GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

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

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Program Regulations and Curriculum 2023-2027

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)

Regulation Number: PU/AC-21.5/SoCSE2/CBC/2023-2027

Resolution No. 5 of the 21st Meeting of the Academic Council held on 06th Sept 2023, and ratified by the Board of Management in its 22nd Meeting held on 02nd Nov 2023.

(As amended up to the 26th Meeting of the Academic Council held on 25th July 2025, and ratified by the Board of Management in its 27th Meeting held on 28th July 2025)

September 2023

Table of Contents

PART A – PROGRAM REGULATIONS

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

1.1 Vision of the University

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

1.2 Mission of the University

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

1.3 Vision of Presidency School of Computer Science and Engineering

To be a value-based, practice-driven Presidency School of Computer Science and Engineering, committed to developing globally competent engineers, dedicated to developing cutting-edge technology to enhance the quality of life.

1.4 Mission of Presidency School of Computer Science and Engineering

- Cultivate a practice-driven environment with computing-based pedagogy, integrating theory and practice.
- Attract and nurture world-class faculty to excel in teaching and research in the realm of computing sciences.
- Establish state-of-the-art computing facilities for effective teaching and learning experiences.
- Promote interdisciplinary studies to nurture talent for global impact.
- Instill entrepreneurial and leadership skills to address social, environmental and community needs.

2. Preamble to the Program Regulations and Curriculum

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

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

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

3. Short Title and Applicability

a. These Regulations shall be called the Bachelor of Technology Degree Program Regulations and Curriculum 2023-2027.
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 2023-2027 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 2023-2024.

4. Definitions

In these Regulations, unless the context otherwise requires:

a. "Academic Calendar" means the schedule of academic and miscellaneous events as approved by the Vice Chancellor;

b. "Academic Council" means the Academic Council of the University;

c. "Academic Regulations" means the Academic Regulations, of the University;

d. "Academic Term" means a Semester or Summer Term;

e. "Act" means the Presidency University Act, 2013;

f. "AICTE" means All India Council for Technical Education;

g. "Basket" means a group of courses bundled together based on the nature/type of the course;

h. "BOE" means the Board of Examinations of the University;

i. "BOG" means the Board of Governors of the University;

j. "BOM" means the Board of Management of the University;

k. "BOS" means the Board of Studies of a particular Department/Program of Study of the University;

l. "CGPA" means Cumulative Grade Point Average as defined in the Academic Regulations;

m. "Clause" means the duly numbered Clause, with Sub-Clauses included, if any, of these Regulations;

n. "COE" means the Controller of Examinations of the University;

o. "Course In Charge" means the teacher/faculty member responsible for developing and organising the delivery of the Course;

p. "Course Instructor" means the teacher/faculty member responsible for teaching and evaluation of a Course;

q. "Course" means a specific subject usually identified by its Course-code and Course-title, with specified credits and syllabus/course-description, a set of *references, taught by some teacher(s)/course-instructor(s) to a specific class (group of students) during a specific Academic Term;*

r. "Curriculum Structure" means the Curriculum governing a specific Degree Program offered by the University, and, includes the set of Baskets of Courses along with minimum credit requirements to be earned under each basket for a degree/degree with specialization/minor/honours in addition to the relevant details of the Courses and Course catalogues (which describes the Course content and other important information about the Course). Any specific requirements for a particular program may be brought into the Curriculum structure of the specific program and relevant approvals should be taken from the BOS and Academic Council at that time.

s. "DAC" means the Departmental Academic Committee of a concerned Department/Program of Study of the University;

t. "Dean" means the Dean / Director of the concerned School;

u. "Degree Program" includes all Degree Programs;

v. "Department" means the Department offering the degree Program(s) / Course(s) / School offering the concerned Degree Programs / other Administrative Offices;

w. "Discipline" means specialization or branch of B.Tech. Degree Program;

x. "HOD" means the Head of the concerned Department;

y. "L-T-P-C" means Lecture-Tutorial-Practical-Credit – refers to the teaching – learning periods and the credit associated;

z. "MOOC" means Massive Open Online Courses;

aa. "MOU" means the Memorandum of Understanding;

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

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

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

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

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

gg. "PSCS" means the Presidency School of Computer Sciece and Engineering;

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

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

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

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

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

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

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

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

pp. "UGC" means University Grant Commission;

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

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

5. Program Description

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

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

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

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

6. Minimum and Maximum Duration

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

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

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

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

6.5 The enrolment of the student who fails to complete the mandatory requirements for the award of the concerned Degree (refer Section: 19.0**.** of Academic Regulations) in the prescribed maximum duration (Clauses 18.1

and 18.2 of Academic Regulations), shall stand terminated and no Degree shall be awarded.

7 Programme Educational Objectives (PEO)

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

PEO 01: Demonstrate success as Computer Science and Engineering with innovative skills, moral and ethical values.

PEO 02: Engage in lifelong learning through research and professional development.

PEO 03: Serve as a leader in the profession through consultancy, extension activities and/ or entrepreneurship

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

8.1 Programme Outcomes (PO)

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

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

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

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

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

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

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

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

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

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

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

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

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

8.2 Program Specific Outcomes (PSOs):

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

PSO 01: Problem Analysis: Identify, formulate, research literature, and analysis complex engineering problems related to Block Chain principles and practices, Programming and Computing technologies reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PSO 02: Design/development of Solutions: Design solutions for complex engineering problems related to Block Chain principles and practices, Programming and Computing technologies and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, cultural, societal and environmental consideration.

PSO 03: Modern Tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities related to Block Chain principles and practices, Programming Blockchain Computing & analytics with an understanding of the limitations.

9 Admission Criteria (as per the concerned Statutory Body)

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

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

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

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

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

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

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

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

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

10 Lateral Entry / Transfer Students requirements

10.1 Lateral Entry

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

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

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

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

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

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

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

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

The ***Minimum Credit Requirements*** for the award of the Bachelor of Technology (B.Tech. Computer Science and Engineering Block Chain) Degree prescribed by the concerned Bachelor of Technology Degree Program Regulations and Curriculum, 2023-2027, 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. Computer Science and Engineering Block Chain) Degree as prescribed by the Regulations for B.Tech. () 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. / B.E. / B.S. Four Degree Program from the concerned University, are declared equivalent and acceptable by the Equivalence Committee constituted by the Vice Chancellor for this purpose. Further, the Equivalence Committee may also prescribe the Courses and Credits the concerned students shall have to mandatorily complete, if admitted to the 2nd Year of the B.Tech. Program of the University.

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

11 Change of Branch / Discipline / Specialization

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

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

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

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

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

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

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

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

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

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

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

12.2 Academic performance evaluation of every registered student in every Course registered by the student is carried out through various components of Assessments spread across the Semester. The nature of components of Continuous Assessments and the weightage given to each component of Continuous Assessments (refer Clause 8.8 of academic regulation) 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 regulation) shall be awarded to a student based on her/his overall performance relative to the class performance distribution in the concerned Course. These Letter Grades not only indicate a qualitative assessment of the student's performance but also carry a quantitative (numeric) equivalent called the Grade Point.

12.5 Assessment Components and Weightage

12.6

Table 1: Assessment Components and Weightage for different category of Courses

Nature of Course and Structure	Evaluation Component	Weightage
Lecture-based Course L component in the L-T-P Structure is predominant (more than 1) (Examples: 3-0-0; 3-0-2; 2-1-0; 2-0-2, 2-0-4 etc.)	Continuous Assessments	50%
	End Term Examination	50%
Lab/Practice-based Course P component in the L-T-P Structure is predominant (Examples: 0-0-4; 1-0-4; 1-0-2; etc.)	Continuous Assessments	75%
	End Term Examination	25%
Skill based Courses like Industry Internship, Capstone project, Research Dissertation, Integrative Studio, Interdisciplinary Project, Summer / Short Internship, Social Engagement / Field Projects, Portfolio, and such similar Non-Teaching Credit Courses, where the pedagogy does not lend itself to a typical L-T-P structure	Guidelines for the assessment components for the various types of Courses, with recommended weightages, shall be specified in the concerned Program Regulations and Curriculum / Course Plans, as applicable.	

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

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

12.7 Minimum Performance Criteria:

12.7.1 Theory only Course and Lab/Practice Embedded Theory Course

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

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

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

12.7.2 Lab/Practice only Course and Project Based Courses

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

12.7.3 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-Clause 12.7.1 and 12.7.2 of academic regulation) in the "Make-Up Examinations" of the concerned Course. Further, the student has an option to re-register for the Course and clear the same in the summer term/ subsequent semester if he/she wishes to do so, provided the Course is offered.

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

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

13.1 The transfer of credits shall be examined and recommended by the Equivalence Committee (Refer **Error! Reference source not found.** of Academic Regulations) and approved by the Dean - Academics.

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

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

13.3.1 A student may complete SWAYAM/NPTEL/other approved MOOCs as mentioned in Clause 13.3 (as per of Academic Regulations) and transfer equivalent credits to partially or fully complete the mandatory credit requirements of Discipline Elective Courses and/or the mandatory credit requirements of Open Elective Courses as prescribed in the concerned Curriculum Structure. However, it is the sole responsibility of the student to complete the mandatory credit requirements of the Discipline Elective Courses and the Open Elective Courses as prescribed by the Curriculum Structure of the concerned Program.

13.3.2 SWAYAM/NPTEL/ other approved MOOCs as mentioned in Clause 13.3 (as per of Academic Regulations) shall be approved by the concerned Board of Studies and placed (as Annexures) in the concerned PRC.

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

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

13.3.5 A student shall request for transfer of credits only from such approved Courses as mentioned in Sub-Clause 13.3.2 above.

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

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

13.3.8 The credit equivalence of the SWAYAM/NPTEL/other approved MOOCs are based on Course durations and/or as recommended by the Course offering body/institute/university. The Credit Equivalence mapped to SWAYAM/ NPTEL approved Courses based on Course durations for transfer of credits is summarised in Table shown below. The Grade will be calculated from the marks received by the Absolute Grading Table **Error! Reference source not found.**. in the Academic Regulation.

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

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

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

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

13.4 The maximum number of credits that can be transferred by a student shall be limited to forty percent (40%) of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum for the award of the concerned Degree. However, the grades obtained in the Courses transferred from other Institutions/MOOCs, as mentioned in this Section (13.**Error! Reference source not found.**), shall not be included in the calculation of the CGPA.

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

- S (Satisfactorily Completed): Awarded when the student successfully completes all prescribed course requirements.
- NC (Not Completed): Awarded when the student fails to meet the prescribed course requirements.

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

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

PART B – PROGRAM STRUCTURE

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

The B.Tech. (Computer Science and Engineering-Block Chain) Program Structure (2023-2027) totalling 160 credits. Table 3 summarizes the type of baskets, number of courses under each basket and the associated credits that are mandatorily required for the completion of the Degree.

Table 3: B.Tech. (Computer Science and Engineering Block Chain) 2023-2027: Summary of Mandatory Courses and Minimum Credit Contribution from various Baskets

Sl. No.	Baskets	Credit Contribution
1	School Core	**68**
2	Professional Core	**65**
3	Discipline Elective	**18**
4	Open Elective	**9**
	Total Credits	**160 (Minimum)**

In the entire Program, the practical and skill based course component contribute to an extent of approximately 60% out of the total credits of 160 for B.Tech. (Computer Science and 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 of Academic Regulations;

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

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

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

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

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

Table 3.1 List of School Core Courses

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Type of Skill
1	MAT1001	Calculus and Linear Algebra	3	0	2	4	5	EM
2	PHY1002	Optoelectronics and Device Physics	2	0	2	3	4	F
3	ECE1001	Elements of Electronics Engineering	3	0	2	4	5	F
4	ENG1002	Technical English	1	0	2	2	3	S
5	PPS1001	Introduction to soft skills	0	0	2	1	2	S
6	CSE1004	Problem Solving Using C	1	0	4	3	5	S
7	CHE1018	Environmental Science	1	0	2	0	3	F
8	PPS1011	Introduction to Verbal Ability	0	1	0	0	1	S

9	MAT1003	Applied Statistics	1	0	2	2	3	EM
10	ECE2007	Digital Design	2	0	2	3	4	F
11	CIV1008	Basic Engineering Sciences	2	0	0	2	2	F
12	MEC1006	Engineering Graphics	2	0	0	2	2	F
13	CSE1001	Problem Solving using JAVA	2	0	2	3	4	S
14	ENG2001	Advanced English	1	0	2	2	3	S
15	PPS1012	Enhancing personality through Soft Skills	0	0	2	1	2	S
16	ECE2010	Innovative Projects using Arduino	-	-	-	1	0	S
17	MAT1002	Transform Techniques, Partial Differential Equations and Their Applications	3	0	0	3	3	EM
18	CSE2001	Data Structures and Algorithms	3	0	2	4	5	S
19	MAT2004	Discrete Mathematical Structures	3	0	0	3	3	S
20	ECE2011	Innovative Projects Using Raspberry Pi	-	-	-	1	0	S
21	PPS4002	Introduction to Aptitude	0	0	2	1	2	S
22	MAT2003	Numerical Methods for Engineers	1	0	2	2	3	EM
23	PPS4004	Aptitutde Training Intermediate	0	0	2	1	3	S
24	CSE3216	Mastering Object-Oriented Concepts in Python	0	0	2	1	3	S
25	CSE7000	Internship	-	-	-	2	0	S
26	CSE2274	Competitive Programming and Problem Solving	0	0	4	2	4	S
27	CSE7101	Mini Project	0	0	0	5	0	S
28	CSE7300	Capstone Project	0	0	0	10	0	S/EM
29	APT4006	Logical and Critical Thinking	0	0	2	0	2	AT
30	APT4026	Aptitude for Employability	0	0	2	0	2	AT
31	PPS4027	Preparedness for Interview	0	0	2	0	2	SS
32	CIV7601	Universal Human Values and Ethics	0	0	0	0	-	S
33	LAW7601	Indian Constitution	0	0	0	0	-	S
		TOTAL				68		

Table 3.2 List of Program Core Courses

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Type of Skill
3	CSE3155	Data Communications and Computer Networks	3	0	2	4	5	S
4	CSE2009	Computer Organization and Architecture	3	0	0	3	3	S
6	CSE3190	Fundamentals of Data Analytics	2	0	2	3	4	S
8	CSE2014	Software Engineering	3	0	0	3	3	S
10	CSE1005	Programming in Python	1	0	4	3	5	S
2	CSE2007	Design and Analysis of Algorithms	3	0	0	3	6	S

3	CSE3156	Database Management Systems	3	0	2	4	9	S
4	CSE3155	Operating Systems	3	0	0	3	6	S
5	CSE3078	Cryptography and Network Security	3	0	0	3	6	S
1	CBC2500	Smart Contract and Solidity	3	0	0	3	3	S
2	CBC2502	Distributed Ledger Technology	3	0	0	3	3	S
5	CSE2264	Essentials of AI	3	0	0	3	3	S
6	CBC2509	Consensus Algorithms and Network Design	3	0	0	3	3	S
8	CBC2501	Smart Contract and Solidity Lab	0	0	2	1	2	S
9	CBC2503	Distributed Ledger Technology Lab	0	0	2	1	2	S
10	CSE2265	Essentials of AI Lab	0	0	2	1	2	S
11	CSE2266	Theory of Computation	3	0	0	3	3	S
13	CSE2500	Data Analytics	2	0	0	2	2	S
14	CSE2501	Data Analytics Lab	0	0	2	1	2	S
1	CBC2507	Token Standards and Creation	3	0	0	3	3	S
2	CBC2508	Token Standards and Creation Lab	0	0	2	1	2	S
3	CBC2504	Blockchain Security and Performance	3	0	0	3	3	S
4	CBC2505	Blockchain Security and Performance Lab	0	0	4	2	4	S
5	CBC2506	Blockchain Architecture Design	3	0	0	3	3	S
10	CSE2272	Cloud Computing	2	0	0	2	2	S
11	CSE2273	Cloud Computing Lab	0	0	2	1	2	S
		Total				65		

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

18.2.2 The student may do the project work in an Industry / Company or academic / research institution of her / his choice subject to the above mentioned condition (Sub-Clause 18.2.1). Provided further, that the Industry / Company or academic / research institution offering such project work confirms to the University that the project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.3 Capstone Project

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

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

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

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

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

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

18.4 Research Project / Dissertation

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

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

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

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

Minor Stream: Minimum Credits to be earned from this basket = 18 Credits

Table 3.3 List of Discipline Elective Courses Discipline Elective courses/Specilaization Tracks – Minimum of 18 credits is to be earned by the student in a particular track "									
	Track 1: Block Chain with Cryptography						Contact Hours		
1	CSE2021	Data Mining	3	0	0	3	3	S/EM	MAT1001

2	CSE3352	Information Visualization	3	0	0	3	3	S/EM	CSE2027
3	CSE3082	Object Oriented Analysis and Design	3	0	0	3	3	S/EM	
4	CBC3400	Cryptography and Security in Blockchain	3	0	0	3	3	S/EM	
5	CBC3401	Crypto Trading Strategies & Risk Management	3	0	0	3	3	S/EM	
6	CBC3402	Bitcoin and Ethereum Protocols	3	0	0	3	3	S/EM	
7	CBC3403	Blockchain for Digital Identity Management	3	0	0	3	3	S/EM	
8	CBC3404	Cryptocurrency Wallet Development	3	0	0	3	3	S/EM	
9	CBC3405	Blockchain Security & Ethical Hacking	3	0	0	3	3	S/EM	
	TRACK 2	**Block Chain with Artificial Intelligence**	L	T	P	C			
10	CBC3406	Introduction to Artificial Intelligence in Block Chain	3	0	0	3	3	S/EM	
11	CBC3407	Machine Learning for Cyber Threat Detection	3	0	0	3	3	S/EM	
12	CBC3408	AI-Powered Fraud Detection in Blockchain	3	0	0	3	3	S/EM	
13	CBC3409	Optimizing Blockchain Networks with AI	3	0	0	3	3	S/EM	
14	CBC3410	Generative AI for Blockchain Applications	3	0	0	3	3	S/EM	
15	CBC3411	Quantum Computing & Blockchain-AI Security	3	0	0	3	3	S/EM	
	TRACK 3	**Decentralized Finance (DeFi) for Blockchain**	L	T	P	C			
16	CBC3412	Introduction to Decentralized Finance (DeFi)	3	0	0	3	3	S/EM	
17	CBC3413	Blockchain in Financial Services	3	0	0	3	3	S/EM	
18	CBC3414	Building Decentralized Applications (DApps) for Finance	3	0	0	3	3	S/EM	
19	CBC3415	Smart Contracts for Financial Products	3	0	0	3	3	S/EM	

20	CBC3416	Decentralized Autonomous Organizations and Risk Management in Finance	3	0	0	3	3	S/EM	
21	CBC3417	Regulatory and Compliance Challenges in DeFi	3	0	0	3	3	S/EM	

20. **List of Open Electives to be offered by the School / Department (Separately for ODD and EVEN Semesters.: Minimum Credits to be earned from this basket = 9 Credits**

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

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

Table 3.4 List of Open Elective Courses
Minimum Credits to be earned from this basket = 9 Credits

Sl. No.	Course Code	Course Name	L	T	P	C	Contact Hours	Prerequisites/ Corequisites	Type of Skill/ Focus
Chemistry Basket									
1	CHE1003	Fundamentals of Sensors	3	0	0	3	3	-	S
2	CHE1004	Smart materials for IOT	3	0	0	3	3	-	S
3	CHE1005	Computational Chemistry	2	0	0	2	2	-	S
4	CHE1006	Introduction to Nano technology	3	0	0	3	3	-	S
5	CHE1007	Biodegradable electronics	2	0	0	2	2	-	S
6	CHE1008	Energy and Sustainability	2	0	0	2	2	-	S
7	CHE1009	3D printing with Polymers	2	0	0	2	2	-	S
8	CHE1010	Bioinformatics and Healthcare IT	2	0	0	2	2	-	S
9	CHE1011	Chemical and Petrochemical catalysts	3	0	0	3	3	-	S
10	CHE1012	Introduction to Composite materials	2	0	0	2	2	-	S
11	CHE1013	Chemistry for Engineers	3	0	0	3	3	-	S
12	CHE1014	Surface and Coatings technology	3	0	0	3	3	-	S
13	CHE1015	Waste to Fuels	2	0	0	2	2	-	S
14	CHE1016	Forensic Science	3	0	0	3	3	-	S
Civil Engineering Basket									
1	CIV1001	Disaster mitigation and management	3	0	0	3	3	-	S
2	CIV1002	Environment Science and Disaster Management	3	0	0	3	3	-	FC
3	CIV2001	Sustainability Concepts in Engineering	3	0	0	3	3	-	S
4	CIV2002	Occupational Health and Safety	3	0	0	3	3	-	S
5	CIV2003	Sustainable Materials and Green Buildings	3	0	0	3	3	-	EM

6	CIV2004	Integrated Project Management	3	0	0	3	3	-	EN
7	CIV2005	Environmental Impact Assessment	3	0	0	3	3	-	EN
8	CIV2006	Infrastructure Systems for Smart Cities	3	0	0	3	3	-	EN
9	CIV2044	Geospatial Applications for Engineers	2	0	2	3	4	-	EM
10	CIV2045	Environmental Meteorology	3	0	0	3	3	-	S
11	CIV3046	Project Problem Based Learning	3	0	0	3	3	-	S
12	CIV3059	Sustainability for Professional Practice	3	0	0	3	3	-	EN
Commerce Basket									
1	COM2001	Introduction to Human Resource Management	2	0	0	2	2	-	F
2	COM2002	Finance for Non Finance	2	0	0	2	2	-	S
3	COM2003	Contemporary Management	2	0	0	2	2	-	F
4	COM2004	Introduction to Banking	2	0	0	2	2	-	F
5	COM2005	Introduction to Insurance	2	0	0	2	2	-	F
6	COM2006	Fundamentals of Management	2	0	0	2	2	-	F
7	COM2007	Basics of Accounting	3	0	0	3	3	-	F
Computer Science Basket									
1	CSE2002	Programming in Java	2	0	2	3	4	-	S/EM
2	CSE2003	Social Network Analytics	3	0	0	3	3	-	S
3	CSE2004	Python Application Programming	2	0	2	3	4	-	S/ EM
4	CSE2005	Web design fundamentals	2	0	2	3	4	-	S/ EM/EN
5	CSE3111	Artificial Intelligence : Search Methods For Problem Solving	3	0	0	3	3	-	S/ EM/EN
6	CSE3112	Privacy And Security In Online Social Media	3	0	0	3	3	-	S/ EM/EN
7	CSE3113	Computational Complexity	3	0	0	3	3	-	S/ EM/EN
8	CSE3114	Deep Learning for Computer Vision	3	0	0	3	3	-	S/ EM/EN
9	CSE3115	Learning Analytics Tools	3	0	0	3	3	-	S/ EM/EN
Design Basket									
1	DES1001	Sketching and Painting	0	0	2	1	2	-	S
2	DES1002	Innovation and Creativity	2	0	0	2	2	-	F
3	DES1121	Introduction to UX design	1	0	2	2	3	-	S
4	DES1122	Introduction to Jewellery Making	1	0	2	2	3	-	S
5	DES1124	Spatial Stories	1	0	2	2	3	-	S
6	DES1125	Polymer Clay	1	0	2	2	3	-	S
7	DES2001	Design Thinking	3	0	0	3	3	-	S

8	DES1003	Servicability of Fashion Products	1	0	2	2	3	-	F
9	DES1004	Choices in Virtual Fashion	1	0	2	2	3	-	F
10	DES1005	Fashion Lifestyle and Product Diversity	1	0	2	2	3	-	F
11	DES1006	Colour in Everyday Life	1	0	2	2	3	-	F
12	DES2080	Art of Design Language	3	0	0	3	3	-	S
13	DES2081	Brand Building in Design	3	0	0	3	3	-	S
14	DES2085	Web Design Techniques	3	0	0	3	3	-	S
15	DES2089	3D Modeling for Professionals	1	0	4	3	5	-	S
16	DES2090	Creative Thinking for Professionals	3	0	0	3	3	-	S
17	DES2091	Idea Formulation	3	0	0	3	3	-	S
Electrical and Electronics Basket									
1	EEE1002	IoT based Smart Building Technology	3	0	0	3	3	-	S
2	EEE1003	Basic Circuit Analysis	3	0	0	3	3	-	S
3	EEE1004	Fundamentals of Industrial Automation	3	0	0	3	3	-	S
4	EEE1005	Electric Vehicles & Battery Technology	3	0	0	3	3	-	S
5	EEE1006	Smart Sensors for Engineering Applications	3	0	0	3	3	-	S
Electronics and Communication Basket									
1	ECE1003	Fundamentals of Electronics	3	0	0	3	3	-	F
2	ECE1004	Microprocessor based systems	3	0	0	3	3	-	F
3	ECE1005	Journey of Communication systems	3	0	0	3	3		F
4	ECE3089	Artificial Neural Networks	3	0	0	3	3	-	S
5	ECE3090	Digital System Design using VERILOG	3	0	0	3	3		F/EM
6	ECE3091	Mathematical Physics	3	0	0	3	3		F
7	ECE3092	Photonic Integrated Circuits	3	0	0	3	3		F
8	ECE3093	Machine learning for Music Information Retrieval	3	0	0	3	3		F/EM
9	ECE3094	Video Processing and Computer Vision	3	0	0	3	3		F/EM
10	ECE3095	Blockchain and Cryptocurrency Technologies	3	0	0	3	3		S/EM/EN
11	ECE3096	Natural Language Processing	3	0	0	3	3		S/EM/EN
12	ECE3097	Smart Electronics in Agriculture	3	0	0	3	3	-	F/EM
13	ECE3098	Environment Monitoring Systems	3	0	0	3	3	-	F/EM
14	ECE3099	Modern Wireless Communication with 5G	3	0	0	3	3		F/EM/EN
15	ECE3100	Underwater Communication	3	0	0	3	3		F/EM/EN
16	ECE3101	Printed Circuit Board Design	3	0	0	3	3		S/F/EM
17	ECE3102	Consumer Electronics	3	0	0	3	3	-	F/EM

