for blockchain data. (Level 3 - Apply) • CO5: Use standard fraud detection tools, APIs, and real-time dashboards to monitor and analyze blockchain transactions for potential fraud. (Level 3 - Apply)		
Course Content:			
Module 1	Blockchain Fraud, Risk Landscape & Explainability	Assessment : Case Analysis	9 hours
Topics: • Blockchain fraud types: double-spending, Sybil attacks, wash trading, rug pulls, phishing, insider attacks • Emerging threats in DeFi, NFTs, flash loan exploits • AML/KYC challenges in pseudonymous networks • Introduction to Explainable AI (XAI) in regulatory compliance			

• Risk scoring and fraud taxonomies (e.g., FATF red flags)			
Module 2	Machine Learning & Deep Learning for Fraud Detection	Assessment : ML Model Training Task	9 hours
Topics: • Supervised algorithms: Logistic regression, Decision Trees, XGBoost, SVM • Unsupervised techniques: K-means, Isolation Forest, Autoencoders • Transaction graph analytics: Graph Neural Networks (GNNs), DeepWalk • Feature engineering for behavioral fingerprinting • Real-world datasets: Elliptic, AMLSim, TONIC			
Module 3	Smart Contract, NFT & Wallet Behavior Analytics	Assessment: Smart Contract Analyzer Project	9 hours
Topics: • Behavior analysis of wallets and high-risk contracts • Detection of honeypots, token backdoors, and time-locked exploits • Contract risk profiling using LLMs (OpenAI Code Interpreter, GPT-4) • NFT frauds: wash trading, fake ownership trails • Wallet clustering, profiling, and social graph inference			
Module 4	Real-Time Detection Systems and Visualization	Assessment : Fraud Alert Dashboard Implementation	9 hours
Topics: • Building fraud dashboards: Kibana, Grafana, ElasticSearch • Real-time stream analytics (Apache Kafka, Streamlit) • Integration of alert engines (SIEMs, SOAR) • Case studies: Curve exploit, Wormhole hack, Ronin bridge attack • Use of Chainalysis, CipherTrace, AMLBot APIs for transaction enrichment			
Module 5	Blockchain Forensics, Adversarial AI & Ethical Considerations	Assessment : Whitepaper Review and Risk Simulation	9 hours
Topics: • Blockchain forensics workflow and visual tracing (GraphSense, Crystal) • Adversarial AI in fraud detection systems: poisoning, evasion attacks • Risk of model hallucinations and false positives in real-time systems • Ethical AI and trust frameworks for fraud detection • Regulatory frameworks (MiCA, FATF Travel Rule, FinCEN guidelines)			
Targeted Applications & Tools:			
Etherscan API, Python (pandas, scikit-learn), NetworkX	Analyze Ethereum transaction data to detect potential wash trading patterns		

Solidity contract dataset, Jupyter Notebook, scikit-learn, XGBoost	Build a supervised ML model to classify fraudulent smart contracts
Python, matplotlib, Prophet, Isolation Forest	Detect anomalous wallet behavior using time-series data
Graph Neural Networks (PyTorch Geometric), Elliptic Dataset	Implement a graph-based detection model to identify Ponzi schemes
Apache Kafka, Kibana/Grafana, Logstash, AMLBot API	Visualize real-time fraud alerts from streaming blockchain data

Skill-Building Activities through Experiential Learning:

- Blockchain Fraud Pattern Detection Lab
  Learners download real-world Ethereum transaction data from sources like Etherscan or Kaggle. They engineer relevant features such as transaction frequency and address clustering. Using unsupervised models like Isolation Forest or DBSCAN, participants detect anomalous behaviors, gaining practical skills in identifying fraud patterns in blockchain ecosystems.

- Smart Contract Vulnerability Profiling and Classification
  Participants collect Solidity smart contracts from public repositories like the REKT database. They extract key features such as opcode sequences and gas usage patterns. Using machine learning models like Decision Trees or SVM, learners classify contracts based on their fraud risk, enhancing their ability to detect exploitable code.

- Real-Time Fraud Monitoring Dashboard Development
  Learners develop a real-time fraud monitoring dashboard using streaming blockchain data with Kafka or Web3. They visualize fraud risk scores and suspicious addresses using Kibana or Grafana. Integration with APIs like Chainalysis or AMLBot provides real-time intelligence, simulating how professionals monitor blockchain networks for illicit activities.

Text Book

- M. Rezaul Karim, Artificial Intelligence for Blockchain, Packt Publishing, 2023
- Sudeep Tanwar et al., Blockchain and AI for Cybersecurity and Privacy, Springer, 2022
- Applications of Blockchain and Artificial Intelligence in Finance and Governance. (2024). United States: CRC Press.

References

- Thomas Holt, Cybercrime Through the Blockchain, Routledge, 2022
- Imran Bashir, Mastering Blockchain, Packt Publishing, 2023
- Ashish Mishra, AI in Finance and Blockchain, Wiley, 2022
- Yulia Timofeeva, Detecting Fraud and Financial Crime Using Machine Learning, Springer, 2023

E-Resources/Web Resources

- https://www.chainalysis.com
- https://ciphertrace.com

- https://etherscan.io
- https://elliptic.co/resources
- https://www.geeksforgeeks.org/ethical-hacking/integration-of-blockchain-and-ai/
- https://www.ibm.com/think/topics/blockchain-ai

<table>
<tr><td>Course Code:
CBC3409</td><td colspan="2">Course Name: Optimizing Blockchain Networks with AI
Type of Course: Theory Course</td><td>L- T-P- C</td><td>3-0-0-3</td></tr>
<tr><td>Version No.</td><td colspan="4"></td></tr>
<tr><td>NCrF Level-</td><td colspan="4">5.5</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="4">CAI2500</td></tr>
<tr><td>Co-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="4"></td></tr>
<tr><td>Course Description</td><td colspan="4">This course explores the integration of Artificial Intelligence with blockchain networks to enhance scalability, efficiency, and security. It covers AI-driven optimization of consensus protocols, resource management, anomaly detection, and predictive analytics. Emphasis is placed on real-time monitoring, ethical considerations, and the application of intelligent automation in decentralized environments.</td></tr>
<tr><td>Course Objective</td><td colspan="4"></td></tr>
<tr><td>Course Outcomes</td><td colspan="4">On successful completion of the course the students shall be able to:
• CO1: Describe the architectural components and performance limitations of blockchain networks. (Level 2 – Understand)
• CO2: Summarize the role of AI in addressing scalability, latency, and energy inefficiency issues in distributed ledger systems. (Level 2 – Understand)
• CO3: Examine AI-based models and their effectiveness in optimizing block propagation, transaction throughput, and node behavior. (Level 4 – Analyze)
• CO4: Implement AI techniques such as reinforcement learning and anomaly detection for real-time tuning of blockchain operations. (Level 3 – Apply)
• CO5: Develop intelligent consensus frameworks or predictive modules that enhance security, load balancing, or validator selection. (Level 6 – Create)</td></tr>
<tr><td colspan="5">Course Content:</td></tr>
<tr><td>Module 1</td><td colspan="2">Blockchain Network Architecture & Bottlenecks</td><td>Assessment : Network Mapping & Bottleneck Case Study</td><td>9 hours</td></tr>
<tr><td colspan="5">Topics:
• Node types and peer-to-peer topology (light, full, validator, archival)
• Transaction lifecycle: mempool, nonce, finality, state transitions
• Consensus mechanisms: PoW, PoS, PBFT, DAGs, hybrid models</td></tr>
</table>

- Scalability trilemma: throughput, decentralization, and security trade-offs
- Gas fee dynamics, fee markets (EIP-1559), transaction latency issues
- Multichain architecture and Layer-2 network bottlenecks

Module 2	AI Techniques for Network Optimization	Assessment : AI-Based Simulation Assignment	9 hours

Topics:

- Reinforcement learning for gas optimization and block size tuning
- Congestion prediction using neural networks and time-series models
- Unsupervised learning for node behavior anomaly detection
- AI-assisted block propagation and prioritization strategies
- Graph neural networks (GNNs) for routing optimization
- AI modeling for fork prediction and consensus instability

Module 3	Resource & Energy Optimization in Blockchain Networks	Assessment : Energy Profiling & Optimization Lab	9 hours

Topics:

- Machine learning for energy-aware validator/miner scheduling
- Node clustering and intelligent load balancing algorithms
- Predictive fault detection and real-time node health analytics
- Optimization for resource-constrained environments (e.g., IoT-Blockchain)
- Case studies: Chia, Filecoin, Helium on energy-efficiency metrics
- Integration with green blockchain protocols and frameworks

Module 4	Intelligent Consensus & Federated AI Integration	Assessment : Federated Learning Implementation Exercise	9 hours

Topics:

- AI-enhanced consensus mechanisms (AI-PoS, Proof-of-Reputation)
- Federated learning for decentralized AI model training
- Swarm intelligence and bio-inspired optimization models
- Intent-based consensus protocol adaptation
- Integration of decentralized AI agents (OpenMined, SingularityNET)
- Applications in CBDC optimization, cross-chain DeFi, and supply chain

Module 5	Monitoring, Visualization & Ethical Considerations	Assessment : AI Monitoring Dashboard + Risk Evaluation	9 hours

Topics:

- Real-time blockchain monitoring dashboards (Grafana, Prometheus)
- Stream processing using Apache Kafka for live optimization
- Explainable AI (XAI) for transparency in consensus decision-making

- Ethical challenges: data privacy, model bias, self-evolving protocols
- Governance models for AI in blockchain networks
- Regulatory outlook: GDPR, ISO/IEC 42001 (AI management), AI Act alignment

Targeted Applications & Tools:

Python, Scikit-learn, Pandas, Ethereum Etherscan API	Develop a congestion prediction model using historical transaction data
TensorFlow, OpenAI Gym, Web3.py, Ganache, MetaMask	Optimize gas fees dynamically based on network load using regression or RL models
Isolation Forest (Scikit-learn), Prometheus for node metrics, Grafana	Detect node anomalies in a blockchain testnet using unsupervised learning
RLlib (Ray), SimPy for network simulation, Jupyter Notebook	Implement a reinforcement learning agent to tune block size or interval for PoS chain
ElasticSearch, Kibana, Dockerized Ethereum Testnet (e.g., Geth, Besu)	Visualize blockchain node health and throughput metrics in real-time

Skill-Building Activities through Experiential Learning:

- AI-Powered Gas Fee Estimator Simulation
- Learners extract live Ethereum network data using Web3.py and Etherscan API to train a regression model that estimates optimal gas fees. The model, built with Scikit-learn or XGBoost, is integrated into a simple DApp using Streamlit, helping users predict gas fees based on real-time network congestion.
-
- Reinforcement Learning for Consensus Optimization
- Participants design a reinforcement learning agent using OpenAI Gym to dynamically adjust block intervals in a simulated blockchain environment. The agent optimizes throughput and reduces latency based on transaction load and network conditions, providing hands-on experience with AI-driven consensus improvements.
-
- Real-Time Blockchain Node Monitoring Dashboard
- Learners deploy a monitoring solution using Prometheus and Grafana to visualize the health, transaction throughput, and latency of blockchain nodes on a test network. They configure alerts for node failures or performance bottlenecks, enabling practical understanding of maintaining operational efficiency in blockchain infrastructures.

Text Book

Sudeep Tanwar et al., Blockchain with AI and Machine Learning, Springer, 2023

M. Rezaul Karim, Artificial Intelligence for Blockchain, Packt Publishing, 2023

References

- Imran Bashir, Mastering Blockchain, Packt Publishing, 4th Edition, 2023
- Zheng Zibin, Blockchain Technologies for AI Applications, Springer, 2022
- Md. Sadek Ferdous, AI and Blockchain for Smart Systems, CRC Press, 2022
- Qusay H. Mahmoud, Cognitive Blockchain, Springer, 2021

E-Resources/Web Resources

- https://ethereum.org/en/developers/docs/scaling
- https://deeplearning.ai
- https://github.com/OpenMined
- https://www.ibm.com/think/topics/blockchain-ai
- https://medium.com/@p.noblebose/blockchain-in-machine-learning-ml-and-artificial-intelligence-ai-a-deeper-dive-606107dfc472

Course Code: CBC3410	Course Name: Generative AI for Blockchain Applications Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.			
NCrF Level-	5.5		
Course Pre-requisites	CAI2500		
Co-requisites	NIL		
Anti-requisites			
Course Description	This course explores the integration of Generative AI with blockchain to build decentralized, intelligent applications. It covers LLM-powered smart contracts, AI-generated NFTs, autonomous agents, and content provenance. Emphasis is placed on secure model deployment, ethical considerations, decentralized storage, and real-world use cases involving generative content and AI-token ecosystems.		
Course Objective			
Course Outcomes	On successful completion of the course the students shall be able to: • CO1: Explain the basic concepts of Generative AI and blockchain, and how they can work together in decentralized systems. (Level 2 – Understand) • CO2: Describe how large language models (LLMs) and generative models can be used to create NFTs, smart contract outputs, and decentralized content. (Level 2 – Understand) • CO3: Identify ethical issues related to AI-generated content and implement provenance tracking using blockchain features. (Level 2 – Understand) • CO4: Use blockchain tools and platforms (e.g., Ethereum, IPFS) to store, link, and verify AI-generated content. (Level 3 – Apply) • CO5: Explore simple blockchain applications that incorporate generative AI outputs in smart contracts or NFTs. (Level 3 – Apply)		
Course Content:			
Module 1	Foundations of Generative AI & Blockchain Integration	Assessment : Concept Mapping	9 hours
Topics: • Fundamentals of LLMs (GPT, Claude, LLaMA), GANs, VAEs, Diffusion models • Smart contract architectures and DApps • Prompt engineering for AI-Blockchain workflows			

• Blockchain as audit infrastructure for AI-generated data • On-chain vs off-chain inference pipelines			
Module 2	NFTs, Content Provenance & Creative Automation	Assessment : NFT Metadata Deployment Project	9 hours
Topics: • Generative AI for creating NFT images, music, 3D models • Smart contract logic for auto-minting & licensing (ERC721, ERC1155) • Provenance using IPFS, Arweave, Filecoin • Generative identity and composable NFTs (e.g., AI avatars, AI bots) • AI-authored content: copyright, traceability, and authenticity			
Module 3	Autonomous AI Agents & Decentralized Intelligence Systems	Assessment : DAO Assistant Bot Simulation	9 hours
Topics: • Autonomous agents in DeFi governance (AutoGPT, AgentGPT) • Agent-based interaction with smart contracts (via Chainlink, Gelato) • Prompt markets (e.g., PromptBase) and DAO voting assistants • Staking and trust scores for generative AI agents • AI-token economies and programmable incentives (e.g., TAO, Alethea AI)			
Module 4	Security, Ethics, and Regulatory Infrastructure	Assessment : Generative AI Ethics Policy Brief	9 hours
Topics: • Model authenticity: hashing, checkpoints, and registry using blockchain • Adversarial examples, misuse detection, and AI hallucination risks • Model validation and explainability using blockchain timestamps • Ethics of AI-generated content in Web3: bias, deepfakes, and surveillance • Regulations (e.g., EU AI Act, US AI Bill of Rights, India's Digital Data Bill			
Module 5	Future Directions in Generative AI and Decentralized Systems	Assessment : Proposal Presentation on AI x Blockchain Start-up Idea	9 hours
Topics: • Decentralized AI marketplaces (e.g., Bittensor, Ocean Protocol) • On-chain model training & federated learning using ZK-proofs • Tokenizing compute, data, and intelligence (e.g., GRT, AGIX, FET) • Cross-chain interoperability for AI-generated assets • Case studies: Worldcoin, dGPT, Alethea, Numeraire, SingularityNET			
Targeted Applications & Tools:			
DALL·E / Stable Diffusion, OpenAI API, Ethereum Remix, MetaMask, OpenSea, IPFS		Generate AI-based visual art and mint it as an NFT on Ethereum	

OpenAI GPT API, Solidity, Remix IDE, Hardhat, Web3.js	Create a GPT-powered smart contract assistant that generates Solidity code
Audacity (for audio), IPFS, Arweave, NFT.Storage, Ethereum, Pinata	Store AI-generated music securely and verify ownership using blockchain
ChatGPT API, Chainlink Oracles, Solidity, Moralis, Ether.js	Build a DApp that uses an LLM to summarize DAO proposals and logs them on-chain
Markdown, IPFS, NFT Metadata Standards (ERC-721/1155), Ceramic Network, Fleek	Track provenance and licensing of AI-generated blog posts using verifiable metadata

Skill-Building Activities through Experiential Learning:

- AI-Generated NFT Creation and On-Chain Deployment
- Students will use generative AI tools like DALL·E to create digital artwork and deploy it as NFTs on a blockchain testnet. This activity involves generating images, storing them on IPFS, writing ERC-721 smart contracts, minting NFTs, and verifying them via OpenSea or blockchain explorers.
-
- Integrate GPT with a Smart Contract for DAO Proposal Summarization
- Learners will integrate GPT-generated text into a decentralized governance system. They will generate DAO proposal summaries using GPT, store them on IPFS, and use Chainlink Oracles to link off-chain content to smart contracts for transparent DAO activity logging.
-
- Verifying Ownership and Provenance of AI-Generated Content
- This activity enables students to apply blockchain for authenticity of AI-generated content. They will generate content using GPT-4, hash it, store the hash on a public blockchain, and verify its immutability using block explorers, thus ensuring provenance and ownership tracking.

Text Book

- Rex St. John, Generative AI on Web3: LLMs and Blockchain Integration, Independently Published, 2023
- M. Rezaul Karim, Artificial Intelligence and Blockchain for Beginners, Packt Publishing, 2023
- Kumble, G. P. (2020). Practical Artificial Intelligence and Blockchain: A Guide to Converging Blockchain and AI to Build Smart Applications for New Economies. India: Packt Publishing, Limited.

References

- Imran Bashir, Mastering Blockchain, Packt Publishing, 2023
- Peter Van Hardenberg, Understanding Large Language Models, O'Reilly Media, 2023
- Sudeep Tanwar et al., Blockchain and AI for Cybersecurity and Privacy, Springer, 2022
- OpenAI Research Papers & Alethea Whitepapers

E-Resources/Web Resources

- https://platform.openai.com/docs
- https://ethereum.org/en/developers
- https://ipfs.tech

• https://alethea.ai • https://bittensor.com • https://www.coursera.org/learn/generative-ai-and-blockchain • https://aws.amazon.com/startups/learn/blockchain-and-generative-ai-fueling-innovation-within-the-digital-economy

Course Code: CBC3411	Course Name: Quantum Computing & Blockchain-AI Security Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.			
NCrF Level-	5.5		
Course Pre-requisites	CSD3432		
Co-requisites	NIL		
Anti-requisites	CSEXXXX		
Course Description	This course explores how quantum computing impacts the security of blockchain and AI systems. It covers quantum algorithms, post-quantum cryptography, and hybrid secure architectures. Emphasis is placed on integrating quantum-resilient techniques into smart contracts, consensus protocols, and AI models to build decentralized systems that withstand emerging quantum threats.		
Course Objective			
Course Outcomes	On successful completion of the course the students shall be able to: • CO1: Describe the foundational concepts of quantum computing, including qubits, entanglement, and quantum gates. (Level 2 – Understand) • CO2: Explain how quantum algorithms such as Shor's and Grover's impact existing cryptographic systems used in blockchain and AI. (Level 2 – Understand) • CO3: Examine the vulnerabilities in blockchain consensus protocols and AI model integrity in the context of quantum threats. (Level 4 – Analyze) • CO4: Demonstrate the use of post-quantum cryptographic algorithms within a simulated blockchain environment. (Level 3 – Apply) • CO5: Construct a basic hybrid security architecture that combines blockchain, AI, and quantum-resistant mechanisms. (Level 6 – Create)		
Course Content:			
Module 1	Foundations of Quantum Computing	Assessment : Qiskit Programming Tasks	11 hours
Topics: • Qubits, superposition, entanglement • Quantum gates, circuits, and measurements • Quantum parallelism and interference • Classical vs quantum computational models			

• Hands-on: Quantum programming with Qiskit and Cirq • Introduction to IBM Quantum Lab, Google Cirq Notebooks			
Module 2	Quantum Threats to Blockchain and AI	Assessment : Threat Analysis Report	11 hours
Topics: • Shor's and Grover's algorithms • Breaking RSA/ECC with quantum computing • Vulnerabilities in consensus algorithms (PoW, PoS) • Hash and signature disruption in blockchain • Quantum attacks on model weights, training data poisoning • Risks to zk-SNARKs, ZK-rollups, and oracles			
Module 3	Post-Quantum Cryptography & Blockchain	Assessment : PQC Implementation Task	11 hours
Topics: • NIST-standardized post-quantum algorithms • Lattice-based, hash-based, multivariate, code-based cryptography • Quantum-resistant key exchange and digital signatures • Quantum Key Distribution (QKD) for DLTs • Blockchain adaptation for PQC integration • Hybrid cryptography in Ethereum smart contracts.			
Module 4	Secure Architectures & Contemporary Use Cases	Assessment : Quantum-Safe DApp Design Project	12 hours
Topics: • Designing AI-blockchain systems resilient to quantum threats • Federated learning with blockchain and post-quantum security • Use cases: quantum-secure voting, healthcare AI privacy, quantum supply chains • Quantum-enhanced zero-knowledge proofs • Real-world toolkits: IBM Q, Google Quantum AI, Xanadu PennyLane • Ethical frameworks and compliance in quantum-integrated systems (NIST, GDPR, MiCA)			
Targeted Applications & Tools:			
IBM Qiskit, QASM Simulator		Simulate quantum gates and basic quantum circuits to understand superposition/entanglement	
Qiskit Aqua, Python with NumPy/SymPy		Analyze the vulnerability of RSA using Shor's algorithm in a quantum environment	
CRYSTALS-Kyber Library, Python, Open Quantum Safe (OQS)		Implement a lattice-based encryption scheme (e.g., Kyber) for a mock blockchain message	
GoLang, Python, OpenSSL with PQC Extensions		Integrate post-quantum signatures into a simplified blockchain ledger	

TensorFlow Lite, Scikit-learn, Simulated Ledger System	Design a hybrid consensus model using both AI prediction and quantum-safe voting logic
SimulaQron, Python, QKD Toolkits	Simulate quantum key distribution (QKD) for secure identity verification in smart contracts

Skill-Building Activities through Experiential Learning:

- Quantum Circuit Design and Simulation for Cryptographic Concepts
- Learners design quantum circuits to demonstrate superposition and entanglement. Using Qiskit, they simulate how Shor's algorithm could theoretically break RSA encryption, fostering understanding of quantum impacts on classical cryptography and blockchain key security.
-
- Integrate Post-Quantum Signatures into a Mini Blockchain
- Participants build a basic blockchain ledger in Python and integrate lattice-based post-quantum signatures like Kyber/Dilithium. They test signature verification performance and size, gaining insight into securing blockchain networks against future quantum threats.
-
- AI-Augmented Threat Detection in Quantum-Resistant Blockchain Network
- Students simulate blockchain transaction data with injected anomalies. They train a decision tree model for anomaly detection and evaluate its robustness, understanding how AI can enhance security in quantum-resistant blockchain environments.