18	ECE3103	Product Design of Electronic Equipment	3	0	0	3	3	-	S/F/ EM / EN
19	ECE3104	Vehicle to Vehicle Communication	3	0	0	3	3		F/EM/ EN
20	ECE3105	Wavelets and Filter Banks	3	0	0	3	3		F/EM
21	ECE3106	Introduction to Data Analytics	3	0	0	3	3	-	F/EM
22	ECE3107	Machine Vision for Robotics	3	0	0	3	3	-	F/EM
English Basket									
1	ENG1008	Indian Literature	2	0	0	2	2	-	-
2	ENG1009	Reading Advertisement	3	0	0	3	3	-	S
3	ENG1010	Verbal Aptitude for Placement	2	0	2	3	4	-	S
4	ENG1011	English for Career Development	3	0	0	3	3	-	S
5	ENG1012	Gender and Society in India	2	0	0	2	2	-	-
6	ENG1013	Indian English Drama	3	0	0	3	3	-	-
7	ENG1014	Logic and Art of Negotiation	2	0	2	3	4	-	-
8	ENG1015	Professional Communication Skills for Engineers	1	0	0	1	1	-	-
Fitness and Wellness Basket									
1	DSA2001	Spirituality for Health	2	0	0	2	2	-	F
2	DSA2002	Yoga for Health	2	0	0	2	2	-	S
3	DSA2003	Stress Management and Well Being	2	0	0	2	2	-	F
Kannada Basket									
1	KAN1003	Kannada Kaipidi	3	0	0	3	3	-	S
2	KAN2003	Pradharshana Kale	1	0	2	2	3	-	S
3	KAN2004	Sahithya Vimarshe	2	0	0	2	2	-	S
4	KAN2005	Anuvadha Kala Sahithya	3	0	0	3	3	-	S
5	KAN2006	Vichara Manthana	3	0	0	3	3	-	S
6	KAN2007	Katha Sahithya Sampada	3	0	0	3	3	-	S
7	KAN2008	Ranga Pradarshana Kala	3	0	0	3	3	-	S
Foreign Language Basket									
1	FRL1004	Introduction of French Language	2	0	0	2	2	-	S
2	FRL1005	Fundamentals of French	2	0	0	2	2	-	S
3	FRL1009	Mandarin Chinese for Beginners	3	0	0	3	3	-	S
Law Basket									
1	LAW1001	Introduction to Sociology	2	0	0	2	2	-	F
2	LAW2001	Indian Heritage and Culture	2	0	0	2	2	-	F
3	LAW2002	Introdcution to Law of Succession	2	0	0	2	2	-	F
4	LAW2003	Introduction to Company Law	2	0	0	2	2	-	2
5	LAW2004	Introduction to Contracts	2	0	0	2	2	-	F
6	LAW2005	Introduction to Copy Rights Law	2	0	0	2	2	-	F
7	LAW2006	Introduction to Criminal Law	2	0	0	2	2	-	F
8	LAW2007	Introduction to Insurance Law	2	0	0	2	2	-	F
9	LAW2008	Introduction to Labour Law	2	0	0	2	2	-	F
10	LAW2009	Introduction to Law of Marriages	2	0	0	2	2	-	F
11	LAW2010	Introduction to Patent Law	2	0	0	2	2	-	F
12	LAW2011	Introduction to Personal Income Tax	2	0	0	2	2	-	F
13	LAW2012	Introduction to Real Estate Law	2	0	0	2	2	-	F

14	LAW2013	Introduction to Trademark Law	2	0	0	2	2	-	F
15	LAW2014	Introduction to Competition Law	3	0	0	3	3	-	F
16	LAW2015	Cyber Law	3	0	0	3	3	-	F
17	LAW2016	Law on Sexual Harrassment	2	0	0	2	2	-	F
18	LAW2017	Media Laws and Ethics	2	0	0	2	2	-	F
Mathematics Basket									
1	MAT2008	Mathematical Reasoning	3	0	0	3	3	-	S
2	MAT2014	Advanced Business Mathematics	3	0	0	3	3	-	S
3	MAT2041	Functions of Complex Variables	3	0	0	3	3	-	S
4	MAT2042	Probability and Random Processes	3	0	0	3	3	-	S
5	MAT2043	Elements of Number Theory	3	0	0	3	3	-	S
6	MAT2044	Mathematical Modelling and Applications	3	0	0	3	3	-	S
Mechanical Engineering Basket									
1	MEC1001	Fundamentals of Automobile Engineering	3	0	0	3	3	-	F
2	MEC1002	Introduction to Matlab and Simulink	3	0	0	3	3	-	S/EM
3	MEC1003	Engineering Drawing	1	0	4	3	5	-	S
4	MEC2001	Renewable Energy Systems	3	0	0	3	3	-	F
5	MEC2002	Operations Research & Management	3	0	0	3	3	-	F
6	MEC2003	Supply Chain Management	3	0	0	3	3	-	S/ EM/ EN
7	MEC2004	Six Sigma for Professionals	3	0	0	3	3	-	S/EM
8	MEC2005	Fundamentals of Aerospace Engineering	3	0	0	3	3	-	F
9	MEC2006	Safety Engineering	3	0	0	3	3	-	S/EM
10	MEC2007	Additive Manufacturing	3	0	0	3	3	-	F/EM
11	MEC3069	Engineering Optimisation	3	0	0	3	3	-	S/EM
12	MEC3070	Electronics Waste Management	3	0	0	3	3	-	F/S
13	MEC3071	Hybrid Electric Vehicle Design	3	0	0	3	3	-	S/EM
14	MEC3072	Thermal Management of Electronic Appliances	3	0	0	3	3	-	S/EM
15	MEC3200	Sustainable Technologies and Practices	3	0	0	3	3	-	S/EM
16	MEC3201	Industry 4.0	3	0	0	3	3	-	S/EM
Petroleum Engineering Basket									
1	PET1005	Geology for Engineers	2	0	0	2	2		
2	PET1006	Overview of Energy Industry	2	0	0	2	2		ES / HP
3	PET1007	Introduction to Energy Trading and Future Options	2	0	0	2	2		ES / HP
4	PET1008	Sustainable Energy Management	2	0	0	2	2		ES / HP
5	PET2026	Introduction to Computational Fluids Dynamics	3	0	0	3	3		HP
6	PET2028	Polymer Science and Technology	3	0	0	3	3		ES / HP

7	PET2031	Overview of Material Science	3	0	0	3	3		ES / HP
8	PET2032	Petroleum Economics	3	0	0	3	3		ES / HP
Physics Basket									
1	PHY1003	Mechanics and Physics of Materials	3	0	0	3	3		FC / SD
2	PHY1004	Astronomy	3	0	0	3	3		FC
3	PHY1005	Game Physics	2	0	2	3	4		FC / SD
4	PHY1006	Statistical Mechanics	2	0	0	2	2		FC
5	PHY1007	Physics of Nanomaterials	3	0	0	3	3		FC
6	PHY1008	Adventures in nanoworld	2	0	0	2	2		FC
7	PHY2001	Medical Physics	2	0	0	2	2		FC
8	PHY2002	Sensor Physics	1	0	2	2	3		FC / SD
9	PHY2003	Computational Physics	1	0	2	2	3		FC
10	PHY2004	Laser Physics	3	0	0	3	3		FC
11	PHY2005	Science and Technology of Energy	3	0	0	3	3		FC
12	PHY2009	Essentials of Physics	2	0	0	2	2		FC
Management Basket									
1	MGT1001	Introduction to Psychology	3	0	0	3	3	-	F
2	MGT1002	Business Intelligence	3	0	0	3	3	-	EN
3	MGT1003	NGO Management	3	0	0	3	3	-	S
1	MGT1004	Essentials of Leadership	3	0	0	3	3	-	EM/EN
2	MGT1005	Cross Cultural Communication	3	0	0	3	3	-	S/EM/EN
3	MGT2001	Business Analytics	3	0	0	3	3	-	S/EM/EN
4	MGT2002	Organizational Behaviour	3	0	0	3	3	-	F
5	MGT2003	Competitive Intelligence	3	0	0	3	3	-	S
6	MGT2004	Development of Enterprises	3	0	0	3	3	-	S/ EM/EN
7	MGT2005	Economics and Cost Estimation	3	0	0	3	3	-	S/EM
8	MGT2006	Decision Making Under Uncertainty	3	0	0	3	3	-	S
9	MGT2007	Digital Entrepreneurship	3	0	0	3	3	-	S/EM/EN
10	MGT2008	Econometrics for Managers	3	0	0	3	3	-	S
11	MGT2009	Management Consulting	3	0	0	3	3	-	S/EM/EN
12	MGT2010	Managing People and Performance	3	0	0	3	3	-	S/EM/EN
13	MGT2011	Personal Finance	3	0	0	3	3	-	F
14	MGT2012	E Business for Management	3	0	0	3	3	-	S/EM
15	MGT2013	Project Management	3	0	0	3	3	-	EN/EM

16	MGT2014	Project Finance	3	0	0	3	3	-	EN/EM
17	MGT2015	Engineering Economics	3	0	0	3	3	-	S
18	MGT2016	Business of Entertainment	3	0	0	3	3	-	EN / EM
19	MGT2017	Principles of Management	3	0	0	3	3	-	S/EM/EN
20	MGT2018	Professional and Business Ethics	3	0	0	3	3	-	S/EM/EN
21	MGT2019	Sales Techniques	3	0	0	3	3	-	S/EM/EN
22	MGT2020	Marketing for Engineers	3	0	0	3	3	-	S/EM/EN
23	MGT2021	Finance for Engineers	3	0	0	3	3	-	S/EM/EN
24	MGT2022	Customer Relationship Management	3	0	0	3	3	-	S/EM/EN
25	MGT2023	People Management	3	0	0	3	3	-	S/EM/EN
Media Studies Basket									
1	BAJ3050	Corporate Filmmaking and Film Business	0	0	4	2	4	-	EM
2	BAJ3051	Digital Photography	2	0	2	3	4	-	EM
Research URE Basket									
1	URE2001	University Research Experience	-	0	-	3	0		S/EM/EN
2	URE2002	University Research Experience	-	0	-	0	0		S/EM/EN

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 Enginnering Block Chain

Table 3.5 : MOOC Discipline Elective Courses

Sl.No	Course Code	Course Name	Credits	L-T-P-C
1	CSE3111	Artificial Intelligence: Search Methods for Problem Solving	3	3-0-0-3
2	CSE3112	Privacy and Security in Online Social Media	3	3-0-0-3
3	CSE3113	Computational Complexity	3	3-0-0-3
4	CSE3114	Deep Learning for Computer Vision	3	3-0-0-3
5	CSE3115	Learning Analytics Tools	3	3-0-0-3
6	CSE505	The Joy of Computing Using Python	3	3-0-0-3
7	CSE3119	Coding Skills in Python	3	3-0-0-3
8	CSE3121	Parallel Computer Architecture	3	3-0-0-3
9	CSE3124	Games and Information	3	3-0-0-3
10	CSE3140	Introduction to Industry 4.0 and Industrial Internet of Things	3	3-0-0-3
11	CSE3142	Affective Computing	3	3-0-0-3
12	CSE3196	Foundations of Cyber Physical Systems	3	3-0-0-3
13	CSE3197	Getting Started with Competitive Programming	3	3-0-0-3
14	CSE3198	GPU Architectures and Programming	3	3-0-0-3
15	CSE3199	Artificial Intelligence: Knowledge Representation and Reasoning	3	3-0-0-3
16	CSE3200	Programming in Modern C++	3	3-0-0-3
17	CSE3201	Circuit Complexity Theory	3	3-0-0-3
18	CSE3202	Basics of Computational Complexity	3	3-0-0-3

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

21.3 MOOC - Open Elective Courses for B. Tech. Computer Science and Enginnering Block Chain

Table 3.6: MOOC Open Elective Courses
Courses duration is 4 weeks (01 credit)/ 8 weeks (02 credits)/ 12 weeks (03 credits)

Sl. No.	Course Code	Course Name	Total Credits	L-T-P-C
1	BBA2022	Supply Chain digitization	3	3-0-0-3
2	BBA2021	E Business	3	3-0-0-3
3	BBB2016	Business Analytics for Management Decisions	3	3-0-0-3
4	BBB2015	Artificial Intelligence for Investments	3	3-0-0-3
5	MEC3001	Design and Development of Product	1	1-0-0-1
6	ENG3004	Perspectives of Neurolinguistics	1	1-0-0-1
7	PPS4009	Working in Contemporary Teams	1	1-0-0-1
8	MGT3001	Data Analysis and Decision Making	3	3-0-0-3
9	MEC3001	Design and Development of Product	1	1-0-0-1
10	EEE3105	Microsensors and Nanosensors	3	3-0-0-3
11	CIV3065	Drone Systems and 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

Sl. No.	Course Code	Course Name	L	T	P	Credits	Contact Hours	Type of Skill	Basket
Semester 1 - Physics Cycle			11	1	16	17	28		
1	MAT1001	Calculus and Linear Algebra	3	0	2	4	5	EM	School Core
2	PHY1002	Optoelectronics and Device Physics	2	0	2	3	4	F	School Core
3	ECE1001	Elements of Electronics Engineering	3	0	2	4	5	F	School Core
4	ENG1002	Technical English	1	0	2	2	3	S	School Core
5	PPS1001	Introduction to soft skills	0	0	2	1	2	S	School Core
6	CSE1004	Problem Solving Using C	1	0	4	3	5	S	School Core
7	CHE1018	Environmental Science	1	0	2	0	3	F	School Core
8	PPS1011	Introduction to Verbal Ability	0	1	0	0	1	S	School Core
Semester 2 - Engineering Science Cycle			10	0	10	16	20		
1	MAT1003	Applied Statistics	1	0	2	2	3	EM	School Core
2	ECE2007	Digital Design	2	0	2	3	4	F	School Core
3	CIV1008	Basic Engineering Sciences	2	0	0	2	2	F	School Core
4	MEC1006	Engineering Graphics	2	0	0	2	2	F	School Core
5	CSE1001	Problem Solving using JAVA	2	0	2	3	4	S	School Core
6	ENG2001	Advanced English	1	0	2	2	3	S	School Core
7	PPS1012	Enhancing Personality through Soft Skills	0	0	2	1	2	S	School Core
8	ECE2010	Innovative Projects using Arduino	-	-	-	1	0	S	School Core
Semester 2 - Engineering Science Cycle (Cycle 2)			12	1	12	18			
1	MAT1001	Calculus and Linear Algebra	3	0	2	4	**5**	F	School Core
2	ECE1001	Elements of Electronics Engineering	3	0	2	4	**5**	F	School Core
3	ENG1002	Technical English	1	0	2	2	**3**	S	School Core
4	PPS1001	Introduction to soft skills	0	0	2	1	**2**	S	School Core
5	CSE1004	Problem Solving Using C	1	0	4	3	**5**	S	School Core
6	PPS1011	Introduction to Verbal Ability	0	1	0	0	**1**	S/ EM	School Core
7	CIV1008	Basic Engineering Sciences	2	0	0	2	**2**	S	School Core
8	MEC1006	Engineering Graphics	2	0	0	2	**2**	S	School Core
Semester 2 - Physics Cycle			8	0	16	15			
1	MAT1003	Applied Statistics	1	0	2	2	**3**	EM	School Core
2	ECE2007	Digital Design	2	0	2	3	**4**	F / S	School Core

3	CSE1006	Problem Solving using JAVA	1	0	4	3	**5**	S	School Core
4	ENG2001	Advanced English	1	0	2	2	**3**	S	School Core
5	PPS1012	Enhancing Personality Through Soft Skills	0	0	2	1	**2**	S	School Core
6	CHE1018	Environmental Science	1	0	2	0	**3**	F	School Core
7	PHY1002	Optoelectronics and Device Physics	2	0	2	3	**4**	F	School Core
8	ECE2010	Innovative Projects Using Arduino	-	-	-	1	**0**	S	School Core
Semester 3			21	0	24	28	33		
1	MAT1002	Transform Techniques, Partial Differential Equations and Their Applications	3	0	0	3	3	EM	School Core
2	CSE2001	Data Structures and Algorithms	3	0	2	4	5	S	School Core
3	CSE3155	Data Communications and Computer Networks	3	0	2	4	5	S	Program Core
4	CSE2009	Computer Organization and Architecture	3	0	0	3	3	S	Program Core
5	MAT2004	Discrete Mathematical Structures	3	0	0	3	3	S	School Core
6	CSE3190	Fundamentals of Data Analytics	2	0	2	3	4	S	Program Core
8	CSE2014	Software Engineering	3	0	0	3	3	S	Program Core
9	ECE2011	Innovative Projects Using Raspberry Pi	-	-	-	1	0	S	School Core
10	CSE1005	Programming in Python	1	0	4	3	5	S	Program Core
11	PPS4002	Introduction to Aptitude	0	0	2	1	2	S	School Core
Semester 4			21	0	6	23	51		
1	MAT2003	Numerical Methods for Engineers	1	0	2	2	3	EM	School Core
2	CSE2007	Design and Analysis of Algorithms	3	0	0	3	6	S	Program Core
3	CSE3156	Database Management Systems	3	0	2	4	9	S	Program Core
4	CSE3155	Operating Systems	3	0	0	3	6	S	Program Core
5	CSE3078	Cryptography and Network Security	3	0	0	3	6	S	Program Core
6	CBCXXXX	Discipline Elective - I	3	0	0	3	6	S	Discipline Elective
7	XXXXXXX	Open Elective – I (Management Basket)	3	0	0	3	6	S	Open Elective
8	PPS4004	Aptitutde Training Intermediate	0	0	2	1	3	S	School Core
9	CSE3216	Mastering Object-Oriented Concepts in Python	0	0	2	1	3	S	School Core
Semester 5			20	0	6	26	30		
1	CBC2500	Smart Contract and Solidity	3	0	0	3	3	S	Program Core
2	CBC2502	Distributed Ledger Technology	3	0	0	3	3	S	Program Core
3	CSE2264	Essentials of AI	3	0	0	3	3	S	Program Core

4	CBC2509	Consensus Algorithms and Network Design	3	0	0	3	3	S	Program Core
5	CBCXXXX	Discipline Elective - II	3	0	0	3	3	S	Discipline Elective
6	CBC2501	Smart Contract and Solidity Lab	0	0	2	1	2	S	Program Core
7	CBC2503	Distributed Ledger Technology Lab	0	0	2	1	2	S	Program Core
8	CSE2265	Essentials of AI Lab	0	0	2	1	2	S	Program Core
9	CSE2266	Theory of Computation	3	0	0	3	3	S	Program Core
10	CSE7000	Internship	-	-	-	2	0	S	School Core
11	CSE2500	Data Analytics	2	0	0	2	2	S	Program Core
12	APT4006	Logical and Critical Thinking	0	0	2	0	2	S	School Core
13	CIV7601	Universal Human Values and Ethics	0	0	0	0	0	S	School Core
14	CSE2501	Data Analytics Lab	0	0	2	1	2	S	Program Core
Semester 6			16	0	12	23	31		
1	CBC2507	Token Standards and Creation	3	0	0	3	3	S	Program Core
2	CBC2508	Token Standards and Creation Lab	0	0	2	1	2	S	Program Core
3	CBC2504	Blockchain Security and Performance	3	0	0	3	3	S	Program Core
4	CBC2505	Blockchain Security and Performance Lab	0	0	4	2	4	S	Program Core
5	CBC2506	Blockchain Architecture Design	3	0	0	3	3	S	Program Core
6	CBCXXXX	Discipline Elective - III	3	0	0	3	3	S	Discipline Elective
7	CBCXXXX	Open Elective – II	3	0	0	3	3	S	Open Elective
8	PPSXXXX	Industry Preparedness Program	2	0	0	0	2	S	School Core
9	CSE2274	Competitive Programming and Problem Solving	0	0	4	2	4	S	School Core
10	CSE2272	Cloud Computing	2	0	0	2	2	S	Program Core
11	CSE2273	Cloud Computing Lab	0	0	2	1	2	S	Program Core
12	LAW7601	Indian Constitution	0	0	0	0	-	S	School Core
Semester 7			12	0	0	17	12		
1	CSEXXXX	Professional Elective – IV	3	0	0	3	3	S	Discipline Elective
2	CSEXXXX	Professional Elective – V	3	0	0	3	3	S	Discipline Elective
3	CSEXXXX	Professional Elective – VI	3	0	0	3	3	S	Discipline Elective
4	XXXXXXX	Open Elective – III	3	0	0	3	3	S	Open Elective
5	CSE7101	Mini Project	0	0	0	5	0	S	School Core
Semester 8			0	0	0	10	0		
1	CSE7300	Capstone Project	0	0	0	10	0	S/EM	School Core
			117	5	54	160	205		

IV. Course Catalogues:

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

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

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

Course Code: MAT1001	Course Title: Calculus and Linear Algebra Type of Course: School Core Lab Integrated	L-T-P- C	2	1	2	4
Version No.	3.0					
Course Pre-requisites	Basic Concepts of Limits, Differentiation, Integration					
Anti-requisites	NIL					
Course Description	The course focuses on the concepts of calculus and linear algebra with reference to specific engineering problems. The course is of both conceptual and analytical type in nature. The lab sessions associated with the course are concerned with acquiring an ability to use the MATLAB software.					
Course Objective	The objective of the course is Skill Development of student by using Problem Solving Techniques.					
Course Out Comes	On successful completion of the course the students shall be able to: 1) Comprehend the knowledge of applications of matrix principles. 2) Understand the concept of partial derivatives and their applications. 3) Apply the principles of integral calculus to evaluate integrals. 4) Adopt the various analytical methods to solve differential equations. 5) Demonstrate the use of MATLAB software to deal with a variety of mathematical problems.					
Course Content:						
Module 1	Linear Algebra					10 Classes
Review: Types of matrices, elementary transformations, rank of a matrix, normal form, Solution of systems of linear equations: (Homogenous and non-homogenous system) AX = O and AX = B using rank method. Linear Algebra:						

<table>
<tr><td colspan="5">Eigenvalues and Eigenvectors of a real matrix – Characteristic equation – Properties of Eigenvalues and Eigenvectors – Cayley-Hamilton theorem – Diagonalization of matrices – Reduction of a quadratic form to canonical form by orthogonal transformation – Nature of quadratic forms.
Engineering Applications of Linear Algebra.</td></tr>
<tr><td>Module 2</td><td>Partial Derivatives</td><td></td><td></td><td>10 CLASSES</td></tr>
<tr><td colspan="5">Review: Differential calculus with single variable.

Partial Derivatives:
Homogeneous functions and Euler‘s theorem, Total derivative, Change of variables, Jacobians, Partial differentiation of implicit functions, Taylor‘s series for functions of two variables, Maxima and minima of functions of two variables, Lagrange‘s method of undetermined multipliers.
Engineering Applications of partial derivatives.</td></tr>
<tr><td>Module 3</td><td>Advanced Integral calculus</td><td></td><td></td><td>12 Classes</td></tr>
<tr><td colspan="5">Review: Integral calculus for single integrals.

Advanced Integral calculus:
Beta and Gamma functions–interrelation-evaluation of integrals using gamma and beta functions; error function-properties. Multiple Integrals- Double integrals – Change of order of integration – Double integrals in polar coordinates – Area enclosed by plane curves, evaluation of triple integrals-change of variables between Cartesian and cylindrical and spherical polar co-ordinates.
Engineering applications of partial derivatives.</td></tr>
<tr><td>Module 4</td><td>Ordinary Differential Equations</td><td>Assignment</td><td>Programming</td><td>12 Classes</td></tr>
<tr><td colspan="5">Review: First order and first-degree Ordinary Differential Equations, Method of separation of variables, Homogeneous and Non- Homogeneous Equations reducible to Homogeneous form.

Linear Differential Equations, Bernoulli’s Differential Equation, Exact and Non- Exact Differential Equations, Higher order Differential Equation with constant coefficients and with right hand side of the form eax, sinax, cosax, eaxf(x), xnf(x) etc., Linear equations with variable coefficients such as Cauchy Equation and Lagrange’s Equation, D-operators and Inverse D- operators, Method of Variation of Parameters.
Engineering applications of differential equations.</td></tr>
<tr><td colspan="5">List of Laboratory Tasks:
Introductory Task: Introduction to usage of the software and simple programming tasks. [3 Sessions]</td></tr>
</table>

Experiment N0 1: Solution of Simple differentiation with single variable and use of chain Rule.
Experiment No. 2: Solution based on application of Tailors' Series using software
Experiment No. 3: Application of Maxima and Minima condition using software.
Experiment No. 4 Computation of different functions for a specific problem
Experiment No. 5 Computation of Area under a curve.
Experiment No. 6 Solution of a set of simultaneous equations in matrix method
Experiment No. 7 Computation of Eigen Values and Eigen Vectors.
Experiment No. 8 Solution of Partial Differential equation
Experiment No. 9 solution using Cauchy Equation and Lagrange's Equation

Targeted Application & Tools that can be used:
The contents of this course has direct applications in most of the core engineering courses for problem formulations, Problem Solution and system Design.
Tools Used: MatLab, Zylink.

Assignment:

List at least 3 sets of Matrix Applications concerning the respective branch of Engineering and obtain the solution using MATLAB.
Select any one simple differential equation pertaining to the respective branch of engineering, identify the dependent and independent variable – Obtain the solution and compare the solution sets by varying the values of the dependent variable.

Text Book
Sankara Rao, Introduction to Partial differential equations, Prentice Hall of India, edition, 2011
B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

References:
Victor Henner, Tatyana Belozerova, Mickhail Khenner, Ordinary and Partial Differential Equations, CRC Press, Edition, 2013.
Walter Ledermann, Multiple integrals, Springer, 1st edition
Lay, Linear Algebra ansd its applications, 3rd Ed., 2002, Pearson Education India.
Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition
MatLab usage manual
E-resources/ Web links:
1. https://nptel.ac.in/courses/109104124
2. https://nptel.ac.in/courses/111106051
3. https://nptel.ac.in/courses/111102137
4. https://www.cuemath.com/learn/mathematics/algebra-vs-calculus/
5. https://stanford.edu/~shervine/teaching/cs-229/refresher-algebra-calculus
6. https://math.hmc.edu/calculus/hmc-mathematics-calculus-online-tutorials/linear-algebra/
7. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
8. https://www.scu.edu.au/study-at-scu/units/math1005/2022/

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

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

Course Code: ECE1001	Course Title: Elements of Electronics Engineering Type of Course: School Core Theory & Integrated Laboratory	L-T-P-C	3	0	2	4
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	Nil					
Course Description	The purpose of this course is to enable the students to learn the fundamental concepts of electronic devices and circuits. The course aims at nurturing the students with the fundamental principles of electronics engineering, prevailing in various engineering applications. The nature of the course is conceptual and analytical which imparts knowledge of electronic components and their behavior under various operating conditions. The course develops thinking skills of the students, encouraging their quest for knowledge about electronic devices and their usage in higher semester courses. The associated laboratory provides an opportunity to validate the concepts taught in theory classes and enable the students to work with basic electronic circuits using electronics components.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Elements of Electronics Engineering and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING .					
Course Outcomes	On successful completion of this course the students shall be able to: Identify various electrical and electronic components and basic electrical laws. Explainapplications of Diodes and BJTs. Summarize the concepts of Digital Electronics and Communication Systems. Discuss the basic concepts of microprocessorand computer organization. Perform experiments to familiarizevarious Electrical & Electronic components and equipment. Verify Basic Electrical Circuit configurations and Laws.					
Course Content:						
Module 1	Basic Electrical and Electronic Components	Assignment / Quiz	Identification of Practical electronic and electrical components / Memory Recall based Quizzes	10 Sessions		
Topics: ELECTRICAL CIRCUITS AND LAWS:DC Circuits: Classification of Electrical Elements, Ohm's law, Series and Parallel Circuits, Kirchhoff's Voltage and Current laws, Power and Energy, Transformers and their types. ELECTRONIC MATERIALS AND COMPONENTS: Conductors, Insulators, Semi-Conductor Material, P-N Junction diode, Characteristics and Parameters, Ideal Diode approximations, DC load line.						

<table>
<tr><td>Module 2</td><td>Applications of Diodes and Introduction to BJT</td><td>Assignment / Quiz</td><td></td><td>Simulation Task/ Memory Recall based Quizzes</td><td>12 Sessions</td></tr>
<tr><td colspan="6">Topics:
RECTIFIERS: Half-wave rectifier, Two-diode Full-wave rectifier, Bridge rectifier, Capacitor filter circuit (only qualitative approach).
ZENER DIODE: Zener diode, Zener Characteristics, Zener diode as a voltage regulator.
BIPOLAR JUNCTION TRANSISTORS: BJT Construction and Operation, BJT Voltages and Currents, Common Base, Common Emitter Configuration and Characteristics, Current amplification Factor alpha and beta, DC Load line w.r.t. fixed bias circuit (Q-Point), AC Analysis.</td></tr>
<tr><td>Module 3</td><td>Digital Electronics and Communication System</td><td>Assignment / Quiz</td><td></td><td>Simulation Task / Memory Recall based Quizzes</td><td>13 Sessions</td></tr>
<tr><td colspan="6">Topics:
NUMBER SYSTEMS: Decimal Number System, Binary Number System, Hexadecimal Number System, Conversions: Binary to and from Hexadecimal; Hexadecimal to and from Decimal;1’s and 2’s Complement of Binary Numbers, Binary Addition.
BOOLEAN ALGEBRA: Boolean Laws and Theorems, De Morgan’s theorem. Digital Circuits: Logic gates, NOT Gate, AND Gate, OR Gate, XOR Gate, X-NOR Gate, NAND Gate, NOR Gate.
COMMUNICATION SYSTEM: Block diagram of communication system, Modulation: Definition of Modulation, Need of Modulation, Types of Modulation: Amplitude Modulation and Frequency Modulation (Waveforms only).</td></tr>
<tr><td>Module 4</td><td>Microprocessors and Computer Organization</td><td>Assignment / Quiz</td><td colspan="2">Memory recall based Quizzes</td><td>10 Sessions</td></tr>
<tr><td colspan="6">Topics:
INTEL 8085 MICROPROCESSOR: Basic Architecture and features of 8085 Microprocessor.
COMPUTER ORGANISATION:Basic structure of Computer Organisation describing the various Computer types, Functional Units, Basic Operational concepts, Bus Structures, Memory System: RAM and ROM.</td></tr>
<tr><td colspan="6">List of Laboratory Tasks:

Experiment No. 1:Study of Resistors, Measuring instruments and DC Power Supply.
Level 1:Identification of resistor values from color bands and verification with Multimeter.
Level 2:Connecting a resistive circuit to a DC Power Supply and observing the input and output values using Voltmeters, Ammeters and hence calculate resistance values.

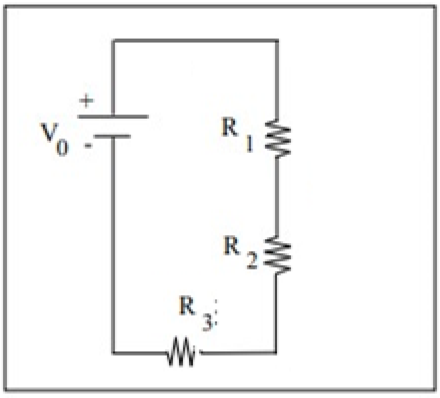
Experiment No. 2:Study of Reactive components, Multimeter, CRO and Function Generator.
Level 1:Identification of various types of capacitive and inductive components and verification with Multimeter.
Level 2:Connecting a reactive circuit to a function generator and observing the input and output waveform on CRO and calculation of Reactance and Impedance.

Experiment No. 3: Study of Ohm’s Law.
Level 1:Rig up the circuit and verify Ohm’s Law.</td></tr>
</table>

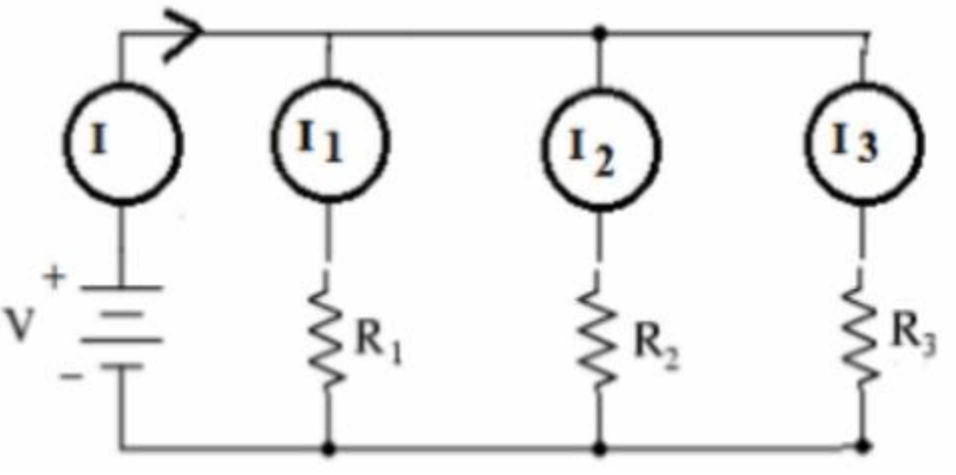
Level 2: Connect a 100Ω Resistor to a Voltage source of 0-5V. Plot a V- I graph by tabulating the Voltage Vs Current Values accordingly. Repeat the experiment for 1KΩ resistor and compare the results.

Experiment No. 4:Study of Series and Parallel Resistor Connections.
Level 1:Carry out the equivalent resistance of given four resistors 100Ω each connected in series and parallel combination using breadboard.
Level 2:Rig up a Current Divider Circuit and a Voltage Divider Circuit and verify the results.

Experiment No. 5:Study of Kirchhoff's Voltage Law and Kirchhoff's Current Law.
Level 1:Verify KVLand KCL with circuit(a) and circuit(b) with #values.

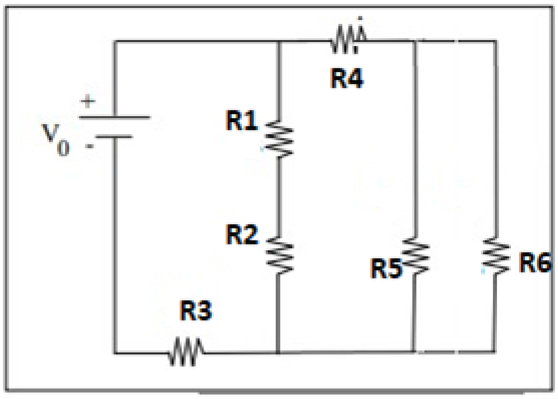


(a)



(b)

Level 2: Verify KCL with the help of given circuit having # values and carry out the equivalent resistance of the circuit by experimental and analytical methods.



Experiment No. 6: Study of PN-Junction Diode Characteristics in Forward and Reverse Bias Conditions.
Level 1:Carry out the experiment to find cut-in voltage on forward characteristics for Silicon P-N Junction diode.
Level 2: Carry out experiment to plot VI Characteristics of Silicon P-N Junction Diode in both forward and reverse biased conditions for Si P-N Junction diode.

Experiment No. 7: Study of Bipolar Junction Transistor in different regions of operation.
Level 1:Carry out the experiment to understand the importance of active, cut off and saturation regions.
Level 2: Carry out the experiment to design and analyze the operation of transistor as switch.

Experiment No. 8: Study of basic Digital Logic Gates using Integrated Chips IC's:
NOT, AND, OR, XOR, NAND and NOR Gates
Level 1:Carry out the experiment to study and verify the truth table of logic gates using Digital ICs.
Level 2:Implementation of operation of a basic Boolean expression using basic gates.

Experiment No. 9: Study of Computer Organization: Identification of Components on Motherboard: CPU: Processor Chips (Processor Socket), PCI, Parallel Ports, Universal Serial Bus: USB, I/O Connectors, RAM Slots.
Level 1:Carry out the experiment to familiarize a computer system layout and mark the positions of SMPS, Motherboard, FDD, HDD, CD / DVD drive and add on cards.
Level 2:Study of a Desktop PC and its assembling.

Targeted Application & Tools that can be used:
Student will be able to find career opportunities in various domains such asAnalog Electronics, Digital Electronics, Microprocessors, VLSI Design, Telecommunication, Computers andWireless Communication. The students will be able to join a profession which involves basics to high level of electronic circuit design.

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

Textbook(s):

T1. John Hiley, Keith Brown and Ian McKenzie Smith, "Hughes Electrical and Electronic Technology", Pearson,12th Edition

T2.William Stallings,“Computer Organization and Architecture Designing for Performance”, Pearson Education, 10th Edition.

Reference(s):

Reference Book(s):

R1. Smarajit Ghosh, “Fundamentals of Electrical and Electronics Engineering”, PHI, 2nd Edition

R2. D.P. Kothari, I. J. Nagrath, “Basic Electronics”, McGraw Hill Education, 1st Edition

R3. Rajendra Prasad, “Fundamentals of Electronics Engineering”, Cengane Learning, 3rd Edition

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

Video lectures on “BASIC ELECTRONICS” by Prof. Dr. Chitralekha Mahanta, Department of Electronics and communication Engineering, IIT Guwahati”: https://nptel.ac.in/courses/117/103/117103063/

Lecture Series on “ Useful Laws in Basic Electronics” by Prof. T.S.Natarajan, Department of physics, IIT Madras: https://www.youtube.com/watch?v=vfVVF58FtCc

Lecture Series on “Introduction to Bipolar Junction Transistors BJT ” by All About Electronics Youtube Channel: https://www.youtube.com/watch?v=-VwPSDQmdjM&list=PLwjK_iyK4LLDoFG8FeiKAr3IStRkPSxqq

Lecture Series on “ PN Junction Diode ” by All About Electronics Youtube Channel: https://www.youtube.com/watch?v=USrY0JspDEg

Lecture Series on “Introduction to Digital Electronics” by All About Electronics Youtube Channel: https://www.youtube.com/watch?v=DBTna2ydmC0&list=PLwjK_iyK4LLBC_so3odA64E2MLgIRKafl

Lecture Series on “Introduction to Microprocessors” by Bharat Acharya Education :https://www.youtube.com/watch?v=0M74z5jEAyA

Lecture Notes on : “Electronic Devices”, Bipolar Junction Transistors, 2nd Chapter, by Shree Krishna Khadka (PDF) Bipolar Junction Transistor (researchgate.net)https://www.researchgate.net/publication/323384291_Bipolar_Junction_Transistor

E-content:

V. Milovanovic, R. van der Toorn, P. Humphries, D. P. Vidal and A. Vafanejad, "Compact model of Zener tunneling current in bipolar transistors featuring a smooth transition to zero forward bias current," 2009 IEEE Bipolar/BiCMOS Circuits and Technology Meeting, 2009, pp. 99-102, doi: 10.1109/BIPOL.2009.5314134.
https://ieeexplore.ieee.org/document/5314134

M. Oueslati, H. Garrab, A. Jedidi and K. Besbes, "The advantage of silicon carbide material in designing of power bipolar junction transistors," 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15), 2015, pp. 1-6.
https://ieeexplore.ieee.org/document/7348149

H. Luo, F. Iannuzzo, F. Blaabjerg, X. Wang, W. Li and X. He, "Elimination of bus voltage impact on temperature sensitive electrical parameter during turn-on transition for junction temperature estimation of high-power IGBT modules," 2017 IEEE Energy Conversion Congress and Exposition (ECCE), 2017, pp. 5892-5898
https://ieeexplore.ieee.org/document/8096974

F. Bauer, I. Nistor, A. Mihaila, M. Antoniou and F. Udrea, "Super junction IGBT Filling the Gap Between SJ MOSFET and Ultrafast IGBT," in IEEE Electron Device Letters, vol. 33, no. 9, pp. 1288-1290, Sept. 2012 https://ieeexplore.ieee.org/document/6246672 https://presiuniv.knimbus.com/user#/home
Topics relevant to "SKILL DEVELOPMENT": Electrical & Electronic component and laws, Fundamentals of Digital Electronics, Communication Systems, Microprocessors and Computer Organization for Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: PHY1002	Course Title: Optoelectronics and Device Physics Type of Course: 1] School Core & Laboratory integrated	L-T-P-C	2-0-2-3
Version No.	1.0		
Course Pre-requisites	NIL		
Anti-requisites	NIL		
Course Description	The purpose of this course is to enable the students to understand the fundamentals, working and applications of optoelectronic devices and to develop the basic abilities to appreciate the applications of advanced microscopy and quantum computers. The course develops the critical thinking, experimental and analytical skills. The associated laboratory provides an opportunity to validate the concepts taught and enhances the ability to use the concepts for technological applications. The laboratory tasks aim to develop following skills: An attitude of enquiry, confidence and ability to tackle new problems, ability to interpret events and results, observe and measure physical phenomena, select suitable equipment, instrument and materials, locate faults in systems.		
Course Out Comes	On successful completion of the course the students shall be able to: CO1: Describe the concepts of semiconductors, magnetic materials and superconductors. CO2: Apply the concept of materials in the working of optoelectronic and magnetic devices. CO3: Discuss the quantum concepts used in advanced microscopy and quantum computers. CO4: Explain the applications of lasers and optical fibers in various technological fields. CO5: Interpret the results of various experiments to verify the concepts used in optoelectronics and advanced devices. [Lab oriented].		
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Optoelectronics and device physics "and attain Skill Development through Experiential Learning techniques		

Course Content:				
Module 1	Fundamentals of Materials.	Assignment	Plotting of magnetization (M) v/s Magnetic field (H) for diamagnetic, paramagnetic and ferromagnetic materials using excel/ origin software.	No. of Classes: 07
	Topics: Concept of energy bands, charge carriers, carrier concentration, concept of Fermi level, Hall effect, Magnetic materials, Superconductors:			
Module 2	Advanced Devices and applications	Assignment	Data collection on efficiency of solar cells.	No. of Classes: 8
	Topics: p-n junctions, Zener diode, transistor characteristics, Optoelectronic devices:, Solar cells, I-V characteristics, and LEDs			
Module 3	Quantum concepts and Applications	Term paper	Seminar on quantum computers.	No. of classes: 8
	Topics: Planck's quantum theory, applications of Quantum theory: de-Broglie hypothesis, matter waves, properties. de-Broglie wavelength associated with an electron. Heisenberg's uncertainty principle. Schrodinger time independent wave equation. Particle in a box			
Module 4	Lasers and Optical fibers	Term paper	Case study on medical applications of Lasers.	No. of classes :07
	Topics: Interactions of radiations with matter, Characteristics of laser, conditions and requisites of laser, Modern day applications of laser: LIDAR, LASIK, Cutting, Welding and Drilling. Principle of optical fibers, Numerical aperture and acceptance angle (Qualitative), Attenuation, Applications: Point to point communication with block diagram, application of optical fibers in endoscopy.			
	List of Laboratory Tasks: Experiment No. 1: Experimental errors and uncertainty using excel Level 1: Calculation of accuracy and precision of a given data Level 2: propagation of errors in addition, subtraction, multiplication and division. Experiment N0 2: To determine the wavelength of semiconductor diode Laser and to estimate the particle size of lycopodium powder using diffraction. Level 1: Determination of Wavelength of Laser Level 2: Finding the particle size of lycopodium powder. Experiment No. 3: To determine the proportionality of Hall Voltage, magnetic flux density and the polarity of Charge carrier. Level 1: To determine the proportionality of Hall Voltage and magnetic flux density Level 2: To determine the polarity of Charge carrier. Experiment No. 4: To study the I-V characteristics of a given zener diode in forward and reverse bias conditions. Level 1: To study I –V characteristics of the given Zener diode in reverse bias and to determine break down voltage.			

	Level 2: To study I –V characteristics of the given Zener diode in forward bias and to determine knee voltage and forward resistance. Experiment No. 5: To study input and output characteristics of a given Transistor. Level 1: To determine the input resistance of a given transistor. Level 2: To determine current transfer characteristics and transistor parameters of a given transistor. Experiment No. 6: Determination of Fermi energy and Fermi temperature of a given metal and bimetallic wire. Level 1: Determination of Fermi energy and Fermi temperature of given metal wire. Level 2: Determination of Fermi energy and Fermi temperature of given bimetallic wire. Experiment No. 7: To study the current vs voltage characteristics of CdS photo-resistor at constant irradiance and To measure the photo-current as a function of the irradiance at constant voltage. Level 1 To study the current vs voltage characteristics of CdS photo-resistor at constant irradiance. Level 2: To measure the photo-current as a function of the irradiance at constant voltage. Experiment No. 8: To study the I-V characteristics and I-R characteristics of a solar cell as a function of the irradiance. Level 1: To study the I-V characteristics Level 2: I-R characteristics of a solar cell as a function of the irradiance. Experiment No. 9: Calculate the numerical aperture and study the losses that occur in optical fiber cable. . Level 1: Calculate the numerical aperture. Level 2: study the losses that occur in optical fiber cable. Experiment No. 10: To determine the magnetic susceptibility of a given diamagnetic and paramagnetic substances using Quincke’s method. Level 1: To determine the magnetic susceptibility of a given diamagnetic substance. Level 2: To determine the magnetic susceptibility of a given paramagnetic substance. Experiment No. 11: Plotting I-V characteristics in forward and reverse bias for LEDs and Determination of knee voltage. Level 1: Plotting I-V characteristics in forward and reverse bias for LEDs Level 2: Determination of knee voltage. Experiment No. 12: Determination of Stefan’s constant and verification of Stefan-Boltzmann Law. Level 1: Determination of Stefan’s constant Level 2: Verification of Stefan-Boltzmann Law.
	Targeted Application & Tools that can be used: Areas of application are optoelectronics industry, Solar panel technologies, quantum computing software, electronic devices using transistors and diodes, memory devices, endoscopy, SQUIDS in MRI, Advanced material characterizations using SEM and STM. Origin, excel and Mat lab soft wares for programming and data analysis.

	Project work/Assignment: Mention the Type of Project /Assignment proposed for this course
	Assessment Type Midterm exam Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screen shot accessing digital resource.) Quiz End Term Exam Self-Learning 1. Prepare a comprehensive report on non-conventional energy resources in Karnataka and their pros and cons. 2. Write a report on importance of quantum entanglement in supercomputers.
	Text Book Engineering Physics by Avadhanalu, Revised edition, S. Chand Publications,2018.
	References: 1. Elementary Solid state Physics: Principles and Applications by M.A. Omar, 1st Edition, Pearson Publications, 2002. 2. Principles of Quantum Mechanics by R Shankar, 2nd edition, springer Publications, 2011. 3. Optoelectronics: An Introduction by John Wilson and John Hawkes, 3rd edition, Pearson Publications, 2017. 4. Engineering Physics by Gaur and Gupta, Dhanpat Rai Publications, 2012. 5. Introduction to Quantum Mechanics, David J Griffiths, Cambridge University Press, 2019
	E-Resourses: https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=553045&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=833068&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=323988&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1530910&site=ehost-live https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=486032&site=ehost-live
	Topics relevant to “SKILL DEVELOPMENT”: Fundamentals of materials, Lasers and optical fibers. for Skill Development through Participative Learning Techniques. This is attained through the Assignment/ Presentation as mentioned in the assessment component in course handout.