Text Book

- Mikio Nakahara & Tetsuo Ohmi, Quantum Computing: An Applied Approach, Springer, 2023
- Mehran Mozaffari, Post-Quantum Blockchain, Springer, 2023

References

- Imran Bashir, Mastering Blockchain, Packt Publishing, 2023
- Scott Aaronson, Quantum Computing Since Democritus, Cambridge University Press, 2022
- Ronald de Wolf, Quantum Computing: Lecture Notes, CWI Amsterdam, 2022
- NIST PQC Project Resources and IBM Q Experience Whitepapers

E-Resources/Web Resources

- https://qiskit.org
- https://quantumai.google
- https://csrc.nist.gov/projects/post-quantum-cryptography
- https://quantum-computing.ibm.com

Course Code: CBC3412	Course Name: Introduction to Decentralized Finance (DeFi)Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	(new version to be added by PU)		

NCrF Level-	5.5
Course Pre-requisites	CBC2000
Co-requisites	NIL
Anti-requisites	
Course Description	This course introduces the foundational and advanced concepts of Decentralized Finance (DeFi), covering smart contracts, lending, DEXs, DAOs, and stablecoins. It includes emerging areas such as DeFi 2.0, real-world asset tokenization, and ZK-rollups, while addressing security risks, governance, and compliance challenges through practical insights and protocol simulations.
Course Objective	
Course Outcomes	On successful completion of the course the students shall be able to: • CO1: Describe the foundational architecture of DeFi systems, including blockchain infrastructure, smart contracts, and token standards. (Level 2 – Understand) • CO2: Explain the roles and functionalities of key DeFi components such as decentralized exchanges, lending platforms, stablecoins, and oracles. (Level 2 – Understand) • CO3: Differentiate between DeFi and traditional finance models in terms of transparency, risk, accessibility, and governance. (Level 4– Analyze) • CO4: Evaluate the design and mechanics of DeFi protocols, identifying security risks, economic vulnerabilities, and compliance challenges. (Level 5 – Evaluate) • CO5: Implement DeFi operations such as liquidity provision, staking, or governance voting using blockchain testnets and smart contracts. (Level 3 – Apply)

Course Content:

Module 1	Foundations of DeFi and Blockchain Infrastructure	Assessment : Concept Review Assignment	9 hours

Topics:

- What is DeFi? DeFi vs CeFi vs TradFi
- Ethereum, EVM, and smart contracts
- Token standards: ERC-20, ERC-721, ERC-4626
- Role of oracles (Chainlink, Band Protocol)
- Wallets: MetaMask, WalletConnect, hardware wallets

Module 2	DeFi Primitives and Protocols	Assessment : Protocol Comparison Report	9 hours

Topics:

- Decentralized Exchanges (Uniswap v3, Curve, Balancer)
- Lending & Borrowing (Aave v3, Compound, Euler)
- Stablecoins: DAI, USDC, FRAX, LUSD, algorithmic models
- Derivatives: Perpetuals, synthetic assets (Synthetix, dYdX)
- Cross-chain DeFi and bridges (LayerZero, Wormhole)

Module 3	Participation, Yield Strategies & DAOs	Assessment : Yield Strategy Lab	9 hours
Topics: • Liquidity provisioning and impermanent loss • Yield farming and staking: manual vs auto-compounding • DAO formation and governance (Snapshot, Aragon) • Token bonding curves, launchpads, IDOs • Simulations using testnets (Goerli, Mumbai, Base Sepolia)			
Module 4	DeFi 2.0, RWA Tokenization & Scaling	Assessment : Case Study Presentation	9 hours
Topics: • DeFi 2.0 concepts: protocol-owned liquidity, OlympusDAO • Real-world asset (RWA) tokenization (Centrifuge, Goldfinch) • ZK-rollups and Layer 2s (zkSync, StarkNet, Arbitrum) • Flash loans and composability • Liquidity abstraction and DeFi middleware			
Module 5	Security, Compliance & Risk Mitigation	Assessment : DeFi Risk Simulation	9 hours
Topics: • Smart contract exploits and audit frameworks (Certik, Code4rena) • Flash loan attacks, rug pulls, governance attacks • DeFi insurance (Nexus Mutual, InsurAce) • Regulatory challenges: FATF, SEC, MiCA, AML/KYC tooling • Security tools: DeFi Score, OpenZeppelin Defender, Slither			
Targeted Applications & Tools:			
Remix IDE, MetaMask, Uniswap V3 Interface, Ethereum Testnet (Goerli/Mumbai), Hardhat		Deploy a basic ERC-20 token and simulate a liquidity pool using Uniswap on a testnet	
Aave Testnet, Compound Testnet, MetaMask, Polygon Mumbai, QuickNode RPC		Participate in staking and yield farming simulations on a DeFi testnet platform	
OpenZeppelin Contracts, Snapshot.org, Aragon SDK, Hardhat, IPFS, Web3.js		Build a DAO voting contract and demonstrate governance proposals and token-based voting	
Aave Flash Loan SDK, Remix IDE, Slither (static analysis), Tenderly, Hardhat		Simulate a flash loan exploit scenario and analyze smart contract vulnerability	
Dune Analytics, The Graph, DeBank API, Etherscan APIs, Python (Dash/Plotly)		Create a dashboard to visualize DeFi protocol metrics such as TVL, APY, and asset flows	
Mythril, Slither, Echidna, Hardhat Security Plugin, OpenZeppelin Defender		Conduct a security audit of a DeFi smart contract to identify rug pull or reentrancy bugs	
Skill-Building Activities through Experiential Learning:			
• DeFi Protocol Simulation on Testnet • Simulate real-world DeFi operations like lending, staking, and token swaps using Ethereum testnets. Deploy a			

custom ERC-20 token, add it to Uniswap liquidity pools, and interact with Aave or Compound to experience APY fluctuations, slippage, and liquidity risks in a safe, test environment.

-
- DAO Governance Role-Play and Smart Contract Voting
- Experience decentralized governance by forming mock DAOs and voting on proposals using token-weighted logic. Utilize platforms like Aragon or Snapshot, write smart contracts for voting, and engage in debate on proposal outcomes, reflecting on governance challenges in decentralized ecosystems.
-
- Smart Contract Audit Simulation and Flash Loan Risk Analysis
- Perform peer-led audits of sample DeFi smart contracts using tools like Slither or Mythril. Identify vulnerabilities, simulate flash loan exploits in a safe test environment, and propose mitigation strategies such as reentrancy protection or time-locks to strengthen DeFi protocol security.

Text Book

Camila Russo, The Infinite Machine, Harper Business, 2022

Lasse Clausen, Mastering DeFi: Decentralized Finance for Everyone, Independently Published, 2023

References

- Imran Bashir, Mastering Blockchain, Packt Publishing, 2023
- Ashwin Ramachandran et al., The DeFi Stack: A Guide to Decentralized Finance, Github Release, 2023
- Rainer Böhme, Cryptoassets and DeFi, Springer, 2022
- Binance Academy & CoinGecko Research Papers

E-Resources/Web Resources

- https://defillama.com
- https://ethereum.org/en/defi
- https://aave.com
- https://uniswap.org
- https://dappradar.com
- https://www.coursera.org/specializations/decentralized-finance-duke#courses

Course Code: CBC3413	Course Name: Blockchain in Financial Services Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	1.0		
Course Pre-requisites	NIL		
Co-requisites	NIL		
Anti-requisites	NIL		

Course Description	This course explores how blockchain technology is transforming the financial services industry, from traditional banking to insurance and capital markets. Topics include decentralized clearing and settlement, digital identity, smart contracts, CBDCs, and regulatory frameworks. Emphasis is placed on architecture, security, scalability, and practical applications.
Course Objective	Understand blockchain fundamentals in the context of financial services Explore real-world applications in payments, lending, trade finance, and capital markets Learn about digital identity, regulatory compliance, and tokenization Analyze challenges of adoption, scalability, and integration with legacy systems
Course Outcomes	**On successful completion of the course the students shall be able to:** CO1 : Explain how blockchain redefines processes in financial services [Level 2 - Understand] CO2 : Evaluate blockchain use cases across banking, insurance, and investment sectors [Level 2 – Analyze] CO3 : Design secure blockchain-based workflows for financial operations [Level 3 – Apply] CO4 : Assess integration, scalability, and regulatory strategies for blockchain solutions in finance [Level 3 – Apply]

Course Content:

Module 1	Blockchain Basics for Finance	Assessment: Assignment	11 Sessions
Topics: Distributed ledgers and consensus, Cryptographic primitives (hashing, digital signatures), Smart contracts, Tokenization of assets, Permissioned vs permissionless blockchains, Architecture of financial blockchain platforms (Corda, Quorum)			
Module 2	Applications in Core Financial Services	Assessment : Assignment	11 Sessions
Topics: Cross-border payments (Ripple, Stellar), Clearing and settlement, Lending & credit scoring (DeFi, tokenized loans), Trade finance (Letter of credit digitization), Digital identity and KYC/AML			
Module 3	Capital Markets, Insurance & Emerging Trends	Assessment : Assignment	11 Sessions
Topics: Tokenized securities and STOs, Fractional ownership, Reinsurance on chain, Parametric insurance, CBDCs and stablecoins, Robo-advisors with blockchain backend, ESG finance with smart contracts			
Module 4	Risk, Regulation & Integration	Assessment : Assignment	12 Sessions
Topics: Security risks (51%, front-running, reentrancy), Regulatory standards (MiCA, FATF), Auditability and traceability, Integration with core banking systems, Blockchain sandboxing, Interoperability (Polkadot, Cosmos)			

Text Book

T1: Antony Welfare, Commercializing Blockchain: Strategic Applications in the Real World, Wiley, 2023

T2: Imran Bashir, Mastering Blockchain, Packt Publishing, 4th Edition, 2023

References

R1: Arvind Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press, 2022
R2: David Shrier, Basic Blockchain, Little Brown, 2022
R3: Ash Costello, Blockchain in Banking and Finance: Innovating with Purpose, Springer, 2022
R4: Deloitte and World Bank Blockchain Reports

E-Resources/Web Resources

1. https://r3.com
2. https://www.weforum.org/projects/blockchain
3. https://www.bis.org
4. https://coincenter.org
5. https://www2.deloitte.com/global/en/pages/financial-services/articles/blockchain-in-banking.html

Course Code: CBC3414	Course Name Building Decentralized Applications (DApps) for Finance Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	(new version to be added by PU)		
NCrF Level-	5.5		
Course Pre-requisites	CBC2000		
Co-requisites	NIL		
Anti-requisites			
Course Description	This course focuses on the design and development of Decentralized Applications (DApps) for financial use cases using blockchain platforms. Students will explore smart contract programming, Web3 integration, decentralized finance protocols, and wallet connectivity. The course emphasizes security, UX design, and compliance in DApp development.		
Course Objective			
Course Outcomes	On completion of this course, students will be able to: • CO1: Define the key components and architecture of decentralized applications used in financial services. (Level 1 – Remember) • CO2: Explain the role of smart contracts, blockchain protocols, and wallets in enabling secure financial DApps. (Level 2 – Understand) • CO3: Interpret the functions of Web3 tools, decentralized storage systems, and token standards in DApp development. (Level 3 – Apply) • CO4: Differentiate between DApp frameworks and evaluate design choices based on financial use case requirements. (Level 4 – Analyze) • CO5: Explore basic smart contract workflows and simulate DApp interactions using testnets and Web3 interfaces. (Level 3 – Apply)		
Course Content:			

Module 1	DApp Architecture & Financial Use Cases	Assessment : Use Case Design Report	9 hours

Topics:

- Components of a DApp (frontend, smart contract, storage, wallets)
- Financial use cases: lending, swaps, insurance, stablecoin vaults
- Blockchain platforms: Ethereum, Polygon, Avalanche
- DApp lifecycle: ideation to deployment
- Comparison: DApps vs Web2 fintech apps

Module 2	Smart Contracts for DApps	Assessment : Concept Review Assignment	9 hours

Topics:

- Solidity programming: inheritance, events, modifiers, data types
- Secure DeFi logic: lending pools, staking contracts, token swaps
- Smart contract upgradeability (proxy patterns)
- Gas optimization techniques (inline assembly, calldata vs memory)
- Security practices: reentrancy, overflow, audit tools (MythX, Slither)

Module 3	Web3 Frontend, Wallets & UX Integration	Assessment : Functional Wallet-Enabled DApp	9 hours

Topics:

- Web3.js, ethers.js, viem for contract interaction
- Wallets: MetaMask, WalletConnect, social login wallets (Web3Auth)
- React.js frontend for DApps, form validation, error handling
- Gasless transactions and account abstraction (ERC-4337)
- UX patterns in financial DApps: onboarding, risk disclosures, alerts

Module 4	Deployment, APIs & Layer-2 Scaling	Assessment : End-to-End DApp Deployment	9 hours

Topics:

- Hardhat & Foundry for testing, deployment, and forked simulation
- Decentralized file storage: IPFS, Arweave
- L2 scaling: Optimism, Arbitrum, zkSync (deployment, bridging)
- DeFi protocol APIs: Aave, Uniswap, 1inch, Chainlink
- Continuous deployment via Alchemy, Infura, Thirdweb

Module 5	Privacy, Compliance & DApp Governance	Assessment : DAO-Enabled Financial DApp Simulation	9 hours

Topics

- zk-SNARKs and zk-rollups for privacy
- Decentralized identity (DID) & verifiable credentials
- Governance via DAOs: token voting, quorum, delegation
- User data compliance: GDPR, MiCA, FATF Travel Rule
- Case studies: Tornado Cash, MakerDAO, Compound Governance

Targeted Application & Tools:

Tools/ Software/ Platform	Application
Solidity, Hardhat, Remix IDE, MetaMask	Develop a smart contract for a decentralized lending platform
React.js, Web3.js, Ethers.js, MetaMask	Create a DApp interface to interact with deployed contracts on Ethereum testnet

IPFS, Arweave, NFT.Storage	Integrate decentralized storage for DApp metadata and documents
WalletConnect, Web3Modal, Infura, Alchemy API	Build a wallet-enabled DApp with real-time balance and transaction updates
Aave Testnet, Uniswap SDK, Chainlink Price Feeds	Simulate staking and yield-farming interaction with existing DeFi protocols
Solidity, OpenZeppelin Governor, Snapshot, Hardhat	Design and test DAO voting logic with smart contracts
Solidity, OpenZeppelin Governor, Snapshot, Hardhat	Design and test DAO voting logic with smart contracts
Slither, MythX, Gas Reporter Plugin for Hardhat	Conduct security audit and gas optimization on a smart contract
Fleek, IPFS, Pinata, Netlify (for static hosting)	Deploy a full-stack DApp and host frontend using decentralized platforms

Skill-Building Activities through Experiential Learning:

AI-Powered Smart Contract Prototype

Design and deploy a smart contract on an Ethereum testnet that interacts with a simple machine learning model for decision-making. An oracle fetches AI model outputs, enabling the smart contract to execute automated actions such as triggering payments or trades based on AI predictions.

Decentralized Federated Learning Simulation

Simulate federated learning where multiple nodes train AI models locally and share encrypted updates via blockchain. This demonstrates how blockchain enhances data privacy and coordination in distributed AI training without central data sharing, highlighting the synergy of decentralized technologies for secure collaborative learning.

Blockchain-Based AI Model Marketplace

Build a decentralized application where users can upload, discover, and access pre-trained AI models using blockchain transactions. Payments are made with test tokens, and model details are immutably recorded on-chain. This introduces students to tokenized AI asset exchange and decentralized infrastructure for AI services.

Text Book

T1: Arshdeep Bahga & Vijay Madisetti, Blockchain Applications: A Hands-On Approach, VPT, 2023

T2: Mohammad Rezaul Karim, AI and Blockchain for Beginners, Packt Publishing, 2023

T3: Artificial Intelligence and Blockchain for Future Cybersecurity Applications. (2021). Germany: Springer International Publishing.

References

R1: Imran Bashir, Mastering Blockchain, Packt Publishing, 2023

R2: Kai-Fu Lee, AI Superpowers: China, Silicon Valley, and the New World Order, HMH, 2022

R3: Bhaskar Krishnamachari, Blockchain and AI for Internet of Things, Springer, 2022

R4: Jaspreet Bindra, The Tech Whisperer: On Digital Transformation and the Technologies That Enable It, Penguin, 2021

E-Resources/Web Resources

1. https://ai.google
2. https://ethereum.org/en/developers/docs/oracles
3. https://deeplearning.ai
4. https://oceanprotocol.com
5. https://www.ibm.com/think/topics/blockchain-ai
6. https://aws.amazon.com/startups/learn/blockchain-and-generative-ai-fueling-innovation-within-the-digital-economy

Course Code: CBC3415	Course Name Smart Contracts for Financial Products Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	(new version to be added by PU)		
NCrF Level-	5.5		
Course Pre-requisites	CBC2500		
Co-requisites	NIL		
Anti-requisites			
Course Description	This course focuses on designing, developing, and securing smart contracts for financial applications using platforms like Ethereum and Hyperledger. It covers programmable finance, tokenized assets, DeFi mechanisms, contract auditing, and compliance frameworks. Emphasis is placed on real-world financial workflows, gas optimization, and secure deployment of automated financial agreements.		
Course Objective			
Course Outcomes	On completion of this course, students will be able to: • CO1: Describe the architecture, components, and operational lifecycle of smart contracts in blockchain-based financial systems. (Level 2 – Understand) • CO2: Explain the use of smart contracts in modeling various financial instruments such as loans, escrows, and insurance. (Level 2 – Understand) • CO3: Compare different smart contract platforms and frameworks in terms of suitability for financial product development. (Level 4 – Analyze) • CO4: Develop and deploy basic smart contracts for financial services using Solidity and appropriate blockchain tools. (Level 6 – Create) • CO5: Use auditing tools and frameworks to identify vulnerabilities and improve contract performance and compliance. (Level 3 – Apply)		
Course Content:			
Module 1	Foundations of Smart Contracts in Finance	Assessment : Concept Mapping and Oral Quiz	9 hours
Topics: • Smart contract fundamentals: execution logic, triggers, and states • Ethereum Virtual Machine (EVM), gas and state transitions • Overview of Solidity, Vyper, and DAML • Use of Oracles (Chainlink, API3) for off-chain data • Smart contract design patterns and financial use case architecture • Intro to smart legal contracts (Ricardian contracts)			

Module 2	Modeling and Implementing Financial Instruments	Assessment : Mini Project – Loan or Insurance DApp	9 hours
Topics: • Parametric and usage-based insurance contracts • Tokenized bonds, RWA (Real-World Assets), and fractional ownership • Flash loans, collateralized lending, and interest-bearing tokens • Use of ERC standards: ERC-20, ERC-721, ERC-4626 (Tokenized Vaults) • Design of structured products and yield aggregators • Digital escrow contracts with milestone-based payouts			
Module 3	Development, Testing & Deployment	Assessment : Contract Deployment Report	9 hours
Topics: • Writing secure contracts using Solidity and OpenZeppelin libraries • Contract inheritance, modifiers, and access control • Deployment using Truffle, Hardhat, Foundry • Testing strategies: unit testing, simulation testing, gas profiling • Deployment on testnets: Goerli, Mumbai, Sepolia • Integrating with frontends using Ethers.js/Web3.js			
Module 4	Security, Audit Practices & Risk Management	Assessment : Vulnerability Report Submission	9 hours
Topics: • Common vulnerabilities: reentrancy, overflow, DoS with gas limit • Tools: Slither, Mythril, Echidna, Tenderly • Secure contract upgradeability using proxy patterns • Implementing role-based access control and pause mechanisms • Attack case studies: The DAO, Poly Network, bZx • Insurance for smart contract risk (e.g., Nexus Mutual)			
Module 5	Compliance, Token Standards & Emerging Trends	Assessment : DAO-Enabled Financial DApp Simulation	9 hours
Topics • MiCA, FATF, and SEC regulatory frameworks • On-chain compliance: KYC whitelisting, blacklisting, time-locked tokens • Zero-knowledge proof-based compliance enforcement (ZK-KYC) • Role of Layer-2 (Optimism, zkSync) and rollups in financial DApps • Hybrid contract execution with off-chain compute (Chainlink Functions, Gelato) • Future of programmable finance and AI-assisted smart contracts			
Targeted Application & Tools:			
Solidity, Remix IDE, MetaMask, Hardhat	Develop a smart contract for a collateralized loan with repayment logic		
Solidity, Ganache, Truffle Suite, MetaMask	Create and deploy an escrow smart contract on a blockchain test network		
Mythril, Slither, OpenZeppelin library, Remix	Audit an existing smart contract for vulnerabilities such as reentrancy		

Solidity, OpenZeppelin Contracts, Remix, Etherscan Testnet	Implement and test a tokenized bond using ERC-20 or ERC-1400 standard

Skill-Building Activities through Experiential Learning:

- Smart Contract Development for Escrow Services
  Participants will design, code, and deploy a smart contract that functions as an automated escrow service for financial transactions. The contract will hold funds, release them upon fulfillment of conditions, and handle disputes. Deployment will be done on a blockchain test network.
- Simulation of Tokenized Financial Instruments
  A practical exercise to create and deploy tokenized bonds or equity contracts using ERC-20 or ERC-1400 standards. Participants will simulate ownership transfer and trading of these tokenized assets on a testnet environment, integrating smart contract logic for regulatory compliance.
- Smart Contract Security Audit Workshop
  Participants will perform a guided security audit of a deployed financial smart contract. Common vulnerabilities such as reentrancy, integer overflow, and access control flaws will be identified using automated tools, followed by applying best-practice fixes.

Text Book

- Ritesh Modi, Building Smart Contracts and DApps on Ethereum, BPB Publications, 2023
- Imran Bashir, Mastering Blockchain, Packt Publishing, 4th Edition, 2023

References

- Andreas M. Antonopoulos & Gavin Wood, Mastering Ethereum, O'Reilly Media, 2023
- Chris Dannen, Introducing Ethereum and Solidity, Apress, 2022
- S. Seijas et al., Financial Smart Contracts and the Future of Banking, Springer, 2021
- OpenZeppelin Docs and Solidity Audit Guides

E-Resources/Web Resources

7. https://soliditylang.org
8. https://remix.ethereum.org
9. https://openzeppelin.com
10. https://consensys.net
11. https://chain.link
12. https://www.coursera.org/learn/smarter-contracts

Course Code: CBC3416	Course Name: Decentralized Autonomous Organizations and Risk Management in Finance Type of Course: Theory Course / Laboratory Course	L- T-P- C	3-0-0-3
Version No.	(new version to be added by PU)		
NCrF Level-	5.5		
Course Pre-requisites	CSE2000		
Co-requisites	NIL		
Anti-requisites			
Course Description	This course provides an in-depth understanding of Decentralized Autonomous Organizations (DAOs) and their role in reshaping financial services. It covers DAO governance models, treasury operations, risk management, legal frameworks, and emerging trends such as AI-powered DAOs and cross-chain collaborations, with a focus on decentralized finance, compliance, and security challenges.		

Course Objective			
Course Outcomes	On completion of this course, students will be able to: • CO1: Describe the structure, governance models, and operational principles of DAOs. (Level 2 – Understand) • CO2: Explain various use cases of DAOs in financial services such as lending, investment, and treasury management. (Level 2 – Understand) • CO3: Examine the risks associated with DAO-managed financial operations, including smart contract, liquidity, and governance risks. (Level 4 – Analyze) • CO4: Differentiate between DAO platforms and evaluate their applicability for specific financial services. (Level 4 – Analyze) • CO5: Build and simulate basic DAO smart contracts, voting systems, and treasury operations using blockchain tools. (Level 6 – Create)		
Course Content:			
Module 1	DAO Fundamentals, Evolution & Governance	Assessment : Conceptual Case Review	9 Hours
Topics: • DAO definitions, origins, and evolution (DAOs, Meta-DAOs, SubDAOs) • DAO structure and operational logic • Governance models: token-based, quadratic voting, conviction voting, delegated governance • DAO platforms and tooling: Aragon, DAOstack, Safe (Gnosis Safe), Snapshot • Role of smart contracts, treasury operations, and proposal systems • Trends: Reputation-based governance, Soulbound tokens (SBTs)			
Module 2	Financial Applications and DAO Ecosystem Use Cases	Assessment : DAO Use Case Presentation	9 Hours
Topics: • Investment DAOs, Lending platforms, and Yield aggregators • DAO-managed stablecoins and algorithmic treasury mechanisms • DAO for insurance, reinsurance, and risk pooling • Cross-chain DAOs and DAO2DAO collaborations • Real-world examples: MakerDAO, Yearn Finance, ConstitutionDAO, KlimaDAO • Impact of DAOs in DeFi, ESG finance, and philanthropic finance			
Module 3	Risk Management in DAO Ecosystems	Assessment : Risk Scenario Simulation	9 Hours
Topics: • Categories of risk: protocol, market, operational, governance, regulatory • DAO treasury risk modeling and diversification • Smart contract vulnerabilities: flash loan attacks, reentrancy, oracle manipulation			

- Governance attacks: Sybil attacks, whale dominance, collusion risks
- Contemporary mitigation tools: OpenZeppelin Defender, Hats Finance, DAO insurance protocols
- Risk scoring models and community-driven defense mechanisms.