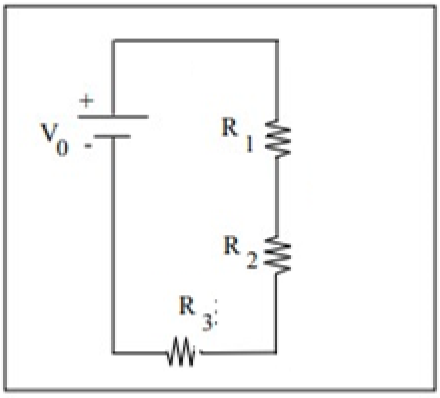
Course Code: ECE1001	Course Title: Elements of Electronics Engineering Type of Course: School Core Theory & Integrated Laboratory	L-T-P-C	3	0	2	4

Version No.	1.0				
Course Pre-requisites	NIL				
Anti-requisites	Nil				
Course Description	The purpose of this course is to enable the students to learn the fundamental concepts of electronic devices and circuits. The course aims at nurturing the students with the fundamental principles of electronics engineering, prevailing in various engineering applications. The nature of the course is conceptual and analytical which imparts knowledge of electronic components and their behavior under various operating conditions. The course develops thinking skills of the students, encouraging their quest for knowledge about electronic devices and their usage in higher semester courses. The associated laboratory provides an opportunity to validate the concepts taught in theory classes and enable the students to work with basic electronic circuits using electronics components.				
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Elements of Electronics Engineering and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING .				
Course Outcomes	On successful completion of this course the students shall be able to: Identify various electrical and electronic components and basic electrical laws. Explainapplications of Diodes and BJTs. Summarize the concepts of Digital Electronics and Communication Systems. Discuss the basic concepts of microprocessorand computer organization. Perform experiments to familiarizevarious Electrical & Electronic components and equipment. Verify Basic Electrical Circuit configurations and Laws.				
Course Content:					
Module 1	Basic Electrical and Electronic Components	Assignment / Quiz	Identification of Practical electronic and electrical components / Memory Recall based Quizzes		10 Sessions
Topics: ELECTRICAL CIRCUITS AND LAWS:DC Circuits: Classification of Electrical Elements, Ohm’s law, Series and Parallel Circuits, Kirchhoff’s Voltage and Current laws, Power and Energy, Transformers and their types. ELECTRONIC MATERIALS AND COMPONENTS: Conductors, Insulators, Semi-Conductor Material, P-N Junction diode, Characteristics and Parameters, Ideal Diode approximations, DC load line.					
Module 2	Applications of Diodes and Introduction to BJT	Assignment / Quiz		Simulation Task/ Memory Recall based Quizzes	12 Sessions

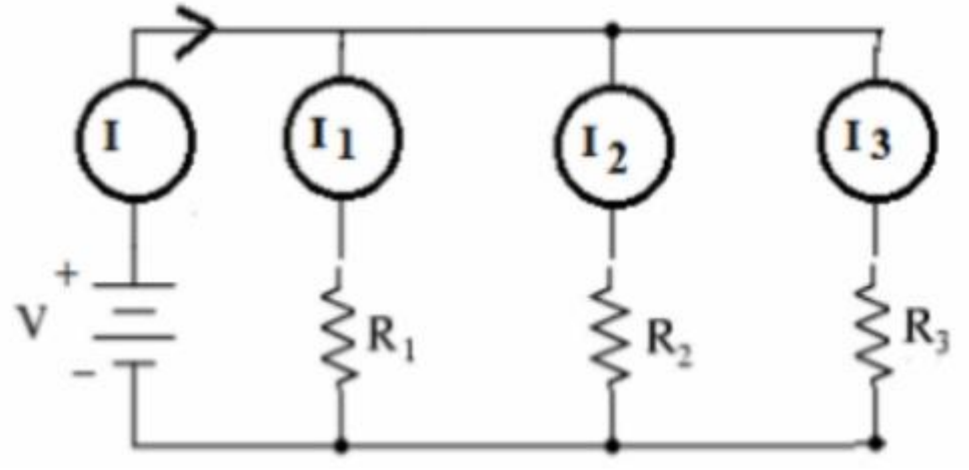
Topics: RECTIFIERS: Half-wave rectifier, Two-diode Full-wave rectifier, Bridge rectifier, Capacitor filter circuit (only qualitative approach). ZENER DIODE: Zener diode, Zener Characteristics, Zener diode as a voltage regulator. BIPOLAR JUNCTION TRANSISTORS: BJT Construction and Operation, BJT Voltages and Currents, Common Base, Common Emitter Configuration and Characteristics, Current amplification Factor alpha and beta, DC Load line w.r.t. fixed bias circuit (Q-Point), AC Analysis.					
Module 3	Digital Electronics and Communication System	Assignment / Quiz		Simulation Task / Memory Recall based Quizzes	13 Sessions
Topics: NUMBER SYSTEMS: Decimal Number System, Binary Number System, Hexadecimal Number System, Conversions: Binary to and from Hexadecimal; Hexadecimal to and from Decimal;1’s and 2’s Complement of Binary Numbers, Binary Addition. BOOLEAN ALGEBRA: Boolean Laws and Theorems, De Morgan’s theorem. Digital Circuits: Logic gates, NOT Gate, AND Gate, OR Gate, XOR Gate, X-NOR Gate, NAND Gate, NOR Gate. COMMUNICATION SYSTEM: Block diagram of communication system, Modulation: Definition of Modulation, Need of Modulation, Types of Modulation: Amplitude Modulation and Frequency Modulation (Waveforms only).					
Module 4	Microprocessors and Computer Organization	Assignment / Quiz	Memory recall based Quizzes		10 Sessions
Topics: INTEL 8085 MICROPROCESSOR: Basic Architecture and features of 8085 Microprocessor. COMPUTER ORGANISATION:Basic structure of Computer Organisation describing the various Computer types, Functional Units, Basic Operational concepts, Bus Structures, Memory System: RAM and ROM.					
List of Laboratory Tasks: Experiment No. 1:Study of Resistors, Measuring instruments and DC Power Supply. Level 1:Identification of resistor values from color bands and verification with Multimeter. Level 2:Connecting a resistive circuit to a DC Power Supply and observing the input and output values using Voltmeters, Ammeters and hence calculate resistance values. Experiment No. 2:Study of Reactive components, Multimeter, CRO and Function Generator. Level 1:Identification of various types of capacitive and inductive components and verification with Multimeter. Level 2:Connecting a reactive circuit to a function generator and observing the input and output waveform on CRO and calculation of Reactance and Impedance. Experiment No. 3: Study of Ohm’s Law. Level 1:Rig up the circuit and verify Ohm’s Law. Level 2: Connect a 100Ω Resistor to a Voltage source of 0-5V. Plot a V- I graph by tabulating the Voltage Vs Current Values accordingly. Repeat the experiment for 1KΩ resistor and compare the results. Experiment No. 4:Study of Series and Parallel Resistor Connections.					

Level 1:Carry out the equivalent resistance of given four resistors 100Ω each connected in series and parallel combination using breadboard.
Level 2:Rig up a Current Divider Circuit and a Voltage Divider Circuit and verify the results.

Experiment No. 5:Study of Kirchhoff's Voltage Law and Kirchhoff's Current Law.
Level 1:Verify KVLand KCL with circuit(a) and circuit(b) with #values.

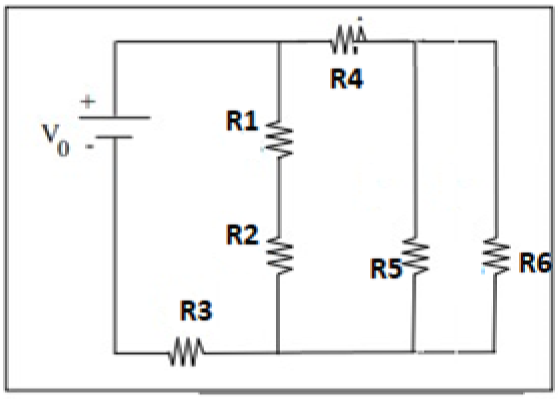


(a)



(b)

Level 2: Verify KCL with the help of given circuit having # values and carry out the equivalent resistance of the circuit by experimental and analytical methods.



Experiment No. 6: Study of PN-Junction Diode Characteristics in Forward and Reverse Bias Conditions.

Level 1:Carry out the experiment to find cut-in voltage on forward characteristics for Silicon P-N Junction diode.

Level 2: Carry out experiment to plot VI Characteristics of Silicon P-N Junction Diode in both forward and reverse biased conditions for Si P-N Junction diode.

Experiment No. 7: Study of Bipolar Junction Transistor in different regions of operation.

Level 1:Carry out the experiment to understand the importance of active, cut off and saturation regions.

Level 2: Carry out the experiment to design and analyze the operation of transistor as switch.

Experiment No. 8: Study of basic Digital Logic Gates using Integrated Chips IC's:
NOT, AND, OR, XOR, NAND and NOR Gates

Level 1:Carry out the experiment to study and verify the truth table of logic gates using Digital ICs.

Level 2:Implementation of operation of a basic Boolean expression using basic gates.

Experiment No. 9: Study of Computer Organization: Identification of Components on Motherboard: CPU: Processor Chips (Processor Socket), PCI, Parallel Ports, Universal Serial Bus: USB, I/O Connectors, RAM Slots.

Level 1:Carry out the experiment to familiarize a computer system layout and mark the positions of SMPS, Motherboard, FDD, HDD, CD / DVD drive and add on cards.

Level 2:Study of a Desktop PC and its assembling.

Targeted Application & Tools that can be used:

Student will be able to find career opportunities in various domains such asAnalog Electronics, Digital Electronics, Microprocessors, VLSI Design, Telecommunication, Computers andWireless Communication. The students will be able to join a profession which involves basics to high level of electronic circuit design.

Professionally Used Software: MultiSim/ PSpice

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

Textbook(s):

T1. John Hiley, Keith Brown and Ian McKenzie Smith, "Hughes Electrical and Electronic Technology", Pearson,12th Edition

T2.William Stallings,"Computer Organization and Architecture Designing for Performance", Pearson Education, 10th Edition.

Reference(s):

Reference Book(s):

R1. Smarajit Ghosh, "Fundamentals of Electrical and Electronics Engineering", PHI, 2nd Edition

R2. D.P. Kothari, I. J. Nagrath, "Basic Electronics", McGraw Hill Education, 1st Edition

R3. Rajendra Prasad, "Fundamentals of Electronics Engineering", Cengane Learning, 3rd Edition

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

Video lectures on “BASIC ELECTRONICS” by Prof. Dr. Chitralekha Mahanta, Department of Electronics and communication Engineering, IIT Guwahati”: https://nptel.ac.in/courses/117/103/117103063/

Lecture Series on “ Useful Laws in Basic Electronics” by Prof. T.S.Natarajan, Department of physics, IIT Madras: https://www.youtube.com/watch?v=vfVVF58FtCc

Lecture Series on “Introduction to Bipolar Junction Transistors BJT ” by All About Electronics Youtube Channel: https://www.youtube.com/watch?v=-VwPSDQmdjM&list=PLwjK_iyK4LLDoFG8FeiKAr3IStRkPSxqq

Lecture Series on “ PN Junction Diode ” by All About Electronics Youtube Channel: https://www.youtube.com/watch?v=USrY0JspDEg

Lecture Series on “Introduction to Digital Electronics” by All About Electronics Youtube Channel: https://www.youtube.com/watch?v=DBTna2ydmC0&list=PLwjK_iyK4LLBC_so3odA64E2MLgIRKafl

Lecture Series on “Introduction to Microprocessors” by Bharat Acharya Education :https://www.youtube.com/watch?v=0M74z5jEAyA

Lecture Notes on : “Electronic Devices”, Bipolar Junction Transistors, 2nd Chapter, by Shree Krishna Khadka (PDF) Bipolar Junction Transistor (researchgate.net)https://www.researchgate.net/publication/323384291_Bipolar_Junction_Transistor

E-content:

V. Milovanovic, R. van der Toorn, P. Humphries, D. P. Vidal and A. Vafanejad, "Compact model of Zener tunneling current in bipolar transistors featuring a smooth transition to zero forward bias current," 2009 IEEE Bipolar/BiCMOS Circuits and Technology Meeting, 2009, pp. 99-102, doi: 10.1109/BIPOL.2009.5314134. https://ieeexplore.ieee.org/document/5314134

M. Oueslati, H. Garrab, A. Jedidi and K. Besbes, "The advantage of silicon carbide material in designing of power bipolar junction transistors," 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15), 2015, pp. 1-6. https://ieeexplore.ieee.org/document/7348149

H. Luo, F. Iannuzzo, F. Blaabjerg, X. Wang, W. Li and X. He, "Elimination of bus voltage impact on temperature sensitive electrical parameter during turn-on transition for junction temperature estimation of high-power IGBT modules," 2017 IEEE Energy Conversion Congress and Exposition (ECCE), 2017, pp. 5892-5898 https://ieeexplore.ieee.org/document/8096974

F. Bauer, I. Nistor, A. Mihaila, M. Antoniou and F. Udrea, "Super junction IGBT Filling the Gap Between SJ MOSFET and Ultrafast IGBT," in IEEE Electron Device Letters, vol. 33, no. 9, pp. 1288-1290, Sept. 2012 https://ieeexplore.ieee.org/document/6246672

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

Topics relevant to “SKILL DEVELOPMENT”: Electrical & Electronic component and laws, Fundamentals of Digital Electronics, Communication Systems, Microprocessors and Computer Organization for Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: ENG1002	Course Title: Technical English Type of Course:1] School Core	L-T-P-C	1-0-2-2

	2] Laboratory integrated			
Version No.	V. 3			
Course Pre-requisites	Intermediate Level English			
Course Anti-requisites	NIL			
Course Description	Technical English course is designed to equip students with the language skills necessary for effective communication in technical and scientific contexts. The course focuses on the specialized vocabulary, writing styles, and communication techniques used in various technical fields, including engineering and information technology.			
Course Objectives	The objective of this course is to develop the learners' EMPLOYABILITY SKILLS by using EXPERIENTIAL LEARNING and PARTICIPATIVE LEARNING TECHNIQUES.			
Course Outcomes	On successful completion of the course, the students shall be able to: Develop proficiency in using technical vocabulary and terminology. Apply language skills for better speaking skills in technical fields. Write technical descriptions Demonstrate writing skills in writing technical documents such as reports, manuals, and articles.			
Course Content:				
Module 1	Fundamentals of Technical Communication	Worksheets& Quiz	Vocabulary building	9 Classes
Introduction to Technical English Differences between Technical English and General English Technical Writing Basics Technical Vocabulary				
Module 2	Technical Presentation	Presentations	Speaking Skills	12 Classes
Introduction Planning the Presentation Creating the Presentation Giving the Presentation				
Module 3	Technical Description	Assignment	Group Presentation	12 Classes
Product Description Process Description User Manuals Transcoding: Diagrams, charts and images				

<table>
<tr><td>Module 4</td><td>Technical Writing</td><td>Assignment</td><td>Writing Skills</td><td>12 Classes</td></tr>
<tr><td colspan="5">Email Writing
Persuasive and Descriptive Language
Professional Email Etiquette
Writing clear and concise technical emails
Communicating technical information effectively
Technical Report Writing
Types of technical reports (Lab reports, research reports, etc.)
Components of technical reports
Writing an abstract and executive summary
Structure and content organization
Transcoding: diagrams, charts and images</td></tr>
<tr><td colspan="5">List of Laboratory Tasks:
Module-1
Level 1: Worksheets
Level 2: Worksheets
Module 2
Level 1: Preparing Presentation
Level 2: Giving Presentation (Individual)
Module-3
Level 1: Product Description & User Manual
Level 2: Process Description & Transcoding
Module 4
Level 1: Email Writing
Level 2: Report Writing</td></tr>
<tr><td colspan="5">Targeted Applications & Tools that can be used:
Flipgrid
Quizzes
Youtube Videos
Podcast</td></tr>
<tr><td colspan="5">Project work/Assignment: Mention the Type of Project /Assignment proposed for this course

Bring out the essence of technical communication with reference to the conventions of technical communication, with examples
Prepare a technical presentation on the importance of Technical Communication and its relevance in a technical field, with real-life examples.</td></tr>
<tr><td colspan="5">The following individual, as well as group Assignments, will be given to the students.
Presentation
Describing a product/process
Individual Reports</td></tr>
<tr><td colspan="5">Text Books
Kumar, Sanjay; Pushpalatha. English Language and Communication Skills for Engineers. Oxford University Press. 2018.
Brieger, Nick and Alison Paul. Technical English Vocabulary and Grammar. https://nmetau.edu.ua/file/technical_english_vocabulary_and_grammar.pdf</td></tr>
</table>

Reference Book: Chauhan, Gajendra Singh, and Kashmiramka, Smita, Technical Communication. Cengage Publication. 2018. Sunder Jain. Technical Report Writing. Centrum Press, 2013. John Bowden. “Writing a Report: How to Prepare, Write & Present Really Effective Reports?”. 9th Edition 2011 Comfort, Jeremy et. al. 1984. Business Reports in English. Cambridge University Press. Sharma, R.C. and K. Mohan. 2011. Business Correspondence and Report Writing, Fourth Edition. Tata McGraw Hill.
Web Resources: 1:https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=JSTOR1_3307. 2;https://puniversity.informaticsglobal.com:2282/ehost/detail/detail?vid=5&sid=3a77d69b-abe5-4681-b39d-32dfdcb8f4a5%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=154223466&db=iih 3: Last,Suzan, et. al. Technical Writing Essentials. University of Victoria, British Columbia, 2019 (E- Book) 4 Wambui, Tabita Wangare, et al. Communication Skills- Volume 1, LAP LAMBRET, USA, 2012 (E Book)
Topics Relevant to the Development of Employability Skills: Speaking Skills, Writing Skills, Critical Thinking and Critical Analysis, and Group Communication.

Course Code: CSE1004	Course Title: Problem Solving Using C Type of Course: School Core Lab Integrated.	L-T-P-C	1	0	4	3
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs and applications in C. ACAlso by learning the basic programming constructs they can easily switch over to any other language in future.					
Course Object	The objective of the course is to familiarize the learners with the concepts of Problem Solving Using C and attain Employability through Problem Solving Methodologies.					

Course Outcomes	On successful completion of this course the students shall be able to: Write algorithms and to draw flowcharts for solving problems Demonstrate knowledge and develop simple applications in C programming constructs Develop and implement applications using arrays and strings Decompose a problem into functions and develop modular reusable code Solve applications in C using structures and Union Design applications using Sequential and Random Access File Processing.			
Course Content:				
Module 1	Introduction to C Language	Quiz	Problem Solving	9 Hrs.
Topics: Introduction to Programming – Algorithms – Pseudo Code - Flow Chart – Compilation – Execution – Preprocessor Directives (#define, #include, #undef) - Overview of C – Constants, Variables and Data types – Operators and Expressions – Managing Input and Output Operations – Decision Making and Branching - Decision Making and Looping.				
Module 2	Introduction to Arrays and Strings	Quiz	Problem Solving	9 Hrs.
Topics: Arrays: Introduction – One Dimensional Array – Initialization of One Dimensional Arrays – Example Programs – Sorting (Bubble Sort, Selection Sort) – Searching (Linear Search) - Two Dimensional Arrays – Initialization of Two Dimensional Arrays. Example Programs – Matrix operations. Strings: Introduction – Declaring and Initializing String Variables – Reading Strings from Terminal – Writing String to Screen – String Handling Functions.				
Module 3	Functions and Pointers	Quiz	Problem Solving	9 Hrs.
Topics: Functions: Introduction – Need for User-defined functions – Elements of User-Defined Functions: declaration, definition and function call–Categories of Functions – Recursion. Pointers: Introduction – Declaring Pointer Variables – Initialization of Variables – Pointer Operators – Pointer Arithmetic – Arrays and Pointers – Parameter Passing: Pass by Value, Pass by Reference.				
Module 4	Structures and Union	Quiz	Problem Solving	9 Hrs.
Topics: Structures: Introduction – Defining a Structure – Declaring Structure Variable – Accessing Structure Members – Array of Structures – Arrays within Structures – Union: Introduction – Defining and Declaring Union – Difference Between Union and Structure.				
Module 5	File handling	Case Study	Problem Solving	9 Hrs.
Topics: Files: Defining and Opening a File – Closing a File – Input / Output Operations on File – Random Access Files				

List of Practical Tasks Lab Sheet 1 (Module I)
Programs using IO Statements, Conditional Statements and Looping Statements
Lab Sheet 2 (Module II)
Programs using Arrays and Strings
Lab Sheet 3 (Module III)
Programs using Functions and Pointers
Lab Sheet 4 (Module IV)
Programs using Structures and Unions
Lab Sheet 5 (Module V)
Programs using Files

Text Book(s):
1. E. Balaguruswamy, “Programming in ANSI C”, 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.

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

Web Links and Video Lectures:
1. https://nptel.ac.in/courses/106/105/106105171/
2. https://archive.nptel.ac.in/courses/106/104/106104128/

Course Code: **PPS1012**	Course Title: **Enhancing Personality through Soft Skills** Type of Course: Practical Only Course	L- T - P- C	0	0	2	1
Version No.	1.0					
Course Pre-requisites	• Students are expected to understand Basic English. • Students should have the desire and enthusiasm to be involved, participate and learn.					
Anti-requisites	NIL					
Course Description	This course is designed to enable students to understand soft skills concepts and improve confidence, communication, and professional skills to give the students a competitive advantage and increase chances of success in the professional world. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.					
Course Objective	The objective of the course is to **familiarize the learners with the concepts of “Enhancing Personality through Soft Skills” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.**					

Course Out Comes	**On successful completion of this course, the students shall be able to:** CO 1 **Identify the stages of** team formation (Remember) CO 2 **Demonstrate** effective presentation skills (Apply) CO3 **Prepare** professional social media profile (Apply)		
Course Content:			
Module 1	Professional Brand Building	Brand Framework Activity	**6 Hours**
Topics: Personal brand definition, Crafting a compelling LinkedIn profile, Networking strategies, Leveraging AI tools for developing content for brand visibility. **Activity:** Create a post and enhancing LinkedIn profile			
Module 2	Art of Questioning	Role plays	**4 Hours**
Topics: Framing Questions, 5W1H Technique, Open-ended and Close-ended questions, Funnel technique, Probing questions, Leading questions			
Module 3	Presentation Skills	Practice and evaluation of individual/group presentation	**12 Hours**
Topics: Content development, Delivery techniques, Audience Analysis, Timing and Pacing, handling questions and challenges. **Activity:** Individual presentations or team presentation			
Module 4	Team Building	Team building activities	**6 Hours**
Topics: Importance of team, stages of Team Formation, Trust and collaboration. **Activity: Team Building Activity**			
Module 5	Recap / Revision /Feedback Session	Discussion, Quiz	**2 Hours**
Targeted Applications & Tools that can be used: 1. TED Talks 2. You Tube Links 3. Activities			
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course			

1) Presentation Evaluation

2) LinkedIn assessment

Targeted Applications & Tools that can be used:

1. TED Talks
2. YouTube Links
3. Videos by L&D Team shared on Edhitch/YouTube.com
4. LMS

Course Code: CHE1018	Course Title: Environmental Science Type of Course: School Core- Theory and Lab		L- T-P- C	1	0	2	0
Version No.	2.0						
Course Pre-requisites	NIL						
Anti-requisites	NIL						
Course Description	This course emphasizes the need to conserve biodiversity and adopt a more sustainable lifestyle by utilizing resources in a responsible way. Topics covered include basic principles of ecosystem functions; biodiversity and its conservation; human population growth; water resources, pollution; climate change; energy resources, and sustainability; Sustaining human societies, policies, and education. This course is designed to cater to Environment and Sustainability						
Course Objective	The objective of the course is to familiarize the learners with the concepts of "Environmental Science" and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques.						
Course Outcomes	On successful completion of this course the students shall be able to: Appreciate the historical context of human interactions with the environment and the need for eco-balance. Describe basic knowledge about global climate change with particular reference to the Indian context. Understand biodiversity and its conservation Develop an understanding on types of pollution and ways to protect the environment Learn about various strategies on Global environmental management systems						
Course Content:							
Module 1	Humans and the Environment	Assignment	Data Collection	01 class			

Topics: The man-environment interaction: Mastery of fire; Origin of agriculture; Emergence of city states; Great ancient civilizations and the environment. Self-learning topics: Humans as hunter-gatherers; Industrial revolution and its impact on the environment; Environmental Ethics and emergence of environmentalism.				
Module 2	Natural Resources and Sustainable Development	Assignment		03 Classes
Topics: Overview of natural resources: Definition of resource; Classification of natural resources-biotic and abiotic, renewable and non-renewable. Water resources: Types of water resources-fresh water and marine resources; Soil and mineral resources: Important minerals; Mineral exploitation Soil as a resource and its degradation. Energy resources: Sources of energy and their classification, renewable and non-renewable sources of energy; Advantages and disadvantages. Self- learning topics: Availability and use of water resources; Environmental impact of over-exploitation, issues and challenges.; Environmental problems due to extraction of minerals and use; Sustainable Development Goals (SDGs)- targets, indicators, and challenges for SDGs.				
Module 3	Environmental Issues: Local, Regional and Global	Case study		02 Classes
Topics: Environmental Pollution: Types of Pollution- air, noise, water, soil, municipal solid waste, hazardous waste; Trans- boundary air pollution; Acid rain; Smog. Land use and Land cover change: land degradation, deforestation, desertification, urbanization. Global change: Ozone layer depletion; Climate change Self -learning topics: Environmental issues and scales				

	Module 4	Conservation of Biodiversity and Ecosystems	Assignment		02 Classes
Topics: Biodiversity-Introduction, types, Species interactions, Extinct, endemic, endangered and rare species, Threats to biodiversity: Natural and anthropogenic activities. Self-learning topics: Mega-biodiversity, Hot-spots, Major conservation policies. Biodiversity loss: past and current trends, impact.					
	Module 5	Environmental Pollution and Health	Case study		03 Classes
Topics: Pollution, Definition, point and nonpoint sources of pollution, Air pollution- sources, major air pollutants, health impacts of air pollution. Water pollution– Pollution sources, adverse health impacts on human and aquatic life and mitigation, Water quality parameters and standards. Soil pollution and solid waste- Soil pollutants and their sources, solid and hazardous waste, Impact on human health. Self-learning topics: Noise pollution, Thermal and radioactive pollution.					

<table>
<tr><td></td><td>Module 6</td><td>Climate Change: Impacts, Adaptation
and Mitigation</td><td>Assignment/case</td><td></td><td>02 Classes</td></tr>
<tr><td colspan="6">Topics:
Understanding climate change: Natural variations in climate; Projections of global climate change with special reference to temperature, rainfall and extreme events; Importance of 1.5 °C and 2.0 °C limits to global warming; Impacts</td></tr>
<tr><td colspan="6">Vulnerability and adaptation to climate change: Observed impacts of climate change on ocean and land systems; Sea level rise, changes in marine and coastal ecosystems; Impacts on forests and natural ecosystems; Indigenous knowledge for adaptation to climate change.
Self-learning topics: Mitigation of climate change: Synergies between adaptation and mitigation measures; National and international policy instruments for mitigation.</td></tr>
<tr><td></td><td>Module 7</td><td>Environmental Management</td><td>Case study</td><td>Data analysis</td><td>02 Classes</td></tr>
<tr><td colspan="6">Topics:
Environmental management system: ISO 14001; Environmental risk assessment Pollution control and management; Waste Management- Concept of 3R (Reduce, Recycle and Reuse) and sustainability.

Self-learning topics: Environmental audit and impact assessment; Eco labeling /Eco mark scheme</td></tr>
<tr><td></td><td>Module 8</td><td>Environmental Treaties and Legislation</td><td>Case study</td><td>Data analysis</td><td>01 Classes</td></tr>
<tr><td colspan="6">Topics:
Major International Environmental Agreements: Convention on Biological Diversity (CBD), Major Indian Environmental Legislations: Environmental Protection Act, Forest Conservation Act, Public awareness.

Self-learning topics: Paris Agreement, Conference of the Parties (COP), India’s status as a party to major conventions: Air (Prevention and Control of Pollution) Act, Water (Prevention and control of Pollution) Act, Wildlife Protection Act.</td></tr>
<tr><td colspan="6">List of laboratory tasks : Any eight experiments will be conducted
Determination of total alkalinity of a water sample (knowledge)
Estimation of water hardness by EDTA method and its removal (by zeolite/ ion exchange method) (Comprehensive)
Estimation of copper from industrial effluents by colorimetric method (Comprehensive)
Estimation of iron from industrial effluents by titrimetric method/potentiometric method (Comprehensive)
Estimation of nickel from industrial effluents by titrimetric method (Comprehensive)
Estimation of chloride in drinking water by titrimetric method (Comprehensive)
Estimation of fluoride in ground water by colorimetric method (Comprehensive)
Determination of calcium in aqueous solution (Comprehensive)
Determination of Total Dissolved Salts, conductivity and pH of a water samples (Knowledge)
Determination of Chemical oxygen demand in the industrial effluent. (Comprehensive)
Biological oxygen demand of waste water sample (Comprehensive)
Determination of dissolved oxygen of an industrial effluent (Comprehensive)
Quality monitoring analysis of a soil sample (knowledge)
Flame photometric estimation of Sodium and potassium (Application)</td></tr>
</table>

Gas Chromatographic analysis of volatile organic compounds (Application)
Targeted Application & Tools that can be used: Application areas are Energy, Environment and sustainability Tools: Statistical analysis of environmental pollutants using excel, origin etc.
Project work/Assignment:
Assessment Type Midterm exam Assignment (review of digital/ e-resource from PU link given in references section - mandatory to submit screenshot accessing the digital resource.) Lab evaluation/Assignment End Term Exam Self-learning Assignment 1: Write a Statement of Environment report of your town/city/state/country
Assignment 2: Individual students will carry out the analyses of polluted solid, liquid, and gaseous samples and propose suitable mitigation measures. A detailed and in-depth report needs to be submitted for each case. This may include preparation of reagents, sample preparation (extraction), chemical analysis carried out, instruments and tools used, data collected and processed, inferences made and conclusions arrived at. Necessary support is given in the form of lab manual and reference links to e-books.
Text Book G. Tyler Miller and Scott Spoolman (2020), Living in the Environment, 20th Edition, Cengage Learning, USA Krishnamurthy, K.V. (2003) Text book of Biodiversity, Science Publishers, Plymouth, UK. Jackson, A.R. & Jackson, J.M. (2000), Environmental Science: The natural environment and human impact, Pearson Education.

Reference Books

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

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

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

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

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

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

E-resources:

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AB_1_06082022_18126

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AB_1_06082022_8761

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AJ_1_02082022_3333

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AB_1_06082022_3063

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AB_1_06082022_20719

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AB_1_06082022_16824

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AB_1_06082022_3954

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=DO AB_1_06082022_491

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=CU STOM_PACKAGE_16012023_WORLD_BUSINESS_COUNCIL_SUSTAINABLE_488

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=CU STOM_PACKAGE_16012023_WORLD_BUSINESS_COUNCIL_SUSTAINABLE_583

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=SP RINGER_INDEST_1_171

https://presiuniv.knimbus.com/user#/searchresult?searchId=3R%20principle&_t=1687427221129

https://presiuniv.knimbus.com/user#/searchresult?searchId=eco%20labelling&_t=1687427279979

https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=TE XTBOOK_LIBRARY01_06082022_395&xIndex=4

https://www.ugc.gov.in/oldpdf/modelcurriculum/env.pdf

Topics relevant to Skill Development:
Industrial revolution and its impact on the environment, Environmental impact of over-exploitation of water resources, pollution and ill effects, lab experiments for Skills development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.
All topics in theory component are relevant to Environment and Sustainability.

Course Code: PPS 1001	Course Title: Introduction to Soft Skills Type of Course: Practical Only Course	L- T-P- C	0-0-2-1	
Version No.	1.0			
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.			
Anti-requisites	NIL			
Course Description	This course is designed to enable students understand soft skills concepts and improve confidence, communication and professional skills to give the students a competitive advantage and increase chances of success in the professional world. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Soft Skills” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.			
Course Out Comes	On successful completion of this course the students shall be able to: CO1: Recognize significance of soft skills CO2: Illustrate effective communication while introducing oneself and others CO3: List techniques of forming healthy habits CO4: Apply SMART technique to achieve goals and increase productivity			
Course Content:				
Module 1	INTRODUCTION TO SOFT SKILLS		Classroom activity	04 Hours
Topics: Setting Expectations, Ice Breaker, Significance of soft skills, Formal grooming, punctuality				
Module 2	EFFECTIVE COMMUNICATION		Individual Assessment	10 Hours
Topics: Different styles of communication, Difference between hearing and listening, Effective communication for success, Email etiquette, Self-introduction framework, Video introduction, email- writing, Resume Building- Digital, Video, Traditional.				
Module 3	HABIT FORMATION		Worksheets & Assignment	4 Hours

Topics: Professional and personal ethics for success, Identity based habits, Domino effect, Habit Loop, Unlearning, standing up for what is right				
Module 4	Goal setting & Time Management		Goal sheet	8 Hours
A session where students will be introduced to Time management, setting SMART Goals, Introduction to OKR Techniques, Time Management Matrix, steps to managing time through outbound group activity, making a schedule, Daily Plan and calendars (To Do List), Monitoring/charting daily activity				
Targeted Application & Tools that can be used: LMS				
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course				
Individual Assessment LMS MCQ				
The topics related to Skill Development: Communication and professional grooming, Goal setting and presentation for skill development through participative learning techniques. This is attained through assessment component mentioned in course handout.				

Course Code: CSE1004	Course Title: Problem Solving Using C Type of Course: School Core Lab Integrated.	L- T-P-C	1	0	4	3
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs and applications in C. AC Also by learning the basic programming constructs they can easily switch over to any other language in future.					
Course Object	The objective of the course is to familiarize the learners with the concepts of Problem Solving Using C and attain Employability through Problem Solving Methodologies.					
Course Outcomes	On successful completion of this course the students shall be able to: Write algorithms and to draw flowcharts for solving problems Demonstrate knowledge and develop simple applications in C programming constructs Develop and implement applications using arrays and strings Decompose a problem into functions and develop modular reusable code Solve applications in C using structures and Union Design applications using Sequential and Random Access File Processing.					
Course Content:						

Module 1	Introduction to C Language	Quiz	Problem Solving	9 Hrs.

<table>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 2</td><td>Introduction to Arrays and Strings</td><td>Quiz</td><td>Problem Solving</td><td>9 Hrs.</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 3</td><td>Functions and Pointers</td><td>Quiz</td><td>Problem Solving</td><td>9 Hrs.</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 4</td><td>Structures and Union</td><td>Quiz</td><td>Problem Solving</td><td>9 Hrs.</td></tr>
<tr><td colspan="5">Topics:
Structures: Introduction – Defining a Structure – Declaring Structure Variable – Accessing Structure Members – Array of Structures – Arrays within Structures – Union: Introduction – Defining and Declaring Union – Difference Between Union and Structure.</td></tr>
<tr><td>Module 5</td><td>File handling</td><td>Case Study</td><td>Problem Solving</td><td>9 Hrs.</td></tr>
<tr><td colspan="5">Topics:
Files: Defining and Opening a File – Closing a File – Input / Output Operations on File – Random Access Files</td></tr>
<tr><td colspan="5">List of Practical Tasks Lab Sheet 1 (Module I)
CHE1018
Lab Sheet 2 (Module II)
Programs using Arrays and Strings
Lab Sheet 3 (Module III)
Programs using Functions and Pointers
Lab Sheet 4 (Module IV)
Programs using Structures and Unions
Lab Sheet 5 (Module V)
Programs using Files</td></tr>
<tr><td colspan="5">Text Book(s):
1. E. Balaguruswamy, “Programming in ANSI C”, 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316- 513-0.</td></tr>
<tr><td colspan="5">Reference Book(s):
Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020.
ReemaThareja, “Programming in C”, Oxford University Press, Second Edition, 2016.</td></tr>
</table>

Kernighan, B.W and Ritchie,D.M, “The C Programming language”, Second Edition, Pearson Education, 2015 Schildt Herbert, “C: The Complete Reference”, Tata McGraw Hill Education, 4th Edition, 2014. Stephen G. Kochan, “Programming in C”, Addison-Wesley Professional, 4th Edition, 2014.
Web Links and Video Lectures: 1. https://nptel.ac.in/courses/106/105/106105171/ 2. https://archive.nptel.ac.in/courses/106/104/106104128/

Course Code: PPS 1011	Course Title: Introduction to Verbal Ability Type of Course: Theory Only Course	L- T- P- C	0	1	0	0
Version No.	1.0					
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.					
Anti-requisites	NIL					
Course Description	This course is designed to enable students understand the importance of Verbal Ability and improve confidence, communication and professional skills to give them a competitive advantage and increase chances of success in the professional world. The course will benefit learners in presenting themselves effectively through various worksheets and learning methodologies.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Verbal Ability” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.					
Course Out Comes	On successful completion of this course the students shall be able to: CO1: Recognize significance of verbal ability CO2: Utilize the rules of communication CO3: Apply techniques of vocabulary building to showcase effective communication					
Course Content:						

Module 1	INTRODUCTION TO VERBAL ABILITY	Individual Assessment	01 Hour
Topics: Setting Expectations, Ice Breaker, Significance of verbal ability, pre-assessment			
Module 2	EFFECTIVE VERBAL COMMUNICATION	Practice Worksheets	06 Hours
Topics: Different rules of grammar and application, Subject-Verb Agreement, Tenses			
Module 3	VOCABULARY BUILDING	Practice Worksheets	04 Hours
Topics:Root words, Synonyms and antonyms, analogies, para-jumbles			
Module 4	READING COMPREHENSION	Individual Assessment	02 Hours
A session where students will be introduced to speed reading and comprehension, post-assessment			
Targeted Application & Tools that can be used: LMS			
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course			
Individual Assessment LMS MCQ			
The topics related to Skill Development: Communication, grammar rules, vocabulary building, effective presentation for skill development through participative learning techniques. This is attained through learning and practicing the rules of effective communication through worksheets as mentioned in the assessment component.			

Course Code: MAT1003	Course Title: Applied Statistics Type of Course: School Core	L T P C	1	0	2	2
Version No.	3.0					
Course Pre-requisites	None					
Anti-requisites	None					
Course Description	The goal of this course is to provide a firm understanding of probability and statistics by means of a thorough treatment of descriptive statistics, probability and probability distributions keeping in mind the future courses having statistical, quantitative and probabilistic components. The course covers topics such as descriptive statistics, probability, rules for probability, random variables and					

	probability distributions, standard discrete and continuous probability distributions.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Applied Statistics” and attain Skill Development Through Problem Solving techniques.			
Expected Outcome:	At the end of this course, students will be in a position to apply the techniques of descriptive statistics effectively interpret the ideas of probability and conditional probability demonstrate the knowledge of probability distributions Compute statistical parameters, correlation and regression, probability and sampling distributions using R software.			
Module 1	Descriptive Statistics	Assignment	Coding needed	10 classes
Introduction to Statistics, Data and statistical thinking, review of basic statistical parameters, Covariance, Correlation, Types of Measures of Correlation - Karl Pearson’s Correlation Coefficient, Spearman Rank Correlation, linear regression, Multi linear regression .				
Module 2	Probability			6 classes
Introduction to Probability, Probability of an event, Addition Principle, Multiplication law, Conditional Probability, Total Probability and Baye’s theorem with examples				
Module 3	Random Variables and Probability Distributions		Coding needed	14 classes
Introduction to Random variables, Discrete Random Variables and Continuous Random Variables, Probability Distributions, Probability Mass Function and Probability Density Function, Various Probability distributions, Binomial, Negative Binominal (Self Study), Poisson, Normal and Exponential distributions				
Module 4	Sampling Theory		Coding needed	15 classes
Introduction to Sampling Theory, Population, Statistic, Parameter, Sampling Distribution, Standard Error. Testing of Hypothesis, Types of Errors, Critical Region, level of Significance. Difference between Parametric and Non-parametric Tests, Large Sample Tests: Z-Test for Single Mean and Difference of Means (Self Study), Small Sample Tests: Student’s t-Test for Single Mean and Difference of Means, F-Test, Chi-Square Test.				
Targeted Application & Tools that can be used: The objective of the course is to familiarize students with the theoretical concepts of probability and statistics and to equip them with basic statistical tools to tackle engineering and real-life problems. Tools used: R Software / MS-Excel				
Text Book Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying E Ye, Probability and Statistics for Engineers and Scientists, Pearson Education, 2016.				

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

Course Code: ECE2007	Course Title: Digital Design Type of Course: Theory &Integrated Laboratory	L- T-P-C	2	0	2	3
Version No.	2.0					
Course Pre-requisites	[1] Elements of Electronics/Electrical Engineering, 2] Basic concepts of number representation, Boolean Algebra					
Anti-requisites	NIL					
Course Description	The purpose of this course is to enable the students to appreciate the fundamentals of digital logic circuits and Boolean algebra focusing on both combinational and sequential logic circuits. The course emphasizes on minimization techniques for making canonical and low-cost digital circuit implementations. This course deals with analysis and design of digital electronic circuits. The course also creates a foundation for future courses which includes Computer Architecture, Microprocessors, Microcontrollers, and Embedded Systems etc. The course enhances the Design, Implementation and Programming abilities through laboratory tasks. The associated laboratory provides an opportunity to verify the theoretical knowledge.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Digital Design and attain the SKILL DEVELOPMENT through EXPERIENTIAL LEARNING.					
Course Outcomes	On successful completion of this course the students shall be able to: Describe the concepts of number systems, Boolean algebra and logic gates. Apply minimization techniques to simplify Boolean expressions. Demonstrate the Combinational circuits for a given logic					

	Demonstrate the Sequential and programmable logic circuits Implement various combinational and sequential logic circuits using gates.			
Course Content:				
Module 1	Fundamentals of Number systems- Boolean algebra and digital logic	Application Assignment	Data Analysis task	06 classes
Topics: Review of Number systems and logic gates, Number base conversions, Overview of Boolean functions and simplifications, two, three, four variable K-Maps- Don't care conditions- Both SOP and POS- Universal Gates (NAND & NOR) Implementations. Introduction to HDL.				
Module 2	Boolean function simplification	Application Assignment	Data Analysis task	08 Classes
Topics: Introduction to Combinational circuits, Analysis, Design procedure, Binary Adder and Subtractor, Magnitude comparator, Parity generator and checker, Multiplexers-Demultiplexers, Decoders, Encoders and Priority Encoders, HDL Models of combinational circuits.				
Module 3	Combinational Logic circuits:	Application Assignment	Programming Task & Data Analysis task	08 Classes
Topics: Introduction to sequential circuits, Storage elements: latches and flip flops, Characteristic tables and equations, excitation table, Analysis of clocked sequential circuits, Mealy & Moore Models of finite state machines - Registers & Counters. HDL Models of Sequential circuits.				

List of Laboratory Tasks:
Experiment N0 1: Verify the Logic Gates truth table
Level 1: By using Digital Logic Trainer kit
Level 2: By using Analog devices like RPS, Volt meter, Resistors and ICs

Experiment No. 2: Verify the Boolean Function and Rules
Level 1: By using Digital Logic Trainer kit
Level 2: By using Analog devices like RPS, Volt meter, Resistors and ICs

Experiment No. 3: Design and Implementations of HA/FA
Level 1: By using basic logic gates and Trainer Kit
Level 2: By using Universal logic gates and Trainer Kit

Experiment No. 4: Design and Implementations of HS/FS
Level 1: By using basic logic gates and Trainer Kit
Level 2: By using Universal logic gates and Trainer Kit

Experiment No. 5: Design and Implementations of combinational logic circuit for specifications
Level 1: Specifications given in the form of Truth table

Level 2: Specification should be extracted from the given scenario

Experiment No. 6: Study of Flip flops

Experiment No. 7: Design and Implementations of sequential logic circuit for specifications
Level 1: Specifications given in the form of Truth table
Level 2: Specification should be extracted from the given scenario

Experiment No.8: HDL coding for basic combinational logic circuits
Level 1: Gate level Modeling
Level 2: Behavioral Modeling

Experiment No.9: HDL coding for basic sequential logic circuit
Level 1: Gate level Modeling
Level 2: Behavioral Modeling

Targeted Application & Tools that can be used:

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

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

Text Book(s):
Mano, M. Morris and Ciletti Michael D., “Digital Design”, Pearson Education, 6th edition

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

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

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

Online Resources (e-books, notes, ppts, video lectures etc.): Book Free Download (studymaterialz.in)
eBook1: Mano, M. Morris and Ciletti Michael D., “Digital Design”, Pearson Education. {[PDF] Digital Design By M. Morris Mano, Michael D Ciletti Book Free Download }
eBook2:Floyd “DIGITAL LOGIC DESIGN” fourth edition- ePub, eBook- [PDF] DIGITAL LOGIC DESIGN FOURTH EDITION FLOYD | abri.engenderhealth.org.
NPTEL Course- NPTEL :: Electrical Engineering - NOC:Digital Electronic Circuits
Digital Logic Design PPT Slide 1 (iare.ac.in)
Lab Tutorial: Multisim Tutorial for Digital Circuits - Bing video
CircuitVerse - Digital Circuit Simulator online
Learn Logisim➠ Beginners Tutorial | Easy Explanation! - Bing video
Digital Design 5: LOGISIM Tutorial & Demo
7. https://presiuniv.knimbus.com/user#/home
E-content:

Z. Xin-Li and W. Hong-Ying, "The Application of Digital Electronics in Networking Communication," 2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2016, pp. 684-687, doi: 10.1109/ICMTMA.2016.168. An encoding technique for design and optimization of combinational logic circuit DipayanBhadra;Tanvir Ahmed Tarique;Sultan Uddin Ahmed;Md. Shahjahan;KazuyukiMurase2010 13th International Conference on Computer and Information Technology (ICCIT) A. Matrosova and V. Provkin, "Applying Incompletely Specified Boolean Functions for Patch Circuit Generation," 2021 IEEE East-West Design & Test Symposium (EWDTS), 2021, pp. 1-4, doi: 10.1109/EWDTS52692.2021.9581029. A. Matrosova, V. Provkin and E. Nikolaeva, "Masking Internal Node Faults and Trojan Circuits in Logical Circuits," 2019 IEEE East-West Design & Test Symposium (EWDTS), 2019, pp. 1-4, doi: 10.1109/EWDTS.2019.8884434.
Topics relevant to “SKILL DEVELOPMENT”: Adders, Multiplexers, Decoders / Encoders; Flip-Flops, Counters and Registers for Skill Development through Experiential Learning techniques. This is attained through assessment component mentioned in course handout.

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

	3] Explain various energies, energy generating machineries and energy consumption machineries 4] Describe the fundamental concept and terminology associated with the Petroleum Industry 5] Distinguish between conventional and modern manufacturing techniques.			
Course Content:				
Module 1	Introduction to various fields in Civil Engineering	Assignment	Case studies on different Civil Engineering Projects	6 Sessions
Topics: Introduction to Civil Engineering: Definition, scope and branches of Civil Engineering, Role of Civil Engineer, Overview of Infrastructure.				
Module 2	Current Trends and Evolution in Civil Engineering	Assignment	Article Review	6 Sessions
Topics: Mechanization in Construction, Application of Digital Technologies in Planning, Design, execution, monitoring and maintenance of Construction. Overview of Smart Cities.				
Module 3	Power Production and Consumption Machinery	Assignment & Quiz	Data Collection	6 Sessions
Topics: Energy and its types, Engines and their applications, Pumps-Compressors and their applications.				
Module 4	Overview of Petroleum Engineering	Assignment & Quiz	Article Review	6 Sessions
Overview of the Petroleum Industry, Importance of Petroleum Engineering, lifecycle of Petroleum products, Classifications of E&P activities: Key difference between Offshore and Onshore, Onshore facilities, offshore platforms, Digitization of petroleum engineering				
Module 5	Industry 4.0	Assignment & Quiz	Data Collection	6 Sessions
Topics: Conventional manufacturing process: Metal forming, metal removal and metal joining process. Modern Manufacturing process: 3D Printing / Additive Manufacturing.				
Targeted Application & Tools that can be used: Application Areas include design and implementation of Smart City projects, Infrastructure maintenance, Power production, IC engines, Electric vehicles, onshore and offshore exploration and production activities				
Project work/Assignment:				
Assignment 1: Collect data and prepare report on various Mega Projects in Civil Engineering Assignment 2: Review Articles on current evolutions in Civil Engineering.				

Assignment 3: Collect data related to renewable energy generation (Wind, Solar)

Assignment 4: Prepare an energy consumption chart for a compressor or pumps.

Assignment 5: Prepare a report on role of 3D printing across various industries.

Assignment 6: Prepare an assignment on geopolitical influence on oil and gas industries.

Text Book:

T1. Elements of Civil and Mechanical Engineering, L.S. Jayagopal & R Rudramoorthy, Vikas Publishers

T2. Elements of Mechanical Engineering, by VK Manglik

T3. Fundamentals of Oil & Gas Industry for Beginners by Samir Dalvi, Notion Press; 1st edition

References

K.P. Roy, S.K. Hajra Choudhury, Nirjhar Roy, “Elements of Mechanical Engineering”, Media Promoters and Publishers Pvt Ltd, Mumbai.

Nontechnical Guide to Petroleum Geology, Exploration, Drilling & Production by Norman J. Hyne, PennWell Books; 3rd Revised edition

Web-resources:

Basic Civil Engineering

https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2706932&site=ehost-live

Post-parametric Automation in Design and Construction

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

Smart Cities : Introducing Digital Innovation to Cities

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

Innovation Energy: Trends and Perspectives or Challenges of Energy Innovation

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

Mechanical Engineering

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

Additive Manufacturing: Opportunities, Challenges, Implications

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

Society of Petroleum Engineers (SPE)

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

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

https://petrowiki.spe.org/PetroWiki

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

https://www.rigzone.com/

Topics relevant to the development of SKILLS:

Engines-Turbines and their applications.

Mechanization in Construction.

Digitization in Petroleum Industries

Course Code: 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: Demonstrate competency of Engineering Graphics as per BIS conventions and standards. Comprehend the theory of projection for drawing projections of Points, Lines and Planes under different conditions. Prepare multiview orthographic projections of Solids by visualizing them in different positions. Prepare pictorial drawings using the principles of isometric projections to visualize objects in three dimensions.			
Course Content:				
Module 1	Introduction to Drawing	Assignment	Standard technical drawing	02 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	Assignment	Projection methods Analysis	10 Sessions

<table>
<tr><td></td><td>Lines and Plane Surfaces</td><td></td><td></td><td></td></tr>
<tr><td colspan="5">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]</td></tr>
<tr><td>Module 3</td><td>Orthographic Projections of Solids</td><td>Assignment</td><td>Multi-view drawing Analysis</td><td>10 Sessions</td></tr>
<tr><td colspan="5">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]</td></tr>
<tr><td>Module 4</td><td>Isometric Projections of Solids (Using isometric scale only)</td><td>Assignment</td><td>Spatial Visualization</td><td>8 Sessions</td></tr>
<tr><td colspan="5">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]</td></tr>
<tr><td colspan="5">Text Book:
1.N. D. Bhatt, “Engineering Drawing: Plane and Solid Geometry,” Charotar Publishing House Pvt. Ltd.</td></tr>
<tr><td colspan="5">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</td></tr>
<tr><td colspan="5">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.</td></tr>
</table>

Course Code: CSE1006	Course Title: Problem Solving using JAVA Type of Course: Lab Integrated	L- T-P-C	1	0	4	3
Version No.	2.0					
Course Pre-requisites	CSE1004 – Problem-Solving Using C					
Anti-requisites	Nil					
Course Description	This course introduces the core concepts of object-oriented programming. This course has theory and lab component which emphasizes understanding the implementation and application of object-oriented programming paradigm. It helps the student to build real-time secure applications by applying these concepts and also for effective problem-solving. The students interpret and understand the need for object-oriented programming to build applications.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Problem-Solving using JAVA and attain SKILL DEVELOPMENT through EXPERIENTIAL LEARNING techniques					
Course Out Comes	On successful completion of the course, the students shall be able to: C.O. 1: Describe the basic programming concepts. [Knowledge] C.O. 2: Apply the concept of classes, objects and methods to solve problems. [Application] C.O. 3: Apply the concept of arrays and strings. [Application] C.O. 4: Implement inheritance and polymorphism in building secure applications. [Application] C.O. 5: Apply the concepts of interface and error handling mechanism. [Application]					
Course Content:						
Module 1	Basic Concepts of Programming and Java	Assignment	Data Collection/Interpretation	12 Sessions		
Topics: Introduction to Principles of Programming: Process of Problem Solving, Java program structure, Download Eclipse IDE to run Java programs, Sample program, Data types, Identifiers, Variables, Constants in java, Operators, Assignments and Expression, Basic Input/ Output functions, Control Statements: Branching and Looping.						
Module 2	Classes, objects, methods and Constructors	Case studies / Case let	Case studies / Case let	12 Sessions		
Topics: Classes, Objects and Methods: Introduction to object Oriented Principles, defining a class, adding data members and methods to the class, access specifiers, instantiating objects, reference variable, accessing class members and methods. Static Polymorphism: Method overloading, constructors, constructor overloading, this keyword, static keyword, Nested classes, Accessing members in nested classes.						
Module 3	Arrays, String and String buffer	Quiz	Case studies / Case let	14 Sessions		

Topics: Arrays: Defining an Array, Initializing & Accessing Array, Multi–Dimensional Array, Array of objects. String: Creation & Operation. String builder class, methods in String Buffer.				
Module 4	Inheritance and Polymorphism	Quiz	Case studies / Case let	14 Sessions
Topics: Inheritance: Defining a subclass, Types of Inheritance, super keyword. Dynamic Polymorphism: Method overriding. Final keyword: with data members, with member functions and with class. Abstract keyword: with data members, with member functions and with class, Exception handling.				
Module 5	Input & Output Operation in Java	Quiz	Case studies / Case let	14 Sessions
Input/output Operation in Java(java.io Package), Streams and the new I/O Capabilities, Understanding Streams, working with File Objects, File I/O Basics, Reading and Writing to Files, Buffer and Buffer Management, Read/Write Operations with File Channel, Serializing Objects, Observer and Observable Interfaces.				
List of Laboratory Tasks: P1 - Problem Solving using Basic Concepts. P2 - Problem Solving using Basic Concepts and Command Line Arguments. P3 - Programming assignment with class, objects, methods and Constructors. P4 - Programming assignment with method overloading. P5 - Programming assignment with constructor overloading. P6 - Programming assignment with Static members and static methods. P7 - Programming assignment with Nested classes. P8 - Programming assignment using Arrays. P9 - Programming assignment using Strings. P10 - Programming assignment using String Builder. P11 - Programming assignment using Inheritance and super keyword. P12 - Programming assignment using Method overriding and Dynamic method invocation. P13 - Programming assignment using Final keywords. P14 - Programming assignment using Abstract keywords. P15 - Programming assignment using Interface. P16 - Programming assignment using Interface. P17 - Programming assignment CharacterStream Classes P18 - Programming assignment Read/Write Operations with File Channel				
Targeted Application & Tools that can be used : JDK /eclipse IDE/ net Beans IDE.				
Text Book T1 Herbert Schildt, “The Complete Reference Java 2”, Tata McGraw Hill Education.				
References R1: Cay S Horstmann and Cary Gornell, “CORE JAVA volume I-Fundamentals”, Pearson R2: James W. Cooper, “Java TM Design Patterns – A Tutorial”, Addison-Wesley Publishers. E book link R1: http://rmi.yaht.net/bookz/core.java/9780134177373-Vol-1.pdf E book link R2: Java(tm) Design Patterns: A Tutorial([PDF] [7qmsenjl97t0] (vdoc.pub) Web resources https://youtube.com/playlist?list=PLu0W_9lII9agS67Uits0UnJyrYiXhDS6q				

https://puniversity.informaticsglobal.com:2229/login.aspx
Topics relevant to the development of “Skill Development”: Static Polymorphism Method overloading, constructors constructor overloading this keyword static keyword and Inner classes Inheritance and Polymorphism. for Skill Development through Experiential Learning techniques. This is attained through the assessment component mentioned in the course handout.