Module 4	Legal, Regulatory & Compliance Considerations	Assessment : Regulatory Framework Analysis	9 Hours

Topics:

- Legal entity status of DAOs: Wyoming DAO LLCs, Liechtenstein DAO framework
- Global regulatory landscape: SEC, MiCA, FATF, evolving DAO recognition
- Compliance tools: automated KYC, treasury transparency protocols, real-time auditability
- Legal templates: LexDAO resources, decentralized arbitration mechanisms
- Token classification: security vs utility tokens in DAO ecosystems
- Case studies: Legal precedents and jurisdictional challenges

Module 5	Emerging Trends and Future of DAO-Driven Finance	Assessment : Group Project: DAO for Financial Use Case	9 Hours

Topics:

- Autonomous and AI-powered DAOs
- DAO integrations with DeFi 2.0 protocols
- Treasury automation using AI and Machine Learning
- Interoperability and Cross-chain DAO frameworks (Cosmos, Polkadot, Chainlink CCIP)
- DAO infrastructure innovations: governance NFTs, reputation DAOs, DAO tooling ecosystems
- Ethical considerations, inclusive governance, and future of decentralized governance in finance

Targeted Applications & Tools:

Tools/Software/Platform	Application
Aragon, MetaMask, Ethereum Testnet (Goerli), Hardhat	Deploy a basic DAO with treasury management and token-based voting on a testnet.
DAOstack, OpenZeppelin Defender, Remix IDE	Simulate a DAO governance attack (e.g., Sybil or token concentration) and apply preventive measures.
MakerDAO, DeFiLlama, Dune Analytics	Conduct a risk assessment for a DAO-managed stablecoin or lending protocol.
Solidity, Hardhat, OpenZeppelin Contracts	Design and test smart contract-based insurance logic within a DAO framework.
LexDAO Legal Templates, Wyoming DAO Toolkit, MakerDAO Docs	Evaluate regulatory risks for DAO operations based on global jurisdiction analysis.

Skill-Building Activities through Experiential Learning:

- End-to-End DAO Deployment and Treasury Governance

Deploy a functional DAO on an Ethereum testnet using platforms like Aragon. Design its structure, configure governance tokens, and set up treasury management. Create and execute fund allocation proposals to simulate DAO governance, enhancing technical understanding of decentralized finance and on-chain decision-making processes.

- Risk Assessment and Mitigation Planning for a DAO Protocol

Conduct a risk assessment for an existing DAO such as MakerDAO or Yearn Finance using tools like DeFiLlama and Dune Analytics. Analyze smart contract vulnerabilities, liquidity risks, and governance issues. Propose mitigation strategies including treasury diversification, improved voting mechanisms, and smart contract security enhancements.

- Smart Contract Vulnerability Identification and Governance Attack Simulation

Simulate a governance attack such as token accumulation or flash loan exploits using Remix IDE and Hardhat. Identify vulnerabilities in DAO voting logic, execute the attack in a test environment, and implement preventive measures like quorum requirements, time delays, or multi-signature controls to enhance DAO governance security.

Text Book

- Oriol Caudevilla et al., Decentralized Finance and DAOs, Springer, 2023
- Linda Xie, A Beginner's Guide to DAOs, Ethereum Foundation Series, 2023
- Decentralized Autonomous Organizations (DAO): Innovating in the New Digital Economy. (2024). (n.p.): Hernán Cornejo.

References

- Imran Bashir, Mastering Blockchain, Packt Publishing, 2023
- Nathan Schneider, Exit to Community: DAO Governance and the Future of Finance, Yale University Press, 2022
- Primavera De Filippi et al., Blockchain and the Law: The Rule of Code, Harvard Press, 2022
- DAO Research Collective Whitepapers & LexDAO Legal Templates

E-Resources/Web Resources

- https://daostack.io
- https://aragon.org
- https://ethereum.org/en/dao
- https://makerdao.com
- https://lexdao.substack.com

Course Code: CBC3417	Course Name: Regulatory and Compliance Challenges in DeFi Type of Course: Theory Course	L- T-P- C	3-0-0-3
Version No.	(new version to be added by PU)		
NCrF Level-	5.5		
Course Pre-requisites	FIN1001		
Co-requisites	NIL		
Anti-requisites			

<table>
<tr><td>Course Description</td><td colspan="3">This course offers an in-depth understanding of regulatory, legal, and compliance challenges in Decentralized Finance (DeFi). It covers global frameworks, token classification, enforcement trends, and technical solutions like on-chain KYC, decentralized identity, and risk monitoring. Emphasis is placed on DAO governance, privacy, ethical considerations, and emerging regulated DeFi models.</td></tr>
<tr><td>Course Objective</td><td colspan="3"></td></tr>
<tr><td>Course Outcomes</td><td colspan="3">On completion of this course, students will be able to:
CO1: Describe the key regulatory bodies, frameworks, and principles governing DeFi ecosystems. (Level 2 – Understand)
CO2: Explain the legal challenges posed by decentralized platforms, anonymity, and token classifications. (Level 2 – Understand)
CO3: Differentiate between global regulatory approaches, including MiCA, FATF guidelines, and national compliance models. (Level 4 – Analyze)
CO4: Examine risks, enforcement actions, and compliance gaps in real-world DeFi platforms and case studies. (Level 4 – Analyze)
CO5: Assess the ethical and governance implications of integrating compliance into decentralized financial systems. (Level 3 – Apply)</td></tr>
<tr><td colspan="4">Course Content:</td></tr>
<tr><td>Module 1</td><td>DeFi Regulation Landscape & Emerging Concerns</td><td>Assessment : Case Analysis</td><td>9 hours</td></tr>
<tr><td colspan="4">Topics:
DeFi architecture and regulatory blind spots
Global regulators: FATF, SEC, MiCA (EU), BIS, FSMA (Belgium)
DeFi vs CeFi vs TradFi regulatory approaches
Pseudonymity, privacy-enhancing technologies (e.g., Tornado Cash), and legal challenges
Rise of Regulated DeFi (RegDeFi) and embedded compliance</td></tr>
<tr><td>Module 2</td><td>Token Classification, Jurisdiction & Enforcement</td><td>Assessment : Compliance Mapping Exercise</td><td>9 hours</td></tr>
<tr><td colspan="4">Topics:
Security, utility, payment, and hybrid token classifications
Stablecoin regulation (EU, USA, Singapore)
Jurisdictional complexity in decentralized protocols
Case studies: Ripple (XRP) lawsuit, TerraUSD collapse
Global regulatory sandbox initiatives and experimental compliance frameworks
Digital escrow contracts with milestone-based payouts</td></tr>
<tr><td>Module 3</td><td>Compliance by Design & Technical Solutions</td><td>Assessment : On-Chain Compliance Demonstration</td><td>9 hours</td></tr>
<tr><td colspan="4">Topics:
On-chain KYC/AML models (Soulbound tokens, zero-knowledge proofs)
Decentralized Identity (DID), Verifiable Credentials (VCs) for DeFi access control
Compliance tooling: Chainalysis, TRM Labs, Solidus Labs
Smart contract-based risk scoring and sanctions screening
Oracle-integrated compliance feeds and real-time transaction monitoring</td></tr>
<tr><td>Module 4</td><td>DAO Governance, Risk & Regulatory Implications</td><td>Assessment : DAO Risk Simulation</td><td>9 hours</td></tr>
</table>

Topics:

- Legal status of DAOs (Wyoming, Malta, Liechtenstein)
- Governance risks: voter collusion, Sybil attacks, flash loan governance exploits
- DAO treasury management and compliance obligations
- Real-world DAO enforcement (CFTC action against Ooki DAO)
- Regulated DAO frameworks and legal entity wrappers

Module 5	Ethics, Privacy & Future of DeFi Compliance	Assessment : Regulatory Impact Debat	9 hours

Topics

- Privacy vs compliance: zero-knowledge DeFi, confidential transactions
- Data protection (GDPR, CPRA) in DeFi ecosystems
- Ethical dilemmas: censorship, DeFi access restrictions, permissioned DeFi
- Rise of Self-Regulatory Organizations (SROs) in DeFi
- Future trends: algorithmic compliance, AI-powered risk detection, global convergence of DeFi regulation

Targeted Application & Tools:

Uniswap Interface, Token Sniffer, Etherscan	Analyze a real-world DeFi platform (e.g., Uniswap) for regulatory risks related to anonymity, token classification, and KYC gaps.
Chainalysis KYT Demo, TRM Labs Sandbox, Goerli Testnet	Explore and compare on-chain AML/Compliance monitoring by simulating suspicious transactions using a testnet.
EU MiCA Documents, Aave Testnet, Regulatory Sandboxes	Conduct a policy analysis of the MiCA framework and map its requirements onto an existing DeFi lending protocol.
Remix IDE, Solidity Static Analysis Tools (Mythril, Slither)	Examine smart contract code for potential compliance-related vulnerabilities (e.g., missing KYC/AML controls).
MetaMask, SelfKey DID tools, Ceramic Network	Evaluate the role of decentralized identity in KYC using a DID-enabled wallet simulation.

Skill-Building Activities through Experiential Learning:

Simulated Compliance Audit of a DeFi Protocol

- Participants will perform a simulated compliance audit of a DeFi protocol such as Aave or Uniswap. They will map the protocol's design against regulatory expectations like KYC/AML, token classification, and user privacy. Findings will be presented as a compliance gap report along with proposed mitigation strategies.

On-Chain Risk Detection and Transaction Monitoring Exercise

- Participants will simulate suspicious transactions using testnet tokens on a DeFi platform and apply real-time transaction monitoring tools. They will identify potential money laundering patterns or risky wallet behaviours and generate alerts using AML compliance engines.

Design of a DeFi Compliance-by-Design Workflow

- Participants will conceptualize a compliance-by-design model for a new DeFi platform. They will incorporate elements such as decentralized identity (DID), oracle-based compliance feeds, and automated smart contract restrictions. The focus will be on balancing decentralization with regulatory expectations.

Text Book

- Jason Gottlieb, Crypto Regulation: Navigating Compliance in the Decentralized World, Wiley, 2023
- Christopher Giancarlo, CryptoDad: The Fight for the Future of Money, Wiley, 2023

References

- Primavera De Filippi & Aaron Wright, Blockchain and the Law, Harvard University Press, 2022

- Imran Bashir, Mastering Blockchain, Packt Publishing, 2023
- World Economic Forum, DeFi Policy Toolkit, WEF, 2022
- Global Financial Stability Reports – IMF, BIS, FATF Recommendations

E-Resources/Web Resources

- https://www.fatf-gafi.org
- https://www.sec.gov/spotlight/cybersecurity
- https://www.chainalysis.com
- https://trmlabs.com
- https://www.coursera.org/learn/decentralized-finance-opportunities-and-risk-duke

Course Code: CSS3426	Course Name: Front-end Full Stack Development Course Type : Lab Integrated	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSE2258		
Co-requisites			
Anti-requisites	NIL		
Course Description	This intermediate course enables students to perform front-end full stack development, with emphasis on employability skills. The course covers key technologies and architectures that enables the student to design and implement front-end. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.		
Course Objectives	The objective of the course is to familiarize the learners with the concepts Front end Full Stack Development and attain Employability through experiential Learning techniques.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1 Describe the fundamentals of DevOps and Front-end Full Stack Development. — Level 2 – Comprehension (Understanding) CO2 Illustrate a basic web design using HTML, CSS, and JavaScript. — Level 3 – Application CO3 Illustrate the development of a responsive web application. — Level 3 – Application CO4 Apply the concepts of Angular.js to develop a web front-end application. — Level 3 – Application CO5 Develop and deploy a complete front-end web application by integrating DevOps practices and modern web technologies. — Level 3 – Application		
Course Content:			
Module 1	Fundamentals of DevOps	Assessment: Project	04 Sessions
Topics: Introduction to Agile Methodology; Scrum Fundamentals; Scrum Roles, Artifacts and Rituals; DevOps – Architecture, Lifecycle, Workflow & Principles; DevOps Tools Overview – Jenkins, Docker, Kubernetes. Review of GIT source control.			
Module 2	Web Design & Development	Assessment: Project	03 Sessions
Topics:			

HTML5 – Syntax, Attributes, Events, Web Forms 2.0, Web Storage, Canvas, Web Sockets; CSS3 – Colors, Gradients, Text, Transform;

Assignment: Develop a website for managing HR policies of a department.

Module 3	Responsive web design	Assessment: Project	08 Sessions

Topics:

BootStrap for Responsive Web Design; JavaScript – Core syntax, HTML DOM, objects, classes, Async; Ajax and jQuery Introduction

Assignment: Design and develop a website that can actively keep track of entry-exit information of a housing society.

Module 4	Fundamentals of Angular.js	Project	Programming	15 Sessions

Topics:

Setting up Development & Build Environment: Node.js and NPM; Introduction to TypeScript; Working with OOP concepts with TypeScript; Angular Fundamentals; Angular CLI; Introduction to TypeScript; Debugging Angular applications; Components & Databinding in Depth; Angular Directives; Using Services & Dependency Injection; Angular Routing; Observables; Handling Forms in Angular Apps; Output transformation using Pipes; Making Http Requests; Authentication & Route Protection; Dynamic Components; Angular Modules & Optimizing Angular Apps; Deploying an Angular App; Angular Animations; Adding Offline Capabilities with Service Workers; Unit Testing in Angular Apps (Jasmine, Karma). Overview of React.js

Assignment: Develop a software tool to do inventory management in a warehouse.

Targeted Application & Tools that can be used:

GCC compiler.	An open-source compiler that converts source code written in languages like C and C++ into executable machine code, providing error checking and optimization support.

Text Book:

T1. Fender, Young, "Front-end Fundamentals", Leanpub, 2015

T2. Northwood, Chris, "The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer", APress, 2018

References:

R1. Flanagan D S, "Javascript : The Definitive Guide" 7th Edition. 7th ed. O'Reilly Media; 2020.

R2. Alex Libby, Gaurav Gupta, and Asoj Talesra. "Responsive Web Design with HTML5 and CSS3 Essentials", Packt Publishing, 2016

R3. Duckett J Ruppert G Moore J. "Javascript & Jquery : Interactive Front-End Web Development."; Wiley; 2014.

R4. Web Reference:

https://www.youtube.com/watch?v=JGNTYXkVCVY&list=PLd3UqWTnYXOkTSBCBNyyhxo_jxlY_uTWA&index=2

R5. Web Reference: https://www.freecodecamp.org/news/frontend-web-developer-bootcamp/

https://puniversity.informaticsglobal.com:2229/login.aspx?direct=true&db=nlebk&AN=2233842&site=ehost-live

https://nptel.ac.in/courses/106102064

Topics relevant to development of "Employability": DevOps Tools Overview – Jenkins, Docker, Kubernetes for developing Employability Skills through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CSS3427	Course Name: Java Full Stack Development Course Type: Lab Integrated	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSE2258		
Co-Requisite	NIL		
Anti-requisites	NIL		
Course Description	This advanced level course enables students to perform full stack development using Java, with emphasis on employability skills. The key technologies used for Full Stack development is based on either Java technology or .NET technology. In this course, the focus is on using Java, and the related technologies/tools like Java EE, Java Persistence, Hibernate, Maven, Spring Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.		
Course Objectives	This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Practice the use of Java for Full Stack Development. — Level 3 – Apply CO2: Develop web applications using Java EE. — Level 3 – Apply CO3: Solve application development problems using Java Persistence and Hibernate. — Level 3 – Apply CO4: Apply Spring Framework concepts to develop Full Stack applications. — Level 3 – Apply CO5: Employ automation tools such as Maven and Selenium for Full Stack Development. — Level 3 – Apply		
Course Content:			
Module 1	Introduction	Assesment:Project	03 Sessions
Topics: Review of Java; Advanced concepts of Java; Java generics; Java IO; New Features of Java. Unit Testing tools.			
Module 2	Java EE Web Applications	Assesment: Project	05 Sessions
Topics: Introduction to Eclipse & Tomcat; JSP Fundamentals; Reading HTML form Data with JSP; State Management with JSP; JSP Standard Tag Library - Core & Function Tags; Servlet API Fundamentals; ServletContext, Session, Cookies; Request Redirection Techniques; Building MVC App with Servlets & JSP; Complete App - Integrating JDBC with MVC App Assignment: Develop an application for managing HR policies of a department.			

Module 3	Java Persistence using JPA and Hibernate	Assesment: Project	06 Sessions

Topics:

Fundamentals of Java Persistence with Hibernate; JPA for Object/Relational Mapping, Querying, Caching, Performance and Concurrency; First & Second Level Caching, Batch Fetching, Optimistic Locking & Versioning; Entity Relationships, Inheritance Mapping & Polymorphic Queries; Querying database using JPQL and Criteria API (JPA)

Assignment: Design and develop a website that can actively keep track of entry-exit information of a housing society.

Module 4	Spring Core	Assesment: Project	10 Sessions

Topics:

Spring Core, Spring MVC, Spring Boot REST API; Understanding Spring Framework; Using Spring MVC; Building a Database Web App with Spring and Hibernate o Spring AOP (Aspect Oriented Programming); Implementing Spring Security; Developing Spring REST API; Using Spring Boot for Rapid Development

Assignment: Develop a software tool to do inventory management in a warehouse.

Module 5	Automation tools	Assesment: Project	06 Sessions

Topics:

Introduction to Automation Tools; Apache Maven: Maven Fundamentals, Software Setup - Commandline and Eclipse, pom.xml and Directory Structure, Multi-Module Project Creation, Scopes, Dependency Management, Profiles; Functional/BDD Testing using Selenium, Selenium Fundamentals and IDE, Selenium WebDriver, Installation and Configuration, Locating WebElements, Driver Commands, WebElement Commands

Assignment: Illustrate the use of automation tools in the development of a small software project.

Targeted Application & Tools:

Eclipse	An open-source Integrated Development Environment (IDE) widely used for Java application development.
NetBeans	A free IDE that supports Java, web, and enterprise application development with built-in development tools.
Hibernate	An Object-Relational Mapping (ORM) framework that simplifies database operations in Java applications.
Selenium	An automation testing tool used for testing web applications across different browsers and platforms.
Maven	A project management and build automation tool used for dependency management and project lifecycle management in Java projects.
Git	A distributed version control system used to track changes in source code and support collaborative software development.

Project work/Assignment:

Problem Solving: Design of Algorithms and implementation of programs.

Programming: Implementation of given scenario using Java.

Text Book:

T1: Fender, Young, “Front-end Fundamentals”, Leanpub, 2015

References

R1. Soni, Ravi Kant. “Full Stack AngularJS for Java Developers: Build a Full-Featured Web Application from Scratch Using AngularJS with Spring RESTful.” , Apress, 2017.

R2. Mardan, Azat. “Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB.”, Apress, 2015

Course Code: CSS3428	Course Name: .NET Full Stack Development Course Type: Lab Integrated	L- T-P- C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	CSE2258		
Co-Requisite	NIL		
Anti-requisites	NIL		
Course Description	This advanced level course enables students to perform full stack development using .NET, with emphasis on employability skills. The key technologies used for Full Stack development is based on either Java technology or .NET technology. In this course, the focus is on using .NET and the related technologies/tools like C#, ASP.NET, Entity Framework Core, etc. On successful completion of this course, the student shall be able to pursue a career in full-stack development. The students shall develop strong problem-solving skills as part of this course.		
Course Objectives	This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.		
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Practice the use of C# for developing small applications. — Level 3 – Apply CO2: Develop web applications using Entity Framework. — Level 3 – Apply CO3: Solve simple web application problems using SQL and ASP.NET. — Level 3 – Apply CO4: Apply ASP.NET concepts to develop Full Stack applications. — Level 3 – Apply CO5: Employ debugging, testing, and deployment techniques for developing and maintaining ASP.NET applications. — Level 3 – Apply		
Course Content:			
Module 1	C# Programming for Full Stack Development	Assesment:Project	10 Sessions
	Topics: .NET Framework Fundamentals, Visual Studio IDE Fundamentals, C# Language Features, Working with arrays and collections, Working with variables, operators, and expressions, Decision and iteration statements, Managing program flow and events, Working with classes and methods, OOP concepts, Properties, Auto Implemented, Delegates, Anonymous Methods and Anonymous Types, Extension methods, Sealed Classes/Methods, Partial Classes/Methods, Asynchronous programming and threading, Data validation and working with data collections including LINQ, Handling errors and exceptions, Working with Files, Unit Testing – Nunit framework Assignment: Develop a small application for managing library using C#.		
Module 2	Entity Framework Core 2.0	Assesment:Project	06 Sessions
	Topics: Entity Framework Core 2.0 Code First Approach; Introduction To Entity Framework and EDM; Querying the EDM; Working With Stored Procedures; Advanced Entity Framework - DbContext [EF6]; Advanced Operations; Performance Optimization; Data Access with ADO.NET Assignment: Develop an application for managing HR policies of a department.		

<table>
<tr><td>Module 3</td><td>ASP.NET</td><td>Assesment:Project</td><td>06 Sessions</td></tr>
<tr><td></td><td colspan="3">Topics:

ASP.NET Core, ASP.Net Core 3.1 MVC, ASP.NET Core Middleware and Request pipeline, Review of SQL using MS SQL, Working With Data In Asp.Net, Razor View Engine, State Management In Asp. Net MVC & Layouts;

Assignment: Develop a web application to mark entry/exit of guests in a building.</td></tr>
<tr><td>Module 4</td><td>ASP.NET</td><td>Assesment:Project</td><td>08 Sessions</td></tr>
<tr><td></td><td colspan="3">Topics:

Introduction To Models, Validations In Asp.Net MVC, Authentication and Authorization In Asp.Net MVC, Advanced Asp. Net MVC - Ajax Action Link In MVC, Advanced Asp.Net MVC - Ajax Forms In MVC, Microsoft Testing Framework – Unit Testing the .NET Application

Assignment: Develop a software tool to do inventory management in a warehouse.</td></tr>
<tr><td></td><td colspan="3">Targeted Application & Tools that can be used:
<table><tr><td>Visual Studio</td><td>A comprehensive IDE for developing, debugging, testing, and deploying .NET applications.</td></tr><tr><td>SQL Server</td><td>A relational database management system used to store and manage application data.</td></tr></table></td></tr>
<tr><td></td><td colspan="3">Project work/Assignment:</td></tr>
<tr><td></td><td colspan="3">Problem Solving: Design of Algorithms and implementation of programs.

Programming: Implementation of given scenario using .NET.</td></tr>
<tr><td></td><td colspan="3">Text Book:

T1. Fender, Young, “Front-end Fundamentals”, Leanpub, 2015

T2. Valerio De Sanctis, “ASP.NET Core 5 and Angular: Full-stack web development with .NET 5 and Angular 11”, 4th Edition, Packt, 2021.</td></tr>
<tr><td></td><td colspan="3">References

R1. Benjamin Perkins, Jon D. Reid, “Beginning C# and .NET”, Wiley, 2021 Reid, 2021.