ENG2001	Advanced English		L- T- P- C	1	0 | 2	2
Version No.	1.3					
Course Pre-requisites	ENG1002 Technical English					
Anti-requisites	NIL					
Course Description	The course emphasizes on technical communication at advanced level by exploring critical reading, technical presentation and review writing. The purpose of the course is to enable learners to review literature in any form or any technical article and deliver technical presentations. Extensive activities in practical sessions equip to express themselves in various forms of technical communications. Technical presentations and the module on career setting focus on learners’ area of interests and enhance their English language writing skills to communicate effectively.					
Course Out Come	On successful completion of the course the students shall be able to: Develop a critical and informed response reflectively, analytically, discursively, and creatively to their reading. Communicate effectively, creatively, accurately and appropriately in their writing. Deliver technical presentations Design resume and create professional portfolio to find a suitable career					
Course Content: Theory						
Module 1	Critical Reasoning and Writing	Writing Essays	Critical Reading	4 Classes		
Topics: A Catalog of Reading Strategies The Myth of Multitasking A Guide to Writing Essays Speculating about Causes or Effects Is Google Making Us Stupid (Self Study)						
Module 2	Technical Presentation	Presentation	Oral Skills	3 Classes		
Topics:						

Planning the presentation Creating the presentation Giving the presentation				
Module 3	Writing Reviews	Prezi	Review Writing	4 Classes
Topics: Review Writing Short film reviews Advanced English Grammar (Self Study)				
Module 4	Starting your Career	Online Writing Lab	Writing Skills	4 Classes
Topics: Preparing a Resume Writing Effective Application Letter Creating a Professional Portfolio				

Course Content: Practical Sessions		
Module 1	Critical Reasoning and Writing	8 Classes
Reading and Analyzing Level 1 – Annotation Level 2 - Assumptions Writing Narrative Essays Level 1 – Draft 1 Level 2 – Draft 2		
Module 2	Technical Presentation	10 Classes
Fishbowl In Fishbowl, students form concentric circles with a small group inside and a larger group outside. Students in the inner circle engage in an in-depth discussion, while students in the outer circle listen and critique content, logic, and group interaction. Level 1 – within group Level 2 – Among 2 group Technical Group Presentation		
Module 3	Writing Reviews	Classes
Practice Worksheets Level 1 – Eliminating the Passive Voice Level 2 – Simple, compound and complex sentences Writing Short Film Reviews		
Module 4	Starting your Career	Classes
Collaborative Project Job search and writing report Writing Resume		
Module 1-4	Academic Journal	2 Classes

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

Course Code: ECE2010	Course Title: Innovative Projects using Arduino	L- T-P- C	-	-	-	1
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course is designed to provide an in-depth understanding of Arduino microcontrollers and their application in various real time projects involving sensors. Throughout the course, students will learn the fundamentals of Arduino programming and gain hands-on experience with a wide range of sensors. Students will explore how to connect and					

	interface sensors with Arduino boards, read sensor data, and use it to control various output devices This course is suitable for beginners who are interested in exploring the world of electronics and developing practical applications using Arduino and sensors.			
Course Objective	The objective of the course is Employability Skills of student by using PARTICIPATIVE LEARNING techniques.			
Course Outcomes	On successful completion of the course the students shall be able to Explain the main features of the Arduino prototype board Demonstrate the hardware interfacing of the peripherals to Arduino system. Understand the types of sensors and its functions Demonstrate the functioning of live projects carried out using Arduino system.			
Course Content:				
Module 1	Basic concepts of Arduino	Hands-on	Interfacing Task and Analysis	4 Sessions
Topics: Introduction to Arduino, Pin configuration and architecture, Device and platform features, Concept of digital and analog ports, Familiarizing with Arduino Interfacing Board, API's , Introduction to Embedded C and Arduino platform, Arduino Datatypes and variables, Arduino i/o Functions, Arduino Communications, Arduino IDE, Various Cloud Platforms.				
Module 2	Sensory Devices	Hands-on	Interfacing Task and Analysis	4 Sessions
Arduino Sensors: Humidity Sensor, Temperature Sensor, Water Detector / Sensor, PIR Sensor, Ultrasonic Sensor, Connecting Switches and actuators, sensor interface with Arduino. Introduction to 3D Printer: 3D Printer technology and its working Principles, Applications. Introduction to online Simulators: Working with Tinkercad Simulator.				
Topics: Types of Arduino boards, sensors, 3D Printer				
Targeted Application & Tools that can be used: Application Area: Home Automation, Environmental Monitoring, Agriculture and Farming, Industrial Automation, Internet of Things (IoT), Robotics, Wearable Devices, Security Systems, Education and Learning. These are just a few examples of the many application areas where Arduino and sensors can be applied. The flexibility and affordability of Arduino, combined with the wide range of sensors available, allow for endless possibilities in creating innovative projects. Professionally Used Software: students can use open SOURCE Softwares Arduino IDE and Tincker CAD				
Project work/Assignment:				

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

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

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

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

References
Reference Book(s)
1.   Neerparaj Rai “Arduino Projects for Engineers” BPB publishers,first edition, 2016.
2. Ryan Turner ”Arduino Programming ” Nelly B.L. International Consulting Ltd. first edition,2019.

Online Resources (e-books, notes, ppts, video lectures etc.):
Arduino trending Projects < https://www. https://projecthub.arduino.cc/>
Introduction to Arduino < https://onlinecourses.swayam2.ac.in/aic20_sp04/preview>
Case studies on Wearable technology< https://www.hticiitm.org/wearables>

E-content:
Cattle Health Monitoring System Using Arduino and IOT (April 2021| IJIRT | Volume 7 Issue 11 | ISSN: 2349-6002)
M H Hemanth Kumar, Ravi Pratap Singh, Nishu Sharma, Pragya Singh” IOT BASED SMART SECURITY SYSTEM USING ARDUINO” 2021 JETIR August 2021, Volume 8, Issue 8.
R. Maheswar, P. Jayarajan, S. Vimalraj, G. Sivagnanam, V. Sivasankaran and I. S. Amiri, "Energy Efficient Real Time Environmental Monitoring System Using Buffer Management Protocol," 2018, pp. 1-5, doi: 10.1109/ICCCNT.2018.8494144. https://ieeexplore.ieee.org/document/8494144.
Yaser S Shaheen,Hussam., " Arduino Mega Based Smart Traffic Control System ," December 2021 Asian Journal of Advanced Research and Reports 15(12): 43-52, 2021(15(12): 43-52, 2021):15(12): 43-52, 2021.

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

Course Code: PPS 1002	Course Title: Soft Skills for Engineers Type of Course: Practical Only Course		L- T-P- C	0-0-2-1
Version No.	1.0			
Course Pre-requisites	Students are expected to understand Basic English. Students should have desire and enthusiasm to involve, participate and learn.			
Anti-requisites	NIL			
Course Description	This course is designed to enable students understand soft skills concepts and improve confidence, communication and professional skills to give the students a competitive advantage and increase chances of success in the professional world. The course will benefit learners in presenting themselves effectively through various activities and learning methodologies.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of “Soft Skills” and attain SKILL DEVELOPMENT through PARTICIPATIVE LEARNING techniques.			
Course Out Comes	On successful completion of this course the students shall be able to: CO1: Recognize significance of soft skills CO2: Illustrate effective communication while introducing oneself and others CO3: List techniques of forming healthy habits CO4: Apply SMART technique to achieve goals and increase productivity			
Course Content:				
Module 1	INTRODUCTION TO SOFT SKILLS		Classroom activity	04 Hours
Topics: Setting Expectations, Ice Breaker, Significance of soft skills, Formal grooming, punctuality				
Module 2	EFFECTIVE COMMUNICATION		Individual Assessment	10 Hours
Topics: Different styles of communication, Difference between hearing and listening, Effective communication for success, Email etiquette, Self-introduction framework, Video introduction, email- writing, Resume Building- Digital, Video, Traditional.				
Module 3	HABIT FORMATION		Worksheets & Assignment	4 Hours
Topics: Professional and personal ethics for success, Identity based habits, Domino effect, Habit Loop, Unlearning, standing up for what is right				
Module 4	Goal setting & Time Management		Goal sheet	8 Hours
A session where students will be introduced to Time management, setting SMART Goals, Introduction to OKR Techniques, Time Management Matrix, steps to managing time through outbound group activity, making a schedule, Daily Plan and calendars (To Do List), Monitoring/charting daily activity				

Targeted Application & Tools that can be used: LMS
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course
Individual Assessment LMS MCQ
The topics related to Skill Development: Communication and professional grooming, Goal setting and presentation for skill development through participative learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: MAT1002	Course Title: Transform Techniques, Partial Differential Equations and Their Applications Type of Course: School Core	L-T-P- C	3	0	0	3
Version No.	2.0					
Course Pre-requisites	MAT1001 - Linear Algebra and Calculus					
Anti-requisites	NIL					
Course Description	This course aims to introduce various transform techniques such as Laplace transform, Fourier transform and Z transform in addition to expressing functions in terms of Fourier series. The course covers applications of Laplace transform to LCR circuits and solution of difference equations using z-transform. The course also deals with the analytical methods for solving partial differential equations and the classical applications of partial differential equations.					
Course Objective	The objective of the course is Skill Development of student by using Problem Solving Techniques.					
Course Outcomes	On successful completion of this course the students shall be able to: CO-1: Express functions in terms of uniformly convergent Fourier series. CO-2: Apply Laplace transform technique to solve differential equations. CO-3: Employ z-transform technique to solve difference equations. CO-4: Solve a variety of partial differential equations analytically.					
Course Content:						
Module 1	Fourier Series			10 CLASSES		
Fourier series: Fourier series - Euler's formulae - Dirichlet's conditions - Change of Interval - half range series – RMS value – Parseval's identity – Computation of harmonics. Engineering Applications of Fourier series.						
Module 2	Integral Transforms			15 Classes		
Laplace Transform: Definition and Laplace transforms of elementary functions. Properties of Laplace transform. Laplace transform of periodic function, unit-step function and impulse function and the related problems. Inverse Laplace transform of standard functions and problems, initial and final value theorems. Convolution theorem, solution of linear ordinary differential equations, LCR circuit problems.						

Fourier Transform: Integral transforms, infinite Fourier transforms, Fourier sine and cosine transforms, inverse Fourier transforms. Engineering Applications of Fourier transform.				
Module 3	Z Transform and Difference Equations			8 Classes
Definition of Z-transform, Z transforms of standard functions and the related problems, standard inverse Z transforms and problems, computation of inverse Z-transform by partial fraction and convolution methods, solution of difference equations using Z-transforms. Business and Engineering Applications of Z transform.				
Module 4	Partial Differential Equations			12 Classes
Partial Differential Equations: Formation of PDEs, solution of non-homogeneous PDEs by direct integration, solution of homogeneous PDEs involving derivatives with respect to only one independent variable, method of separation of variables, solution of the Lagrange's PDE of the type Pp + Qq = R. Applications of PDEs: Various possible solutions of the one dimensional wave and heat equations by the method of separation of variables, D'Alembert's solution of the wave equation, solution of related boundary value problems.				
Targeted Applications & Tools that can be used: Applications to electrical engineering, vibrational analysis, acoustics, optics, signal processing, image processing, quantum mechanics, econometrics and shell theory by means of Fourier Series and integral transforms. Opens up new approaches in terms of Z-transform to solving one of the central problems of modern science involving difference equations. Finding the solutions of boundary value problems involving PDEs with reference to wave, heat, and Laplace equations.				
Assignment: Mention the Type of Project /Assignment proposed for this course				
Two Assignments based on the applications of the concepts leading to a minimum of 5 engineering problems from a common pool of problems.				
Text Book Erwin Kreyszig, 2017: " Advanced Engineering Mathematics", 10th Edition, John Wiley.				
References: B. S. Grewal, 2017: "Higher Engineering Mathematics" 45th Edition, Khanna Publishers. Peter V O'Neil, 2015: "Advanced Engineering Mathematics", 7th Edition, Cengage Learning. Glyn James, 2016: "Advanced Modern Engineering Mathematics", 4th Edition, Pearson Education. Michael D. Greenberg, 2018: "Advanced Engineering Mathematics", 2nd Edition, Pearson Education.				
Topics relevant to the development of Foundation Skills: All the solution methods. Topics relevant to development of Employability skills: Use of relevant scientific application packages.				

<table>
<tr><td>Course Code:
CSE2001</td><td colspan="2">Course Title: Data Structures and Algorithms
Type of Course: Integrated</td><td>L- T-P- C</td><td>3-0-2-4</td></tr>
<tr><td>Version No.</td><td colspan="4">1.0</td></tr>
<tr><td>Course Pre-requisites</td><td colspan="4">Problem Solving Using Java</td></tr>
<tr><td>Anti-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Course Description</td><td colspan="4">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.</td></tr>
<tr><td>Course Objective</td><td colspan="4">The objective of the course is to familiarize the learners with the concepts of Data Structures and Algorithms and attain Skill Development through Experiential Learning techniques.</td></tr>
<tr><td>Course Out C omes</td><td colspan="4">On successful completion of the course the students shall be able to:
CO1: Implement program for given problems using fundamentals of data structures. [Application]
CO2: Apply an appropriate linear data structure for a given scenarios. [Application]
CO3: Apply an appropriate non-linear data structure for a given scenarios. [Application]
CO4: Explain the performance analysis of given searching and sorting algorithms.</td></tr>
<tr><td>Course Content:</td><td colspan="4"></td></tr>
<tr><td>Module 1</td><td>Introduction to Data Structure and Linear Data Structure – Stacks and Queues</td><td>Assignment</td><td>Program activity</td><td>18 Sessions</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 2</td><td>Linear Data Structure- Linked List</td><td>Assignment</td><td>Program activity</td><td>17 Sessions</td></tr>
</table>

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, Programming examples.				
Module 3	Non-linear Data Structures - Trees and Graph	Assignment	Program activity	15 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. Graph - Basic Concept of Graph Theory and its Properties, Representation of Graphs.				
Module 4	Searching & Sorting Performance Analysis	Assignment	Program activity	14sessions
Topic: Sorting & Searching - Sequential and Binary Search, Sorting – Selection and Insertion sort. Performance Analysis - Time and space analysis of algorithms – Average, best and worst case analysis.				
List of Laboratory Tasks: Lab sheet -1 Level 1: Prompt the user, read input and print messages. Programs using class, methods and objects Level 2: Programming Exercises on fundamental Data structure - Arrays based on Scenario. Lab sheet -2 Level 1: Programming Exercises on Stack and its operations Level 2: Programming Exercises on Stack and its operations with condition Lab sheet -3 Level 1: Programming on Stack application infix to postfix Conversion Level 2: - Lab sheet -4 Level 1: Programming Exercises on Queues and its operations with conditions Level 2: - Lab sheet -5 Level 1: Programming Exercises on Linked list and its operations. Level 2: Programming Exercises on Linked list and its operations with various positions Lab sheet -6 Level 1: - Level 2: Programming scenario based application using Linked List Lab sheet -7 Level 1: Programming Exercises on factorial of a number Level 2: Programming the tower of Hanoi using recursion Lab sheet -8 Level 1: - Level 2: Programming the tower of Hanoi using recursion Lab sheet -9 Level 1: Programming Exercise on Doubly linked list and its operations Level 2: - Lab sheet -10 Level 1: Program to Construct Binary Search Tree and Graph				

Level 2: Program to traverse the Binary Search Tree in three ways(in-order, pre-order and post-order) and implement BFS and DFS
Lab sheet -11
Level 1: Program to Implement the Linear Search & Binary Search
Level 2: Program to Estimate the Time complexity of Linear Search
Lab sheet -12
Level 1: Program to Implement and Estimate the Time complexity of Insertion Sort
Level 2: Program to Implement and Estimate the Time complexity of Insertion Sort
Lab sheet -13
Level 1: Program to Implement and Estimate the Time complexity of Selection Sort
Level 2: Program to Implement and Estimate the Time complexity of Selection Sort

Targeted Application & Tools that can be used
Use of PowerPoint software for lecture slides and use of Ubuntu for lab programs to execute.
Tool is Codetantra tool.

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 Narasimha Karumanchi: “Data Structures and Algorithms Made Easy in Java”, 5th Edition, CareerMonk Publications, 2017.

References
R1 Mark Allen Weiss: “Data Structures and Algorithm Analysis in Java”, 4th Edition, Pearson Educational Limited, 2014.
R2 Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser: “Data Structures and Algorithms in Java”, 6th Edition, John Wiley & Sons, Inc., ISBN: 978-1-118-77133-4, 2014.
R3 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, 2017: “Introduction to Algorithms”, 3rd Edition, PHI Learning Private Limited.
Web resources:
For theory: https://onlinecourses.nptel.ac.in/noc20_cs85/preview
For Lab : codetantra tool
https://puniversity.informaticsglobal.com/login

Topics relevant to “SKILL DEVELOPMENT”: Linked list and its type, Tree traversal and hashing tables for Skill Development through Experiential Learning techniques. This is attained through the assessment component mentioned in the course handout.

Course Code: CSE3155	Course Title: Data Communications and Computer Networks Type of Course: Program Core Theory–Laboratory integrated	L-T-P-C 3-0-2-4	3	0	2	4

Version No.	1.0
Course Pre-requisites	Digital Design
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 Out Comes	On successful completion of the course, the students shall be able to: 1] I llustrate the Basic Concepts Of Data Communication and Computer Networks. 2] Analyze the functionalities of the Data Link Layer. 3] Apply the Knowledge of IP Addressing and Routing Mechanisms in Computer Networks. 4] Demonstrate the working principles of the Transport layer and Application Layer.
Course Content:	

Module 1	Introduction and Physical Layer-CO1	Assignment	Problem Solving	07 Classes
Introduction to Computer Networks and Data communications, Network Components – Topologies, Transmission Media –Reference Models -OSI Model – TCP/IP Suite. Physical Layer -Analog and Digital Signals – Digital and Analog Signals – Transmission - Multiplexing and Spread Spectrum.				
Module 2	Reference Models and Data Link Layer – CO2	Assignment	Problem Solving	7 Classes

<table>
<tr><td colspan="5">Data Link Layer - Error Detection and Correction – Parity, LRC, CRC, Hamming Code, Flow Control and Error Control, Stop and Wait, ARQ, Sliding Window, Multiple Access Protocols, CSMA/CD,CSMA/CA, IEEE 802.3, IEEE 802.11 Ethernet.</td></tr>
<tr><td>Module 3</td><td>Network Layer – CO 3</td><td>Assignment</td><td>Problem Solving</td><td>10 Classes</td></tr>
<tr><td colspan="5">Network Layer Services - Network Layer Services, Switching Techniques, IP Addressing methods- IPv4 IPV6 – Subnetting. Routing, - Distance Vector Routing – RIP-BGP-Link State Routing –OSPF-Multi cast Routing-MOSPF- DVMRP – Broad Cast Routing. EVPN-VXLAN, VPLS, ELAN.</td></tr>
<tr><td>Module 4</td><td>Transport and Application Layer -CO3</td><td>Assignment</td><td>Problem Solving</td><td>10 Classes</td></tr>
<tr><td colspan="5">Transport Layers - Connection management – Flow control – Retransmission, UDP, TCP, congestion control, – Congestion avoidance (DECbit, RED)
The Application Layer: Domain Name System (DNS), Domain Name Space, SSH, FTP, Electronic Mail (SMTP, POP3, IMAP, MIME) – HTTP – – SNMP, Web Services, Virtual Networking.</td></tr>
<tr><td colspan="5">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.</td></tr>
</table>

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

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

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

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

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

Problem Solving: Choose and appropriate devices and implement various network concepts.
Programming: Simulation of any network using NS2.

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

References

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

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

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

E-Resources:

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

Course Code: CSE2009	Course Title: Computer Organization and Architecture	L-T- P-C	3-0-0-3
Version No.	2.0		
Course Pre-requisites	CSE 2015 Digital Design		
Anti-requisites	NIL		
Course Description	This course introduces the core principles of computer architecture and organization from basic to intermediate level. This theory based course emphasizes on understanding the interaction between computer hardware and software. It equips the students with the intuition behind assembly-level		

	instruction set architectures. It helps the students to interpret the operational concepts of computer technology as well as performance enhancement.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of Computer Organization and Architecture and attain Skill Development through Participative Learning techniques.			
Course Outcomes	On successful completion of the course the students shall be able to: 1] Describe the basic components of a computer, their interconnections, and instruction set architecture [Comprehension] 2] Apply appropriate techniques to carry out selected arithmetic operations 3] Explain the organization of memory and processor sub-system			
Course Content:				
Module 1	Basic Structure of computers	Assignment	Data Analysis task	12 Classes
Topics: Computer Types, Functional Units, Basic Operational concepts, Bus Structures, Computer systems RISC & CISC, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Arithmetic Operations on Signed numbers. Instructions and Instruction Sequencing, Instruction formats, Memory Instructions.				
Module 2	Instruction Set Architecture and Memory Unit	Assignment	Analysis, Data Collection	12 Classes
Topics: Instruction Set Architecture: Addressing Modes, Stacks and Subroutines. Memory System: Memory Location and Addresses, Memory Operations, Semiconductor RAM Memories, Internal Organization of Memory chips, Cache memory mapping Techniques.				
Module 3	Arithmetic and Input/output Design	Case Study	Data analysis task	10 Classes
Topics: Arithmetic: Carry lookahead Adder, Signed-Operand Multiplication, Integer Division, and Floating point operations. Input/output Design: Accessing I/O Devices, I/O communication, Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits				
Module 4	BPU and Pipelining	Assignment	Analysis, Data Collection	11 Classes
Topics: Basic Processing Unit: Fundamental Concepts, Single Bus organization, Control sequence, Execution of a Complete Instruction, Multiple Bus Organization.				

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

Targeted Application & Tools that can be used:

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

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

Text Book
Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Computer Organization”, Fifth Edition, McGraw-Hill Higher Education, 2016 reprint.

References

William Stallings, “Computer Organization & Architecture – Designing for Performance”, 11th
Edition, Pearson Education Inc., 2019

David A. Patterson & John L. Hennessy, “Computer Organization and Design MIPS Edition- The Hardware/Software Interface”, 6th Edition, Morgan Kaufmann, Elsevier Publications, November 2020.
Web References:
NPTEL Course on “Computer architecture and organization” IIT Kharagpur By Prof. Indranil Sengupta, Prof. Kamalika Datta. https://nptel.ac.in/courses/106105163
NPTEL Course on “Computer Organization”, IIT Madras By Prof. S. Raman. https://nptel.ac.in/courses/106106092
https://puniversity.informaticsglobal.com:2229/login.aspx

Topics relevant to “SKILL DEVELOPMENT”: Generation of Computers, CISC and RISC processors, Bus Arbitration, Collaboration and Data collection for Term assignments and Case Studies for Skill Development through Participative Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: MAT2004	Course Title: Discrete Mathematical Structures Type of Course: Program Core	L-T-P- C	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Nil					
Anti-requisites	Nil					

Course Description	The course provides insights into the fundamental aspects of mathematical logic and predicate calculus. The course delves deeply into the concepts of algebraic structures, lattices and Boolean algebras which are widely used in computer science and engineering. It also highlights the principles of counting techniques and their applications.			
Course Objective	The objective of the course is Skill Development of student by using Problem Solving Techniques.			
Course Outcomes	On successful completion of the course the students shall be able to: CO1: Explain logical sentences through predicates, quantifiers and logical connectives. CO2: Comprehend the basic principles of set theory and different types of relations. CO3: Elucidate the concepts of lattices and Boolean algebra. CO4: Deploy the counting techniques to tackle combinatorial problems.			
Course Content:				
Module 1	Mathematical Logic and Predicate Calculus			12 classes
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.				
Module 2	Algebraic Structures			10 classes
Sets and set-operations, functions, relations and their properties & representations of relation by matrix, closure of different type of relations, equivalence relations, primitive recursive function.				
Module 3	Lattices and Boolean Algebra			11 classes
Partial ordering, Posset, Lattices & Algebraic structures, Sub lattice, Basic properties of algebraic systems by lattices, Distributive lattices, complement of an element in a lattice, Boolean lattice & Boolean algebra, cancellation laws and unique complement theorem.				
Module 4	Principles of Counting Techniques			12 classes
Chinese Remainder Theorem, pigeonhole principle, generalized pigeonhole principle, Generalized Permutations and Combinations, Recurrence Relations.				
Targeted Application & Tools that can be used: Discrete mathematics provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems.				

Project work/Assignment: Mention the Type of Project /Assignment proposed for this course
Assignment 1: Logic Equivalences and Predicate calculus. Assignment 2: Equivalence Relations and Lattices Assignment 3: Recurrence Relations
Text Books Kenneth H. Rosen, “Discrete Mathematics and its Applications”, McGraw-Hill’s 7th Edition, 2011. Kolman, Bernard; Busby, Robert C; Ross, Sharon Cutler,” Discrete mathematical structures”, Pearson India, 6th Edition, 2015. Liu, C L Mohapatra, D P.,” Elements of Discrete Mathematics a Computer oriented approach”, New Delhi McGraw Hill Education, 4th Edition, 2015. Mott, Joe L; Kandel, Abraham; Baker, Theodore P, ”Discrete Mathematics for Computer Scientists and Mathematicians”, Pearson India, 2nd Edition, 2015. Epp, Susanna S, “Discrete Mathematics with applications”, New Delhi Cengage Learing, 4th Edition, 2016.
References: Tremblay, J.P. and Manohar.R, “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw Hill Pub. Co. Ltd, New Delhi, 30th Reprint, 2011. Grimaldi, R.P. “Discrete and Combinatorial Mathematics: An Applied Introduction”, 4th Edition, Pearson Education Asia, Delhi, 2007. Discrete Mathematics, Richard Johnsonbaugh, 8th Edition, Prentice Hall, 2017.

Course Code: CSE3190	Course Title: Fundamentals of Data Analytics Type of Course: Theory-embedded Lab	L-T- P-C	2	0	2	3
Version No.	3.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	Fundamentals of Data Analytics is designed for inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, and supports in decision-making. The course begins by covering Data extraction, pre-processing, and transformation. It delivers					

	the basic statistics and taught in an intuitive way to analysis the data. This course will help the students to apply the knowledge on data analysis to a wide range of applications.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of Fundamentals of Data Analytics and attain SKILL DEVELOPMENT through PROBLEM SOLVING Methodologies.			
Course Out Comes	On successful completion of the course the students shall be able to: Explain different types of data and variables. Interpret data using appropriate statistical methods. Demonstrate the collection, processing and analysis of data for any given application and Illustrate various charts using visualization methods. Apply the Data Analysis techniques by R Programming			
Course Content:				
Module 1	Introduction to Data Analysis	Assignment	Data Collection, data analysis, Programming	8 Sessions
Topics: Introducing Data, overview of data analysis: Data in the Real World, Data vs. Information, The Many “Vs” of Data, Structured Data and Unstructured Data, Types of Data, Data Analysis Defined, Types of Variables, Central Tendency of Data, Scales of Data, Sources of Data. Data preparation. R Studio: Base R-R Studio IDE-Introduction to R Projects and R Markdown. Basic R: R as a Calculator-Scripts and Comments-R Variables. Data I/O: Working Directories-Importing Data Exporting Data-More ways to save-Data I/O in Base R.				
Module 2	Data Analysis and Visualization	Case studies	Programming	8 Sessions
Topics: Data Summarization: One Quantitative and Categorical Variable. Data Classes: One Dimensional Data Classes-Data Frames and Matrices-Lists. Data Cleaning: Dealing with Missing Data-Strings and Recoding Variables. Manipulating Data in R: Reshaping Data-Merging Datasets. Data Visualizations: Plotting with ggplot2- Plotting with Base R				
Module 3	Statistical Analysis	Case studies	R programming	7 Sessions
Topics: Proportion tests-Chi squared test-Fisher exact test-Correlation-T test-Wilcoxon Rank sum tests-Wilcoxon signed rank test- one-way ANOVA test- Kruskal Wallis test				
Module 4	Predictive Analysis	Case studies	Programming	8 Sessions
Topics: Linear least-squares – implementation – the goodness of fit – testing a linear model – weighted resampling. Regression using Stats models – multiple regression – nonlinear relationships – logistic regression – estimating parameters – accuracy. Time series analysis – moving averages – missing values – serial correlation – autocorrelation. Introduction to survival analysis				
List of Laboratory Tasks: Experiment No. 1: Introduction to R and RStudio Level 1: Getting Started with R and RStudio Installing R and RStudio. Basic R syntax and commands. Level 2: Working with RStudio				

Understanding the RStudio interface.
Creating and managing R scripts.
Experiment No. 2: Basic Data Handling in R
Level 1: Data Types and Structures in R
Vectors, matrices, and data frames.
Lists and factors.
Level 2: Data Import and Export
Reading data from CSV, Excel, and text files.
Exporting data to different formats.
Level 3: Exploring Datasets
Using functions like head(), summary(), and str().

Experiment No. 3: Basic Data structure in R
Level 1: a. Demonstrate a program to join columns and rows in a data frame using cbind() and rbind() in R.
b.Implement different data structures in R (Vectors, Lists, Data Frames)
Level 2: R AS CALCULATOR APPLICATION a. Using with and without R objects on console
Using mathematical functions on console
Write an R script, to create R objects for the calculator application
Experiment No. 4: Data Cleaning and Preprocessing
Level 1: Handling Missing Data in R
Identifying missing values.
Imputing missing values using mean, median, or other methods.
Level 2: Data Transformation in R
Standardizing and normalizing data.
Log-transformations and scaling.
Experiment No. 5: Exploratory Data Analysis (EDA) with R
Level 1: Descriptive Statistics
Calculating mean, median, and standard deviation.
Visualizing data using histograms, box plots, and scatter plots.
Experiment No. 6: Data Visualization with ggplot2
Level 1: Demonstrate various graphs that can be made and altered using the ggplot2 package.
Level 2: Create 500 random temperature readings for six cities over a season and then plot the generated data using ggplot2 packages in R
Experiment No. 7: Perform Tests of Hypotheses hypothesis test (parametric)
Level 1: How to perform tests of hypotheses about the mean when the variance is known. How to compute the p-value. Explore the connection between the critical region, the test statistic, and the p-value.
Level 2: A teacher claims that people who work for only five hours per week will score significantly lower than people who work for ten hours per week on a quantitative abilities test. He brings twenty people and randomly assigned them to one or two groups. In one group he has participants who work for ten hours and in another group, he has participants who work for five hours. He conducts the test for all participants. Scores on the test range from one to ten with higher scores representing better performance. Test if there is any significant difference between those who work for five hours per week versus those who work for ten hours per week based on the test performance.
Experiment No 8: Hypothesis – Non-Parametric Test

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

Experiment No 9: Correlation and Covariance

Level 1: Using the iris data set in R

Find the correlation matrix.

Plot the correlation plot on dataset and visualize giving an overview of relationships among data on iris data.

Analysis of covariance: variance (ANOVA), if data have categorical variables on iris data.

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

Experiment No 11: Regression Model

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

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

Experiment No. 12: Time Series Analysis in R

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

Targeted Application & Tools that can be used:

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

Text Books

Glenn J. Myatt and Wayne P. Johnson, “Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining Paperback”, Import, 22 July 2014.

Introduction to statistics and Data analytics, Christian H, Michael S, Springer,2016

Introduction to R- Robert Parker, John Mushcelli and Andrew Jaffe, Johns Hopkins University, 2020 (E-resource)

Introduction to Time Series and Forecasting (Springer Texts in Statistics), Peter Brockwell, Richard A. Davis, Springer, 2016.

References

Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining Paperback, Glenn J. Myatt and Wayne P. Johnson, Import, 22 July 2014.

The R Software-Fundamentals of Programming and Statistical Analysis -Pierre Lafaye de Micheaux, Remy Drouilhet, Benoit Liquet, Springer 2013.

Online resources:

http://www.modernstatisticswithr.com/solutions.html#solutionsch3

https://johnmuschelli.com/intro_to_r/

https://users.phhp.ufl.edu/rlp176/Courses/PHC6089/R_notes/

Topics relevant to development of “FOUNDATION SKILLS”:

Statistical Concepts for data, visualization techniques.

Data collection for project based assignments.

Inferential Statistics (T test, Z test)

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

Course Code: CSE2014	Course Title: Software Engineering Type of Course: School Core [Theory Only]	L-T-P- C	3-0-0-3	
Version No.	1.0			
Course Pre-requisites	NIL			
Anti-requisites	NIL			
Course Description	The objective of this course is to provide the fundamentals concepts of Software Engineering process and principles. The course covers software requirement engineering processes, system analysis, design, implementation and testing aspects of software system development. The course covers software quality, configuration management and maintenance.			
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Software Engineering and attain Skill Development through Participative Learning techniques.			
Course Out Comes	On successful completion of this course the students shall be able to: 1] Describe the Software Engineering principles, ethics and process models(Knowledge) 2] Identify the requirements, analysis and appropriate design models for a given application(Comprehension) 3] Understand the Agile Principles(Knowledge) 4] Apply an appropriate planning, scheduling, evaluation and maintenance principles involved in software(Application)			
Module 1	Introduction to Software Engineering and Process Models (Knowledge level)	Quiz		09 Hours
Introduction: Need for Software Engineering, Professional Software Development, Software Engineering Ethics, Software Engineering Practice-Essence of Practice, General Principles Software Development Life Cycle Models: Waterfall Model – Classical Waterfall Model, Iterative Waterfall Model, Evolutionary model-Spiral, Prototype.				
Module 2	Software Requirements, Analysis and Design (Comprehension level)	Assignment	Development of SRS documents for a given scenario	11 Hours
Requirements Engineering: Eliciting requirements, Functional and non- Functional requirements, Software Requirements Specification (SRS), Requirement Analysis and validation. Requirements modelling- Introduction to Use Cases, Activity diagram and Swim				

lane diagram. CASE support in Software Life Cycle, Characteristics of CASE Tools, Architecture of a CASE Environment. Design: Design concepts, Architectural design, Component based design, User interface design.				
Module 3	Agile Principles & Devops (Knowledge level)	Quiz		09 Hours
Agile: Scrum Roles and activities, Sprint Agile software development methods - Scaling, User Stories, Agile estimation techniques, Product backlogs, Stake holder roles, Dynamic System Development Method. Devops: Introduction, definition, history, tools.				
Module 4	Software Testing and Maintenance (Application Level)	Assignment	Apply the testing concepts using Programing	12 Hours
Software Testing-verification and validation, Test Strategies - White Box Testing, Black box Testing. Automation Tools for Testing. Software Quality Assurance-Elements of software quality assurance, SQA Tasks, Goals and Metrics, Software configuration management- SCM process, SCM Tools (GitHub). Maintenance- Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models.				
Targeted Application & Tools that can be used: Selenium, GitHub, CASE Tools				
Text Book 1] Roger S. Pressman, “Software Engineering – A Practitioner’s Approach”, VII Edition, McGraw-Hill, 2017. 2] Bob Hughes, Mike Cotterell, Rajib Mall, “Software Project Management”, VI Edition, McGraw-Hill, 2018.				
References Rajib Mall, “Fundamentals of Software Engineering”, VI Edition, PHI learning private limited, 2015. Ian Sommerville, “Software Engineering”, IX Edition, Pearson Education Asia, 2011. Agile Software Development Principles, Patterns and Practices.1st Edition, Wiley, 2002				
Topics Relevant to “Skill Development: Balck box Testing, White box Testing, Automated Testing for Skill development through Participative Learning Techniques. This is attained through assessment mentioned in the course handout				

Course Code: ECE2001	Course Title: Innovation Project-Raspberry Pi Using Python Type of Course: School Core & Practical Only.	L- T- P- C	0	4 This includes few lecture sessions	2
Version No.	1.0				
Course Pre-requisites	NIL				
Anti-requisites	NIL				

Course Description	The Raspberry Pi is an amazing single board computer (SBC) capable of running Linus and a whole host of applications. Python is a beginner-friendly programming language that is used in schools, web development, scientific research, and in many other industries. This course will enable students in writing own programs with Python to blink lights, respond to button pushes, read sensors, log data on the Raspberry Pi and many more. The course also offers in-depth knowledge of designing, developing, coding and implementing projects using Raspberry Pi.			
Course Outcomes	On successful completion of this course the students shall be able to: Write a program in Python. Explain the main features of the Raspberry Pi board Demonstrate the hardware interfacing of the peripherals to Raspberry Pi system. Demonstrate the functioning of live various projects carried out using Raspberry Pi system.			
Course Content:				
Module 1	Basics of Python, functions	Quiz	Problem Solving	4 Lab Sessions
Topics: Introduction, Structure of Python Program, Data Types and Variables, Input and Output, Operators, Importing libraries, Functions, Development Tool. Concepts will be taught by solving problems through programs.				
Module 2	Python Programming	Quiz	Problem Solving	4 Lab Sessions
Control statements, Lists and Dictionaries, Problem solving using Python. Concepts will be taught by solving problems through programs.				
Module 3	Overview of Raspberry Pi	Project Development	System Design Task and Analysis	4 Lab Sessions
Topics: An exploration of GPIO pins, LED and switch control. Installation of libraries, PuTTY SSH. Raspberry Pi to interface with more complicated sensors and actuators like Pi Camera, servo motor ADS51115 through PIP libraries. Arduino with Raspberry-pi				
Module 4	Interaction with API Services	Project Development	Modeling and Simulation task	3 Lab Sessions
Topics: Raspberry Pi interact with online API services through the use of public APIs and SDKs using Firebase, Gspread API. Node-RED – a programming tool for wiring together hardware devices, MQTT. Android/Case study.				
Targeted Application & Tools that can be used: Making it a reality (Raspberry Pi Projects) : Projects will include but not limited to : 1) Intelligent home locking system. 2) Intelligent water level management system. 3) Home automation using RFID.				

4) Real time clock-based home automation. 5) Intelligent Automatic Irrigation System Professionally Used Software: Raspberry Pi.	
Project work/Python Lab Test:	
Project work Python test.	
Text Book(s): 1) Ashok Namdev Kamthane, Amit Ashok Kamthane, “Problem Solving and Python Programming”, Mc Graw Hill Education, 2018.	
Reference(s): https://github.com/thibmaek/awesome-raspberry-pi MagPi magazine	
Topics relevant to development of “Foundation Skills”: Basic Concepts of Python-Programming, and Raspberry Pi. Topics related to development of “Employability Skills”: Problem solving, Creative Thinking, Team work, Prototype Development. Topics related to development of “Entrepreneurship”: Effective Communication, Strategic Thinking, Creative Thinking.	
Evaluation:	Review-1-20%, Review-2-25%, Python test-25%, Project Expo-30%

Course Code: CSE1005	Course Title: Programming in Python Type of Course: School Core Lab Integrated	L- T-P-C	1	0	4	3
Version No.	1.0					
Course Pre-requisites	Basic knowledge of Computers and Mathematics					
Anti-requisites	NIL					
Course Description	The purpose of this course is to enable the students to develop python scripts using its basic programming features and also to familiarize the Python IDLE and other software’s. This course develops analytical skills to enhance the programming abilities. The associated laboratory provides an opportunity to validate the concepts taught and enhances the ability to build real time applications.					

Course Object	The objective of the course is to familiarize the learners with the concepts of Programming in Python and attain Employability through Problem Solving Methodologies.			
Course Outcomes	On successful completion of this course the students shall be able to: Summarize the basic Concepts of python. 2. Demonstrate proficiency in using data structures. 3. Illustrate user-defined functions and exception handling. 4. Identify the various python libraries.			
Course Content:				
Module 1	Basics of Python programming	Assignment	Programming	14 Classes
Topics: Data types, operators and Expressions, Input and Output Statements. Control Structures – Selective and Repetitive structures				
Module 2	Indexed and Associative Data Structures	Simple applications	Programming	20 Classes
Topics: Strings, Lists, Sets, Tuples, Dictionaries				
Module 3	Functions, Exception handling and libraries	Case study	Programming	10 Classes
Topics: User defined functions, exception handling, Introduction to python built-in libraries				
Targeted Application & Tools that can be used: Targeted Application : Web application development, AI, Operating systems Tools: Python IDLE, ANACONDA Application Areas: Web Development Game Development Scientific and Numeric Applications Artificial Intelligence and Machine Learning Software Development Enterprise-level/Business Applications Education programs and training courses Language Development Operating Systems Web Scrapping Applications Image Processing and Graphic Design Applications Professionally Used Software: Python IDLE, Spyder, Jupyter Notebook, Google Colab				
Project work/Assignment:				

Project Assignment: Developing python scripts using built in methods and functions
Text Books: Martin C. Brown, “Python: The Complete Reference”, McGraw Hill Education, Forth edition (20 March 2018). Alex Campbell, “Python for Beginners: Comprehensive Guide to the Basics of Programming, Machine Learning, Data Science and Analysis with Python”, August 29, 2021. Charles Dierbach, “Introduction to Computer Science Using Python”, Wiley India Edition,2015.
References: E. Balagurusamy, “Introduction to Computing and Problem Solving Using Python”, Tata McGraw-Hill, 2016 Y. Daniel Liang, “Introduction to Programming Using Python”, Pearson, 2017 Brady Ellison, “Python for Beginners: A crash course to learn Python Programming in 1 Week (Programming Languages for Beginners)”, August 25, 2021. Python Tutor - Visualize Python, Java, C, C++, JavaScript, TypeScript, and Ruby code execution https://practice.geeksforgeeks.org/courses/Python-Foundation
Topics relevant to development of “FOUNDATIONS SKILLS”- Solve the real time problems by analyzing and visualizing the data. Topics relevant to “HUMAN VALUES & PROFESSIONAL ETHICS”- Data collection and its arrangement

Course Code: PPS4002	Course Title: Introduction to Aptitude Type of Course: Practical Only Course	L- P- C	0	2	1
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	02 Hours
Topics: Introduction to Aptitude, working of Tables, Squares, Cubes				
Module 2	Logical Reasoning	Assignment	Bloom’s Level : Application	18 Hours
Topics: Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Ordering and Ranking, Clocks and Calendars, Number Series, Wrong number series, Visual Reasoning				
Targeted Application & Tools that can be used: Application area: Placement activities and Competitive examinations. Tools: LMS				
Text Book Quantitative Aptitude by R S Aggarwal Verbal & Non-Verbal Reasoning by R S Aggarwal				
References www.indiabix.com www.youtube.com/c/TheAptitudeGuy/videos				
Topics relevant to Skill development: Quantitative and reasoning aptitude for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.				

Course Code: MAT2003	Course Title: NUMERICAL METHODS FOR ENGINEERS Type of Course: School Core	L-T- P- C	1	0	2	2
Version No.	1.0					

Course Pre-requisites	MAT1002 – Transform Techniques, Partial Differential Equations and Their Applications			
Anti-requisites	Nil			
Course Description	The course focuses on formulating and solving problems concerning real-world engineering applications numerically as well as statistically. This course provides an introduction to basic numerical methods to deal with algebraic and transcendental equations, system of equations, interpolation, differentiation and integration. This course also deals with numerical solution of ordinary differential equations by means of Taylor's series method, modified Euler's method and Runge-Kutta methods.			
Course Objective	The objective of the course is to familiarize the learners with the concepts of " NUMERICAL METHODS FOR ENGINEERS" and attain Skill Development Through Problem Solving.			
Course Outcomes	On successful completion of the course the students shall be able to: 1] Solve algebraic and transcendental equations numerically. 2] Adopt numerical techniques to differentiate and integrate functions. 3] Apply numerical methods to solve ordinary differential equations.			
Course Content:				
Module 1	Numerical solution of Algebraic and Transcendental Equations			15 Classes
Algebraic and Transcendental Equations, Regula - Falsi method, Bisection method (Self study), Secant method, Newton-Raphson method, and NR method for non-linear Equations, Fixed-point iteration method. System of Linear Equations: Introduction, LU decomposition method, Gauss-Jacobi method, Gauss-Seidel iteration method, Largest Eigen value and corresponding Eigen vector by Power method & Jacobi Method.				
Module 2	Numerical Interpolation, differentiation and Integration			15 Classes
Numerical Interpolation: Newton's forward and backward interpolation method, Newton's divided difference method, Lagrange's method, numerical differentiation. Numerical integration: Trapezoidal rule, Simpson's one-third rule, Simpson's three-eighth rule, Weddle's Rule. Area between the two curves.				
Module 3	Numerical solution of ODEs and PDEs			15 Classes
Solution of ordinary differential equations: Initial Value problems: Taylor's series method, Picard's method, Euler's Method, Modified Euler's method, Runge-Kutta method, Milne's predictor-corrector formula. Adams -Bashforth method, Boundary value problems - Finite difference methods for ODE. Numerical solution for LCR & damped forced oscillatory equations.				

Solution of partial differential equations: Schmidt Explicit Formula for Heat Equation, Crank-Nicolson method. Numerical solution to Wave, Laplace & Heat Equation.

Targeted Application & Tools that can be used:

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

Assignment:

Gauss-Jacobi iteration method.
Numerical differentiation.
Gaussian quadrature rule for numerical integration.
Taylor series method for ODEs.
Implicit and explicit schemes for PDEs.

Text Books

T1: M.K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering Computations, 6th Edition, New age Publishing House, 2015.

T2: Erwin Kreyszig, “Advanced Engineering Mathematics”, 10th Edition, John Wiley& Sons (India), 2014.

References:

R1: B.S. Grewal, Numerical methods in engineering and science, 10th Edition, Khanna publishers, 2016.

R2: B.S. Grewal, “Higher Engineering Mathematics”, 44th edition, Khanna Publishers.

R3: Steven C Chapra and Raymond P Canale, “Numerical Methods for Engineers,” 7th Ed., McGraw-Hill Edition, 2015.

R4: C. Ray Wylie and Louis C Barrett, “Advanced Engineering Mathematics”, 6th Edition, McGraw-Hill, 2012.

Topics relevant to SKILL DEVELOPMENT: This course focuses on formulating and solving problems concerning real-world engineering applications numerically as well as statistically. This course provides an introduction to basic numerical methods to deal with algebraic and transcendental equations, system of equations, interpolation, differentiation and integration with numerical solution of ordinary differential equations by means of Taylor’s series method, modified Euler’s method and Runge-Kutta methods for Skill Development through Problem Solving methodologies. This is attained through assessment component mentioned in course handout.

Course Code: CSE2007	Course Title: Design and Analysis of Algorithms Type of Course: Program Core & Theory only		L-T-P-C	3	0	0	3
Version No.		2.1					

Course Pre-requisites		CSE2001, Data Structure and Algorithms
Anti-requisites		NIL
Course Description		This intermediate course enables students to design and analyze efficient algorithms to solve problems. This course covers typical design methods such as divide-and-conquer, dynamic programming and greedy method to solve problems. The students shall develop strong analytical skills as part of this course.
Course Objectives		This course is designed to improve the learners' EMPLOYABILITY SKILLS by using PROBLEM SOLVING Methodologies.
Course Outcomes		On successful completion of the course the students shall be able to: 1] Identify the efficiency of a given algorithm. [Comprehension] 2] Employ divide and conquer approach to solve a problem. [Application] 3] Illustrate dynamic programming approach to solve a given problem. [Application] 4] Solve a problem using the greedy method. [Application] 5] Discuss the techniques to solve a real-world problem based on its complexity classes. [Comprehension]
Course Content:		

Module 1	Introduction to Algorithms	Assignment		Problem Solving	06 Sessions
	Topics: Algorithm Design and efficiency, measuring of running time of algorithms. Insertion sort and merge sort, Asymptotic Growth and Notations. Recurrences--Masters method. Assignment: Comparatively evaluate bubble sort, insertion sort and mergesort.				
Module 2	Review of Searching and Sorting techniques	Assignment		Programming/ Problem Solving	12 Sessions
	Topics: Divide and Conquer: Examples. Strassen’s Matrix multiplication. Sorting: Quicksort, Heapsort, Lower bound of comparison-based sorting, non-comparison-based sorting: Radix sort. Search: Review of Linear Search and Binary Search, Hashing and hash tables. Assignment: Design and develop an algorithm using Divide and Conquer technique for a given scenario.				
Module 3	Greedy Algorithms	Assignment		Programming/ Problem Solving	09 Sessions
	Topics:				

	Introduction, Fractional Knapsack Problem, Minimal Spanning Tree: Prim’s Algorithm and Kruskal’s Algorithm, Single-source Shortest Path: Dijkstra’s Algorithm. Huffman Codes. Assignment: Design and Develop a solution to a given scenario using greedy method.				
Module 4	Dynamic Programming	Assignment		Programming/ Problem Solving	09 Sessions
	Topics: Introduction with examples, Principles of Memoization, 0-1 Knapsack Problem, Bellman-Ford algorithm, Floyd-Warshall’s Algorithms. Optimal Binary Search Trees, Chain Matrix Multiplication. Assignment: For a given scenario, attempt the three design paradigms learned so far and argue the best approach to solve the problem				
Module 5	Complexity Classes and Heuristics	Assignment		Programming/ Problem Solving	09 Hours
	Topics: Complexity classes: P, NP, and NP-Complete Problems. Backtracking: n-Queens. Branch and bound: Travelling Salesman Problem. Assignment: Apply backtracking algorithmic designing technique for solving queen’s problems for 4, 8 and 16 inputs.				
	Targeted Application & Tools that can be used: Application Area is to Design and Analyzing the efficiency of Algorithms. This fundamental course is used by all application developers. Professionally Used Software: GCC compiler.				
	Project work/Assignment:				
	Problem Solving: Design of Algorithms and implementation of programs. Programming: Implementation of given scenario using Java.				
	Text Book: T1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, ‘Introduction to Algorithms’, MIT Press, 2022. T2. J. Kleinberg and E. Tardos, ‘Algorithm Design’, Addison-Wesley, 2005.				
	References R1. Anany Levitin, ‘Introduction to the Design and Analysis of Algorithms’, Pearson Education, 2003. R2. Tim Roughgarden, ‘Algorithms Illuminated’ (books 1 through 3), Soundlikeyourself Publishing, 2017,18,19 respectively. R3. AV Aho, J Hopcroft, JD Ullman, ‘The Design and Analysis of Algorithms’, Addison-Wesley, 1974.				