R2. Piotr Gankiewicz, “Full Stack .NET Web Development”, Packt Publishing, 2017.

R3. Tamir Dresher, Amir Zuker, Shay Friedman, “Hands-On Full-Stack Web Development with ASP.NET Core”, Packt Publishing, 2018.

R4. Dustin Metzgar, “Exploring .NET core with microservices, ASP.NET core, and Entity Framework Core”, Manning, 2017.</td></tr>
</table>

*Course Code: CSS3502	Course Title: Distributed Ledger Architectures and Smart Contract Systems Type of Course: PCC– Theory & Practical	L-T- P- C	2	0	2	3

Version No.	1.0				
Course Prerequisites	CSEXXXX				
Antirequisites	NIL				
Course Description	This course examines decentralized systems from the perspective of consensus theory, smart contract design, and scalable blockchain infrastructures. It equips students with the expertise to develop secure, auditable, and gas-optimized smart contracts, and to architect interoperable blockchain networks using modern frameworks. The curriculum integrates financial and regulatory use cases such as tokenization, identity, compliance automation, and multi-chain settlements, with emphasis on performance bottlenecks, security vulnerabilities, and protocol-level decisions in distributed financial systems.				
Course Objective	To enable students to design, implement, and audit scalable and secure distributed ledger systems and smart contracts tailored to real-world FinTech, RegTech, and DeFi applications.				
Course Outcomes	On successful completion of this course the students shall be able to: 1. Architect permissionless and permissioned blockchain infrastructures with consensus optimization. 2. Develop and deploy secure smart contracts with modular architecture and gasefficient logic. 3. Analyze blockchain security threats, contract vulnerabilities, and governance implications. 4. Integrate DLT components into financial, compliance, and tokenization workflows.				
Course Content:					
Module 1:	Consensus Engineering and Distributed Coordination	Quiz		Knowledge based Quiz	9 Sessions
Topics: Nakamoto vs Byzantine Fault Tolerance (BFT), PoW, PoS, DPoS, leader election, finality guarantees, cross-chain consensus Tools: Ethereum Geth, Tendermint, Hyperledger Fabric Use Cases: Designing a high-throughput private blockchain for credit scoring data exchanges					

Module 2:	Smart Contract Development and Security	Quiz		Knowledge based Quiz	9 Sessions
Topics: Smart contract architecture patterns, reentrancy, integer overflow, ownership & role-based access, gas profiling Tools: Solidity, Hardhat, Foundry, Mythril (security analysis) Use Cases: Building a secure lending contract for DeFi with admin safeguards and circuit breakers					
Module 3:	Interoperability and Cross- Chain Infrastructure	Quiz		Knowledge based Quiz	9 Sessions
Topics: Atomic swaps, wrapped tokens, relays, bridges, ZK rollups, Layer 2 scaling, state channels Tools: Chainlink CCIP, LayerZero, zkSync, Polygon SDK Use Cases: Architecting a bridge for cross-chain stablecoin transfers between CBDC testnets					
Module 4:	Distributed Ledger Identity and Compliance	Quiz		Knowledge based Quiz	9 Sessions
Topics: Decentralized identifiers (DIDs), verifiable credentials, zk-SNARKs for KYC/AML, compliancedriven smart contracts Tools: uPort, Polygon ID, Circom, Semaphore Use Cases: Deploying reusable privacy-preserving KYC credentials for a P2P lending DApp					
Module 5:	Tokenization, Governance, and Ecosystem Integration	Quiz		Assignment	9 Sessions
Topics: Token economics, governance models (on-chain/off-chain), DAO frameworks, treasury automation, oracles, audit trails Tools: OpenZeppelin Governor, Snapshot, Gnosis Safe, The Graph, Chainlink Use Cases: Structuring a DAO for pooled investment in tokenized securities with real-time audit visibility					
Project work/Assignment: Students will design and deploy a complete decentralized system tailored to a real-world FinTech or RegTech scenario. Example projects include a compliance-aware lending platform with cross-chain interoperability or a decentralized identity system integrating zero-knowledge proofs and smart contracts.					

Evaluation includes code audits, testnet deployment, and integration with oracles or bridges.

Text Book

T1. Co-branded Zell and Deloitte Textbook for Distributed Ledger Architectures and Smart Contract

Systems

References

- Mastering Blockchain (Imran Bashir) – Springer
- Building Ethereum DApps (Roberto Infante) – Apress
- DeFi and the Future of Finance (Campbell R. Harvey) – Wiley

Web References

- https://ethereum.org/en/developers/
- https://docs.chain.link/
- https://docs.polygon.technology/
- https://zkdocs.com/
- https://consensys.io/audits/

Topics relevant to SKILL DEVELOPMENT:

This course prepares students to function as blockchain developers, security auditors, and infrastructure architects by providing mastery over smart contract development, consensus protocol tuning, and decentralized identity management. Students will build full-stack decentralized solutions, integrating multi-chain logic, zero-knowledge proofs, and secure governance tools. They will gain experience in formal verification, testnet deployments, and scaling challenges, equipping them to lead innovation in FinTech infrastructure, DeFi products, and regulated digital asset ecosystems.

*Course Code: CSS3502	Course Title: Scalable DApp Engineering and Web3 System Integration Type of Course: PCC– Theory & Practical	L-T-P- C	2	0	2	3

Version No.	1.0
Course Prerequisites	CSEXXXX
Antirequisites	NIL
Course Description	This course empowers students to design, deploy, and scale decentralized applications (DApps) across multi-chain blockchain ecosystems. Learners will engineer modular, gasefficient smart contract architectures, integrate Web3 wallets and permissioned frontends, and use decentralized storage and indexing protocols for backend systems. Emphasis is placed on event-driven execution, DAO integrations, cross-chain communication, and resilient DApp deployments optimized for real-world FinTech, DeFi, and digital asset applications.
Course Objective	To enable students to architect full-stack DApps with modular smart contracts, scalable Web3 frontends, and interoperable backend components, using decentralized protocols and open-source tools suited for real-world blockchain deployments.
Course Outcomes	On successful completion of this course, the students shall be able to: 1. Design scalable Web3 architectures using modular frameworks and open-source blockchain tools. (Create) 2. Integrate decentralized application frontends with wallet protocols, permissioned backends, and decentralized identity layers. (Apply) 3. Optimize smart contract logic for gas efficiency, storage utilization, and secure transaction flows. (Analyze) 4. Implement DAO voting, governance automation, and multi-chain deployment strategies for real-world blockchain applications. (Evaluate)
Course Content:	

Module 1:	Modular Smart Contract Architecture	Quiz		Knowledge based Quiz	9 Sessions
Topics: Upgradeable proxies, modular contracts, logic/data separation, composable design Tools: Foundry, OpenZeppelin SDK, Hardhat Use Case: Lending protocol with modular interest, collateral, and governance modules					

Module 2:	Web3 Frontend & Wallet Integration	Quiz		Knowledge based Quiz	9 Sessions

Topics: Wallet adapters (MetaMask, WalletConnect), contract UIs, event listeners

Tools: React, Wagmi, Ethers.js, RainbowKit

Use Case: Peer-to-peer token swap DApp with secure wallet interaction

Module 3:	Decentralized Storag	Quiz e		Knowledge based Quiz	9 Sessions

Topics: IPFS for asset hosting, token metadata, query indexing with The Graph
Tools: IPFS, Web3.Storage, The Graph, GraphQL
Use Case: NFT DApp with indexed on-chain metadata and decentralized file hosting

Module 4:	Gas Optimization & Secure Transaction Flows	Quiz		Knowledge based Quiz	9 Sessions

Topics: State minimization, calldata compression, gas profiling, batch execution

Tools: Hardhat Gas Reporter, Tenderly (free tier), Solidity Coverage

Use Case: Automated yield farming DApp with gas-minimized contract execution

Module 5:	Cross-Chain Deployment & DAO Integration	Quiz		Assignment	9 Sessions

Topics: EVM-compatible deployments, DAO voting, Gnosis Safe automation, Snapshot governance
Tools: Gnosis Safe (Open Source), Snapshot, Arbitrum Nova, Polygon SDK
Use Case: DAO-governed grant disbursement DApp with multi-chain treasury control

Project work/Assignment:
Students will build and deploy a production-grade DApp (e.g., decentralized identity verification system, crowdfunding DAO, or tokenized invoice platform) with smart contract backend, wallet-enabled frontend, gas optimization strategies, DAO voting flows, and decentralized file storage. Deliverables include testnet deployments, smart contract repositories, audit summaries, and live UI demonstration.

Text Book
T1. Co-branded Zell and Deloitte Textbook for Scalable DApp Engineering and Web3 System Integration

References

- Nader Dabit – The Complete Guide to Full Stack Ethereum Development
- OpenZeppelin Docs (https://docs.openzeppelin.com/
- The Graph Docs (https://thegraph.com/docs/)
- Ethereum Development Docs (https://ethereum.org/developers/)

Web References

• https://viem.sh/docs/ • https://wagmi.sh/ • https://docs.web3.storage/ • https://snapshot.org/#/ • https://hardhat.org/ • https://docs.ipfs.tech/
Topics relevant to SKILL DEVELOPMENT: Students gain real-world expertise in building DApps with modular Solidity contracts, state-efficient execution, and scalable Web3 frontends. By integrating open-source tools like Wagmi, IPFS, The Graph, and Snapshot, they develop end-to-end decentralized systems with governance, authentication, indexing, and storage features. This course prepares learners for roles in Web3 engineering, protocol development, and smart contract architecture, equipping them to create scalable, user-centric blockchain applications across FinTech, digital assets, and DAO ecosystems.

*Course Code: CSS3502	Course Title: Decentralized Finance (DeFi) Protocols on Enterprise Blockchain Applications Type of Course: PCC– Theory & Practical	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					
Anti-requisites	NIL					
Course Description	This course explores advanced decentralized financial architectures implemented on enterprise-grade blockchain networks. It focuses on the design, deployment, and integration of DeFi protocols—such as lending, yield farming, synthetic assets, and automated market makers (AMMs)—within private, consortium, and hybrid enterprise blockchain infrastructures. Students will learn to design tokenomics, model liquidity, enforce compliance, and implement security primitives for real-world DeFi applications, bridging the gap between permissionless innovation and enterprise-grade stability.					
Course Objective	To enable learners to architect, customize, and deploy enterprise-ready DeFi protocols by leveraging permissioned and interoperable blockchain technologies. The course focuses on applying DeFi logic in regulatory environments while addressing issues of scale, composability, asset backing, and real-world collateralization.					

Course Outcomes	On successful completion of this course the students shall be able to: 1. Analyze DeFi protocol architectures (DEXs, lending, DAOs) and adapt them for enterprise blockchain environments. (Analyze) 2. Design permissioned and hybrid smart contracts for asset management, yield generation, and liquidity provisioning. (Design) 3. Develop composable DeFi modules using open-source tools and enterprise blockchain SDKs. (Create) 4. Evaluate the security, governance, and regulatory alignment of DeFi implementations in enterprise contexts. (Evaluate)				
Course Content:					
Module 1:	Enterprise-Compatible DeFi Architectures	Quiz		Knowledge based Quiz	9 Sessions
Topics: Permissioned vs public DeFi, stablecoin design (collateral-backed, algorithmic), liquidity pools in enterprise contexts Tools: Hyperledger Besu, Quorum, OpenZeppelin Contracts, Solidity Use Case: Building a compliant enterprise liquidity pool with KYC gating and interest accrual					
Module 2:	Composable Yield Strategies & Automated Lending	Quiz		Knowledge based Quiz	9 Sessions
Topics: Protocol layering, dynamic yield models, interest rate oracles, liquidation risk modeling Tools: Aave open-source protocol, Compound SDK (forked), Chainlink Keepers Use Case: Automated lending vault with institutional collateral and dynamic rate optimization					
Module 3:	Institutional AMMs & DEXs with Compliance Layers	Quiz		Knowledge based Quiz	9 Sessions
Topics: Constant function market makers (CFMMs), impermanent loss hedging, regulated DEX deployments Tools: Uniswap V2 (forked), Balancer open-source SDK, Solidity + Python backtesting Use Case: Permissioned AMM for tokenized corporate bonds with pre-trade whitelisting					
Module 4:	DeFi Governance, DAOs & Risk Mitigation	Quiz		Knowledge based Quiz	9 Sessions

Topics: DAO smart contracts, quorum rules, treasury automation, voting snapshots, DeFi insurance mechanisms Tools: Gnosis Safe, Snapshot.org, Aragon OSx (open-source DAO framework) Use Case: Governance-enabled DAO for managing DeFi fund allocation and protocol upgrades					
Module 5:	Real-World Asset (RWA) Tokenization & Settlement	Quiz		Assignment	9 Sessions
Topics: Bridging on-chain and off-chain assets, tokenized invoices, yield-bearing RWAs, settlement logic Tools: Centrifuge, AssetMantle, Solidity + IPFS integration Use Case: Invoice financing via tokenized RWAs on permissioned blockchain with automated yield routing					
Project work/Assignment: Students will build and deploy a modular DeFi system on an enterprise blockchain testnet—comprising tokenized RWAs, permissioned liquidity pools, and DAO governance. The solution will integrate off-chain asset origination, on-chain collateral management, protocol-layer compliance, and auditability. Students must demonstrate simulations for yield generation, collateral risk, and user access control.					
Text Book T1. Co-branded Zell and Deloitte Textbook for Decentralized Finance (DeFi) Protocols on Enterprise Blockchain Applications					
References • Shermin Voshmgir – Token Economy • Andreas Antonopoulos & Gavin Wood – Mastering Ethereum • Camila Russo – The Infinite Machine (for protocol design insights) Web References • https://docs.aave.com/ • https://docs.uniswap.org/ • https://besu.hyperledger.org/en/latest/ • https://docs.centrifuge.io/					

Topics relevant to SKILL DEVELOPMENT:

Students will gain deep technical fluency in DeFi architecture design, protocol-level coding, modular yield modeling, and institutional smart contract deployment. They will learn to fork, customize, and deploy opensource DeFi tools while integrating real-world asset logic and DAO-based governance. Learners will understand how to balance performance, compliance, and automation in building DeFi systems suitable for enterprise adoption. This prepares them for roles in Web3 protocol engineering, DeFi product development, blockchain finance consulting, and smart contract auditing.

Course Code: CSS3502	Course Title: Blockchain Data Engineering and Regulatory Intelligence Type of Course: PCC– Theory & Practical	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					
Anti-requisites	NIL					
Course Description	This course equips students with advanced capabilities in blockchain-native data engineering, enabling them to design, extract, validate, and pipeline transactional and smart contract data from public and permissioned ledgers. The focus is on applying regulatory intelligence to real-time blockchain data, ensuring auditability, privacy compliance, financial reporting, and risk-based controls. Students will work with blockchain indexing frameworks, data lakes, zero-knowledge proof systems, and chain analytics tools to build scalable architectures that meet enterprise and regulatory needs in the FinTech and Web3 ecosystems.					
Course Objective	To enable learners to build blockchain-native data engineering systems capable of extracting structured insights from on-chain activity, while implementing regulatory rulesets, anomaly detection, and audit compliance across decentralized applications and financial operations.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Design blockchain-aware data pipelines that support both structured and unstructured smart contract interactions. (Design) 2. Develop secure data extraction and storage mechanisms with privacy preservation and cryptographic validation. (Develop) 3. Implement regulatory rule-checking logic (e.g., AML, KYC, tax) within streaming and batch on-chain data analytics. (Apply) 4. Evaluate blockchain systems for compliance, audit trail generation, and zeroknowledge proofs. (Evaluate)					
Course Content:						
Module 1:	Blockchain Data Modeling and Indexing	Quiz		Knowledge based Quiz	9 Sessions	
Topics: On-chain data schema design, transaction graphs, event logs, smart contract state extraction Tools: The Graph Protocol, Hyperledger Explorer, Geth Trace, Ethers.js, Subgraphs Use Case: Extracting multi-token transfer data and contract events for a DeFi dashboard						
Module 2:	Data Pipelines and Storage Architectures	Quiz		Knowledge based Quiz	9 Sessions	

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

CSS3502</td><td colspan="3">Course Title: Interoperability Engineering and Cross-Chain System Design - I
Type of Course: PCC– Theory & Practical</td><td>L-T-P- C</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Version No.</td><td colspan="8">1.0</td></tr>
<tr><td>Course Prerequisites</td><td colspan="8">CSEXXXX</td></tr>
<tr><td>Antirequisites</td><td colspan="8">NIL</td></tr>
<tr><td>Course Description</td><td colspan="8">This course provides advanced expertise in engineering cross-chain systems and ensuring interoperability across diverse blockchain ecosystems. It focuses on protocols, messaging layers, and architecture patterns that enable seamless asset transfers, state validation, and smart contract calls between networks such as Ethereum, Cosmos, Polkadot, Avalanche, and Solana. Students will learn to design secure interoperability bridges, assess trust assumptions, implement cross-chain asset transfers, and deploy oracle-driven and zeroknowledge-based relayers. The course also covers cross-domain composability, consensus abstraction, and real-world use cases such as DeFi bridges, cross-chain NFT ownership, and omnichain liquidity routing.</td></tr>
<tr><td>Course Objective</td><td colspan="8">To equip students with the technical skills and architectural understanding required to build secure, scalable, and trust-aware interoperability layers across blockchain networks, enabling seamless data, asset, and logic flow for next-generation decentralized applications and FinTech infrastructure.</td></tr>
<tr><td>Course Outcomes</td><td colspan="8">On successful completion of this course the students shall be able to:
1. Design and evaluate interoperability architectures for message, token, and contract logic transfer between heterogeneous chains (Design)
2. Implement secure and verifiable relayers and bridges using LayerZero, IBC, or light client protocols
(Implement)
3. Analyze trust assumptions and consensus compatibility across networks for enterprise cross-chain use cases
(Analyze)
4. Develop and simulate composable cross-chain applications with failover and fallback mechanisms
(Develop)</td></tr>
<tr><td>Course Content:</td><td colspan="8"></td></tr>
<tr><td>Module 1:</td><td colspan="2">Cross-Chain Interoperability Models and Trust Layers</td><td>Quiz</td><td></td><td colspan="2">Knowledge based Quiz</td><td>9 Sessions</td></tr>
<tr><td colspan="9">Topics: Messaging protocols (arbitrary message passing, token bridges), Notary vs Light Client vs Relay-based trust, Atomic swaps vs asynchronous commits, Chain-specific trust constraints
Tools: LayerZero, Axelar GMP, Chainlink CCIP, Cosmos IBC
Use Case: Secure asset transfer and message relay between Ethereum and Avalanche using LayerZero</td></tr>
</table>

Module 2:	Cross-Chain Smart Contract Design and Composability	Quiz		Knowledge based Quiz	9 Sessions

Course Code: CSS3502	Course Title: Interoperability Engineering and Cross-Chain System Design -II Type of Course: PCC– Theory & Practical	L-T-P- C	2	0	2	3
Version No.	1.0					
Course Prerequisites	CSEXXXX					
Antirequisites	NIL					
Course Description	This course focuses on the secure engineering of interoperability protocols between heterogeneous blockchain ecosystems. Students will examine architectural and cryptographic mechanisms that enable communication, state synchronization, asset transfer, and smart contract logic execution across disparate blockchain networks. The course spans trust minimization models, relayer architectures, cross-domain composability, and the practical challenges of bridging decentralized and enterprise-grade systems. Special focus is placed on real-world use cases such as stablecoin transfer, NFT provenance across chains, cross-chain risk pricing, and interbank settlement routing in tokenized finance.					
Course Objective	To enable learners to design, evaluate, and implement scalable cross-chain systems that support secure data, asset, and logic interoperability between public and private blockchain networks—combining cryptographic integrity with enterprise compliance.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Architect modular cross-chain interoperability systems across Layer-1, Layer-2, and sidechain platforms (Create) 2. Apply oracle-based, light-client-based, and ZK-based trust-minimized messaging protocols (Apply) 3. Analyze security risks, audit patterns, and replay prevention strategies for bridges and relay systems (Analyze) 4. Deploy and test cross-chain smart contracts and messaging interfaces in multi-chain simulations (Evaluate)					
Course Content:						

Module 1:	Cross-Chain Messaging Models and Trust Topologies	Quiz		Knowledge based Quiz	9 Sessions
Topics: Cross-chain messaging (arbitrary calls, token standards), Light clients vs relayers, Oracle-based bridges, Finality and consensus abstractions Tools: LayerZero, Chainlink CCIP, Axelar GMP Use Case: Secure USDC token bridging between Ethereum and Avalanche using GMP					
Module 2:	Smart Contract Synchronization and Multi-Chain Composability	Quiz		Knowledge based Quiz	9 Sessions
• Topics: Cross-chain contract triggers, Syncing events and states, Stateless routing, Universal contract interfaces • Tools: Hyperlane, CosmWasm with IBC-Hooks, Solidity with LayerZero • Use Case: NFT minting with verification across Ethereum and Solana for cross-chain loyalty programs					
Module 3:	Security Architecture for Bridges and Relayers	Quiz		Knowledge based Quiz	9 Sessions
• Topics: Replay protection, Fraud proofs, Timeout logic, Validator vs MPC-based signing, Decentralized consensus verification • Tools: Forta, OpenZeppelin Defender, Wormhole Validator Set • Use Case: Audit-resistant bridge for government-issued token between CBDC testnets					
Module 4:	ZK Proofs, Oracle Layers, and Validator Networks	Quiz		Knowledge based Quiz	9 Sessions
• Topics: zkLightClients (Succinct, zkBridge), MPC-based relays, OCR consensus in Chainlink, Zero-knowledge message attestation • Tools: Succinct zkLightClient, Chainlink OCR, zkBridge • Use Case: Cross-chain settlement conditionally triggered by real-world event using oracle + ZKP					
Module 5:	Enterprise and Regulatory Interoperability Systems	Quiz		Knowledge based Quiz	9 Sessions
• Topics: Stablecoin flow through regulated DeFi, Cross-chain identity/auth layers, ISO 20022-toEVM compliance wrappers, Tokenization pipelines • Tools: Fireblocks for MPC routing, Chainlink Proof of Reserve, Ripple XLS-38 bridge • Use Case: Interbank fund rebalancing between DLT-based CBDC systems and Ethereum ERC-20 rails					

Project Work / Assignment: Students will design a complete cross-chain architecture for a decentralized use case (e.g., lending, NFT minting, cross-chain identity, or real-time asset pricing), simulate cross-chain message flow using LayerZero or Hyperlane, and implement fail-safes using monitoring tools such as Forta. The project must include a report on trust assumptions, compliance mapping, and testing across multiple blockchain testnets.
Text Book T1. Co-branded Zell and Deloitte Textbook for Interoperability Engineering and Cross-Chain System Design
References • Vitalik Buterin – Cross-Chain Bridge Security • Sunny Aggarwal – IBC Architecture (Cosmos) • Gavin Wood – XCM Design (Polkadot) Web References • https://layerzero.network/docs • https://docs.axelar.dev/ • https://chain.link/ccip • https://cosmos.network/ibc
Topics relevant to SKILL DEVELOPMENT: This course enhances student expertise in developing secure, scalable, and interoperable systems in a multi-

chain blockchain environment. Students gain practical skills in message routing, bridge deployment, contract synchronization, and fault-tolerant design using real-world cross-chain protocols. The course emphasizes simulation and implementation of messaging models, light-client-based relays, validator security patterns, and composable logic flows across DeFi, NFTs, and tokenized banking systems. These competencies are highly relevant for Web3 infrastructure architects, decentralized app developers, and FinTech engineers working with digital asset interoperability, decentralized identity, and regulated crossborder finance.