Course Code: CSE3156	Course Title: Database Management Systems Type of Course: 1) School Core 2) Laboratory Integrated	L-T-P-C	3	0	2	4
Version No.	1.0					

<table>
<tr><td>Course Pre-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Anti-requisites</td><td colspan="4">NIL</td></tr>
<tr><td>Course Description</td><td colspan="4">This course introduces the core principles and techniques required in the design and implementation of database systems. It covers concepts of relational database systems (RDBMS). More emphasis is set on how to design, develop, organize, maintain and retrieve information efficiently. It helps the students to learn and practice data modeling and database designs. The course also introduces the concept of object oriented and object relational databases.
The associated laboratory is designed to implement database design using MySQL DATABASE in information technology applications. All the exercises will focus on the fundamentals for creating, populating, sophisticated, interactive way of querying, and simultaneous execution of the transactions of database.</td></tr>
<tr><td>Course Objective</td><td colspan="4">The objective of the course is to familiarize the learners with the concepts of Database Management Systems and attain Employability through Problem Solving Methodologies.</td></tr>
<tr><td>Course Out Comes</td><td colspan="4">On successful completion of the course the students shall be able to:
1] Demonstrate a database system using ER model and relational algebra. [Understanding] 2] Build databases using SQL queries query processing. [Applying]
Apply the functional dependencies and design the database using normalization. [Applying]
Interpret the concept of object-oriented databases and object-relational databases. [Understanding]</td></tr>
<tr><td>Course Content:</td><td colspan="4"></td></tr>
<tr><td>Module 1</td><td>Introduction to Database Modelling and Relational Algebra (Understanding)</td><td>Assignment</td><td>Problem Solving</td><td>8 Classes</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 2</td><td>Fundamentals of SQL and Query Optimization (Applying)</td><td>Assignment</td><td>Programming</td><td>8 Classes</td></tr>
</table>

<table>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 3</td><td>Relational Database Design & Transaction Management (Applying)</td><td>Assignment</td><td>Problem Solving</td><td>12 Classes</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 4</td><td>Advanced DBMS Topics (Understanding)</td><td>Assignment</td><td>Case Study</td><td>8 Classes</td></tr>
<tr><td colspan="5">Topics:
Advanced topics: Object oriented database management systems, Deductive database management systems, Spatial database management systems, Temporal database management systems, Constraint database management systems.
New database applications and architectures such as Data warehousing, Multimedia, Mobility, NoSQL, Native XML databases (NXD), Document-oriented databases, Statistical databases.</td></tr>
</table>

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

Experiment No. 4: [2 Session]
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]
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]
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]
To implement the concept of forms and reports. Level 1: Implement the concept of forms and reports. Level 2: Analyze the schema relationship.

Labsheet-6 [2 Practical Sessions] Experiment No. 8: [2 Sessions]
Design a mini project based on the databases such as Inventory Management System, University Management System, Hospital Management System, etc.
Level 1: Implement the real time database.
Level 2: Analyze the working of database in real time.

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

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

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

Text Book
RamaKrishna & Gehrke, “Database Management Systems” 3rd Edition, 2018, McGraw-Hill Education.
Avi Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, McGraw-Hill ,7th Edition, 2019. 3] 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
Elmasri R and Navathe S B, “Fundamentals of Database System”, Pearson Publication, 7th Edition, 2018.
M. Kleppmann, “Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems”, O’Reilly, 2017.

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

Course Code: CSE3351	Course Title: Operating Systems Type of Course: Program Core and Theory Only	L-T- P- C	3	0	0	3
Version No.	1.0					
Course Pre-requisites	CSE2009- Computer Organization, Problem solving using C Students should have basic knowledge on computers, computer software & hardware, and Computer Organization. Prior programming experience in C is recommended.					
Anti-requisites	NIL					
Course Description	This course introduces the concepts of operating system operations, operating system structure and its design and implementation. It covers the classical operating systems internal algorithms such as process scheduling, synchronization, deadlocks detection and recovery and memory management. The course also enhances the problem solving, systems programming ability and case studies.					
Course Object	The objective of the course is to familiarize the learners with the concepts of Operating Systems and attain Employability through Problem Solving Methodologies.					
Course Out Comes	On successful completion of the course the students shall be able to: 1] Describe the fundamental concepts of operating Systems and case studies. [Knowledge] 2] Demonstrate various CPU scheduling algorithms. .[Application] 3] Apply various tools to handle synchronization problems.[Application] 4] Demonstrate deadlock detection and recovery methods [Application] 5] Illustrate various memory management techniques.[Application]					
Course Content:						

Module 1	Introduction to Operating System	Assignment	Programming	9 Hours
Topics: Introduction to OS , Operating-System Operations, Operating System Services, , System Calls and its types, Operating System Structure, System Program and its types, Linkers and Loaders, Overview of OS design and implementation, Open-source operating system				

<table>
<tr><td>Module 2</td><td>Process Management</td><td>Assignment/Case Study</td><td>Programming/Simulation</td><td>11 Hours</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td>Module 3</td><td>Process Synchronization and Deadlocks</td><td>Assignment</td><td>Programming</td><td>11 Hours</td></tr>
<tr><td colspan="5">Topics:
The Critical-Section Problem- Peterson's Solution, Synchronization hardware, Semaphores, Classic Problems of Synchronization with Semaphore Solution- Producer-Consumer Problem, Reader-Writer problems, Dining Philosopher's Problem, . Introduction to Deadlocks, Necessary conditions for deadlock, Resource allocation Graph, Methods for handling deadlock: Deadlock Prevention and Implementation, Deadlock Avoidance and Implementation, Deadlock detection & Recovery from Deadlock.</td></tr>
<tr><td>Module 4</td><td>Memory Management</td><td>Assignment</td><td>Programming/Simulation</td><td>10 Hours</td></tr>
<tr><td colspan="5">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.</td></tr>
<tr><td colspan="5">Targeted Application:
Application area is traffic management system, banking system, health care and many more systems where in there are resources and entities that use and manage the resources.

Software Tools:
Oracle Virtual Box/VMWare Virtualization software [Virtual Machine Managers]. Used to install and work on multiple guest Operating Systems on top of a host OS.

Intel Processor identification utility: This software is used to explain about multi-core processors. It helps to identify the specifications of your Intel processor, like no of cores, Chipset information, technologies supported by the processor etc.</td></tr>
<tr><td colspan="5">Project work/Assignment</td></tr>
<tr><td colspan="5">Demonstrate process concepts in LINUX OS.
Simulation of CPU scheduling algorithms.
Develop program to demonstrate use of Semaphores in threads.
Develop program to demonstrate use of deadlock avoidance algorithms.
Develop program to demonstrate use of page replacement algorithms.
Simulation of memory allocation strategies [first fit, best fit and worst fit].</td></tr>
<tr><td colspan="5">Text Book
Silberschatz A, Galvin P B and Gagne G , "Silberschatz's Operating System Concepts", Paperback, Global Edition Wiley, 2019</td></tr>
</table>

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

Course Code: CSE 3078	Course Title: Cryptography and Network Security Type of Course: Program Core & Theory only	L- T- P- C	3	0	0	3
Version No.		1				
Course Pre-requisites		“Data Communications 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. Topics: 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: Identifies the basic concept of Cryptography (Knowledge) CO2: Express the different types of Cryptographic Algorithms. (Comprehension) CO3: Recognize the Public key Cryptographic Techniques for various applications. (Comprehension) CO4: Apply the network security concepts during their implementation of network security application developments. (Application)					
Course Content:						
Module 1	Introduction to Cryptography	Assignment	Identify the Concepts	08 Sessions		
Topics:						

<table>
<tr><td colspan="5">Introduction to Cryptography, Model of Network Security, OSI Security architecture, Security Attacks: active attacks, passive attacks, services: Authentication, Access Control, Data Confidentiality, Data Integrity, Nonrepudiation, Substitution Ciphers : Caesar, Mono alphabetic, Polyalphabetic, Play-fair and Hill Cipher, Introduction to Block Cipher and Stream Cipher, Festal Structure.</td></tr>
<tr><td>Module 2</td><td>Private Key Cryptography and Number Theory</td><td>Assignment</td><td>Analysis of requirement of complexity in cryptography</td><td>13 Sessions</td></tr>
<tr><td colspan="5">Topics:
Symmetric Encryption Algorithms : Data Encryption Standard, Introduction to Galois Field, Advanced Encryption Standard, Modular Arithmetic, Prime numbers, Fermat's little theorem, brief about primality testing and factorization, Discrete Logarithmic Problem, Euclidean and Extended Euclidean Algorithm, Euler Totient Function, Chinese Remainder Theorem</td></tr>
<tr><td>Module 3</td><td>Public Key Cryptography and its Applications</td><td>Assignment</td><td>Recognize the importance of various security concepts to achieve sufficient solutions</td><td>10 Sessions</td></tr>
<tr><td></td><td colspan="4">Topics:
Overview of Public Key Cryptography, RSA, Diffie - Helman Key exchange, Man in the middle attack, Cryptographic Hash functions, Secure Hash Algorithm, Message Authentication Codes – HMAC, Digital Signature, Discussion on real time practices of Cryptography.</td></tr>
<tr><td>Module 4</td><td>Network Security</td><td>Assignment</td><td>Implement the advanced network security algorithms in recent applications.</td><td>07 Sessions</td></tr>
<tr><td colspan="5">Topics:
Network Security fundamentals, Network Security applications: Authentication: Kerberos, PKI, Network Security applications: e-mail security: PGP, MIME, Network Security applications: IP Security: IP Sec architecture, Network Security applications: Web Security.</td></tr>
<tr><td colspan="5">Targeted Application & Tools that can be used:
Students get the knowledge about cryptography techniques followed, the algorithms used for encryption and decryptions & the techniques for authentication and confidentiality of messages.</td></tr>
<tr><td colspan="5">Assignment:</td></tr>
<tr><td colspan="5">Assignment 1: Solve the problems of basic encryption techniques.
Assignment 2: Solve and analyze the problems on symmetric and asymmetric encryption.</td></tr>
<tr><td colspan="5">Textbooks:
1.William Stallings, "Cryptography and Network Security - Principles and Practices", Prentice Hall, 8th Edition, 2019.
2. Wade Trappe and Lawrence C Washington, "Introduction to Cryptography with Coding Theory", Pearson, 2020.
Reference Books:
1.Behrouz A Forouzan, Debdeep Mukhopadhyay, "Cryptography and Network Security", McGraw Hill, third edition, 2010.</td></tr>
</table>

2. R.Rajaram, “Network Security and Cryptography” SciTech Publication.3rd Edition, 2014. 3. AtulKahate, "Cryptography and Network Security", Tata McGraw-Hill, 2nd Edition, 2019. 4. BruceSchneier, “Applied Cryptography", John Wiley and Sons Inc. Second Edition, 2015.
Web references: 1.https://onlinecourses.nptel.ac.in/noc22_cs90/preview 2.e-pgpathshala UGC lecture series : E-Series and Self learning Materials. https://epgp.inflibnet.ac.in/Home/ViewSubject?catid=fBYckQKJvP3a/8Vd3L08tQ== 3. http://182.72.188.195/cgi-bin/koha/opac-detail.pl?biblionumber=10133&query_desc=kw%2Cwrdl%3A%20Cryptography%20and%20Network%20Security 4.http://182.72.188.195/cgi-bin/koha/opac-detail.pl?biblionumber=5875&query_desc=kw%2Cwrdl%3A%20Cryptography%20and%20Network%20Security.
Topics relevant to “Skill Development”: Symmetric and Asymmetric Encryption Algorithms and its problems.

Course Code: PPS4004	Course Title: Aptitude Training-Intermediate Type of Course: Practical Only Course	L-T P-C	0	0	2	1
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 (Undergraduate). 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 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)					
Course Content:						
Module 1	Quantitative Ability	Assignment		24 Hours		
Topics:						

Number System, Percentage, Ratio and Proportion, Average, Mixture and Allegation, Time and Work, Profit and Loss, Time Speed and Distance, Boats and Streams, Simple Interest and Compound Interest, Probability, Permutation and Combination.				
Targeted Areas Application area: Placement activities and Competitive examinations. Tools: LMS				
Text Book Fast Track Objective by Rajesh Verma R S Aggarwal Rakesh Yadav				
References www.indiabix.com www.testbook.com www.youtube.com/c/TheAptitudeGuy/videos				
Topics relevant to Skill development: Quantitative aptitude for Skill Development through Problem solving Techniques. This is attained through assessment component mentioned in course handout.				
Evaluation – Continuous Evaluation (Topic wise evaluation Mid-Term & End term)				

Course Code: CSE3216	Course Title: Mastering Object- Oriented Concepts in Python Type of Course: Lab	L- T- P- C	0-0-2-1
Version No.	1		
Course Pre-requisites	CSE1005 – Programming in Python		
Anti-requisites	NIL		
Course Description	This course covers mastering object-oriented concepts in Python, including classes, inheritance, polymorphism, and encapsulation. Students will learn to design and implement robust, reusable code using real-world examples. Ideal for those with basic Python knowledge, it enhances problem-solving skills and software development proficiency.		
Course Objective	The objective of the course is to familiarize the learners with the concepts of Mastering Object Oriented Concepts in Python and attain Skill Development through Experiential Learning.		
Course Out Comes	CO1: Explain features of Oops along with creation of Python classes and objects to represent real world Objects. [Understand] CO2: Demonstrate inheritance, polymorphism, and abstraction in Python to build maintainable and extendable software systems.[Apply]		

	CO3: Demonstrate exception handling in Python to build robust error-handling mechanisms and debugging tool and Assess various file handling techniques in Python. [Apply]			
Course Content:				
Module 1	Introduction to OOPS, Classes and Objects	MCQ	Assignment	10 Sessions
Topics: Introduction to OOPs: Problems in Procedure Oriented Approach, Specialty of Python Language, Features of OOPS - Classes and Objects, Encapsulation, Abstraction, Inheritance and Polymorphism. Classes and Objects: Creating a Class, The Self Variable, Constructor, Destructors, Types of Variables, Namespaces, Types of Methods - Instance Methods, Class Methods, Static Methods, Passing Members of One Class to Another Class, Inner Classes.				
Module 2	Inheritance and Polymorphism	MCQ	Assignment	10 Sessions
Constructors in Inheritance, Overriding Super Class Constructors and Methods, The Super() Method, Types of Inheritance – Single Inheritance, Multiple Inheritance, Method Resolution Order(MRO), Polymorphism, Duck Typing Philosophy of Python, Operator Overloading, Method Overloading, Method Overriding. Abstract Classes and Interfaces: Abstract Method and Abstract Class, Interfaces in Python, Abstract Classes vs. Interfaces.				
Module 3	Exceptions and Files in Python	MCQ	Assignment	10 Sessions
Exceptions: Errors in a Python Program – Compile-Time Errors, Runtime Errors, Logical Errors. Exceptions, Exception Handling, Types of Exceptions, The Except Block, The assert Statement, User-Defined Exceptions, Logging the Exceptions. Files in Python: Files, Types of Files in Python, Opening a File, Closing a File, Working with Text Files Containing Strings, Knowing whether a File Exists or Not, Working with Binary Files, The with Statement, Pickle in Python, The seek() and tell() Methods.				
Targeted Application & Tools that can be used: Python, PyCharm				
Project work/Assignment:				
Assignment: Module 1 Assignment: Design and implement a Python application that simulates a banking system using classes and methods for customers and accounts. Module 2 Assignment: Develop a Python application that simulates Library management system that demonstrates inheritance, polymorphism and abstraction concepts. Module 3 Assignment: Develop a Python program that handles different types of exceptions while processing user input for a movie ticket booking system showcasing exception handling and File handling concepts.				
Text Book Dr. R Nageshwara Rao, "Core Python Programming", Dreamtech Press, 3rd Edition, 2021.				

References
Alex Martelli, Anna Ravenscroft & Steve Holden, “Python in a Nutshell The Definitive Reference”, O'Reilly Media, 3rd edition, 2017.
Luciano Ramalho, “Fluent Python Clear, Concise, and Effective Programming”, O'Reilly Media, 2nd edition, 2022.
Mark Lutz, “Learning Python: Powerful Object-Oriented Programming”, O'Reilly Media, 5th edition, 2013.
David Beazley, Brian K. Jones, “Python Cookbook: Recipes for Mastering Python 3”, O'Reilly Media, 3rd edition, 2013.
Weblinks:
www.learnpython.org
https://realpython.com/python3-object-oriented
https://www.tutorialspoint.com/python/python_oops_concepts.htm

Topics relevant to “SKILL DEVELOPMENT”:
Building Real-World Applications Using OOPS Concepts, Error Handling and Debugging Techniques, Concurrency in Python, Advanced File Handling Techniques, Creating and Managing Python Packages and Modules, Designing and Implementing Python Interfaces
This is attained through assessment component mentioned in course handout.

Course Code: CBC2500	**Course Title: Smart Contract and Solidity Type of Course: Integrated**	**L- T- P- C**	3-0-0-3	
Version No.	1			
Course Pre-requisites	Basics of Mathematics and any Programming Language			
Anti-requisites	NONE			
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	The objective of the course is to familiarize the learners with the concepts of **Smart Contract and Solidity** and attain **EMPLOYABILITY** through **Experiential Learning Techniques.**.			

<table>
<tr><td><b>Course Out Comes</b></td><td colspan="4"><b>On successful completion of the course the students shall be able to:</b>
<b>CO 1 :</b>Understand the fundamentals of computational Element of the Blockchain Technology
<b>C.O 2:</b> Implementuser-defined operations of arbitrary complexity that are not possible through plaincryptocurrency protocols
<b>C.O 3:</b> Exhibitbest practices for designing solutions with smart contracts using Solidity and Remix IDE</td></tr>
<tr><td><b>Module 1</b></td><td>Introduction to Smart Contract</td><td>TEST-1</td><td>Fundaments of Smart Contractand Solidity</td><td><b>12Sessions</b></td></tr>
<tr><td colspan="5"><b>Topics:</b> A Simple Smart Contract, Blockchain Basics, The Ethereum Virtual Machine, Versioning, Remix, npm
/ Node.js, Docker, Binary Packages, Building from Source, CMake options.</td></tr>
<tr><td><b>Module 2</b></td><td>Solidity in Depth</td><td>TEST-1</td><td>Case studies / Case let</td><td><b>12 Sessions</b></td></tr>
<tr><td colspan="5"><b>Topics:</b> 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</td></tr>
<tr><td><b>Module 3</b></td><td>Contract Metadata & ContractABI Specification</td><td>Endterm lab Exam</td><td>Implementing Applications</td><td><b>14 Sessions</b></td></tr>
<tr><td colspan="5"><b>Topics:</b> 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</td></tr>
<tr><td colspan="5">Textbooks:
T1: William Stallings, *Cryptography and Network Security: Principles and Practice*, Pearson, 7th Edition, 2017
T2: Behrouz A. Forouzan, *Cryptography and Network Security*, McGraw-Hill Education, 2nd Edition, 2011

Reference Books:
R1: Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 2nd Edition, 1996
R2: Douglas R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 4th Edition, 2018
R3: Charlie Kaufman 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

Web Resources:</td></tr>
</table>

W1: https://cryptography.io W2: https://nvlpubs.nist.gov W3: https://www.owasp.org W4: https://www.tutorialspoint.com/cryptography/index.htm W5: https://www.coursera.org/learn/crypto

Course Code: CBC2501	**Course Title: Smart Contract and Solidity Lab** **Type of Course: Integrated**	**L- T- P- C**	0-0-2-1	
Version No.	1			
Course Pre-requisites	Basics of Mathematics and any Programming Language			
Anti-requisites	NONE			
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	The objective of the course is to familiarize the learners with the concepts of **Smart Contract and Solidity** and attain **EMPLOYABILITY** through **Experiential Learning Techniques.**.			
Course Out Comes	**On successful completion of the course the students shall be able to:** **CO 1 :**Understand the fundamentals of computational Element of the Blockchain Technology **C.O 2:** Implementuser-defined operations of arbitrary complexity that are not possible through plaincryptocurrency protocols **C.O 3:** Exhibitbest practices for designing solutions with smart contracts using Solidity and Remix IDE			
Module 1	Introduction to Smart Contract	TEST-1	Fundaments of Smart Contractand Solidity	**12Sessions**
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	TEST-1	Case studies / Case let	**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 & ContractABI Specification	Endterm lab Exam	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

List of 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	**Network traffic capture and analysis for secure**	**Wireshark**

14	**protocols**	
Week 15	**Mini project: End-to-end secure message exchange with key management**	**Python + GPG + OpenSSL**

🛠 Open-Source Tools Overview

- **CrypTool – Educational tool for learning cryptography visually**
- **Python (PyCryptodome, hashlib, cryptography) – For programmatic encryption/decryption**
- **OpenSSL – Command-line utility for SSL, AES, RSA operations**
- **GnuPG – For creating and verifying digital signatures and PGP**
- **Wireshark – Packet sniffer for analyzing network-level security**

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

Reference Books:
R1: Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 2nd Edition, 1996
R2: Douglas R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 4th Edition, 2018
R3: Charlie Kaufman 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

Web Resources:
W1: https://cryptography.io
W2: https://nvlpubs.nist.gov
W3: https://www.owasp.org
W4: https://www.tutorialspoint.com/cryptography/index.htm
W5: https://www.coursera.org/learn/crypto

| Course Code: CBC2502 | Course Title: Distributed Ledger Technology | L:T:P:C = 3:0:0:3

Version No: 1.0

Course Pre-requisites:

Anti-requisites: NIL

Course Description:
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.

Course Objectives:

1. Understand the fundamentals and types of distributed ledger technologies.
2. Explore blockchain architecture and consensus mechanisms.
3. Analyze security, scalability, and privacy in decentralized systems.
4. Apply DLT in enterprise and public applications through smart contracts and platforms.

Course Outcomes:

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

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

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

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

Course Content:

Module 1: Introduction to DLT and Blockchain | No. of Sessions: 11
Distributed Ledger vs Centralized Systems, Blockchain overview, Key features of DLT (immutability, transparency, decentralization), Cryptographic primitives, Merkle trees, Hash functions, Public vs Private blockchains, Peer-to-peer networking

Module 2: Blockchain Architecture and Consensus Mechanisms | No. of Sessions: 12
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 | No. of Sessions: 11
Introduction to smart contracts, Ethereum architecture, Solidity basics, Hyperledger Fabric architecture and components, Chaincode in Go/Node.js, Smart contract lifecycle, Oracles and off-chain data access

Module 4: Applications, Security, and Industry Use Cases | No. of Sessions: 11
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

Textbooks:
T1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020
T2: Melanie Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly Media, 2015

Reference Books:
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

Web Resources:
W1: https://ethereum.org/en/developers/
W2: https://hyperledger.org/use/fabric
W3: https://www.ibm.com/blockchain
W4: https://www.blockchain-council.org/
W5: https://developer.algorand.org/

+==
==+

Course Code: CSE2264	**Course Title: Essentials** of AI	**L- T-P- C**	3	**0**	0	**3**
Version No.	2.0					
Course Pre-requisi Data **tes**	Basic knowledge of programming, mathematics, understanding of data handling					
Anti-requisites	**NIL**					
Course Description	This course is a comprehensive introductory course designed to equip learners with the fundamental Python programming skills necessary to work with artificial intelligence (AI) technologies. This course is aimed at individuals who are new to AI but have a basic understanding of programming concepts. It combines Python programming fundamentals with hands-on experience in implementing AI techniques such as machine learning, neural networks, and natural language processing.					

<table>
<tr><td><b>Course Objective</b></td><td colspan="4">The objective of the course is to Understand Python Programming Fundamentals, Manipulate and Process Data with Python, Implement Machine Learning Algorithms and Build and Train Neural Networks for AI Applications.</td></tr>
<tr><td><b>Course Outcomes</b></td><td colspan="4">On successful completion of the course the students shall be able to:
CO 1: Apply Python Programming to AI Projects
CO 2: Build and Train Machine Learning Models
CO 3: Develop Deep Learning Models with Neural Networks
CO 4: Deploy AI Solutions and Understand Ethical Implications</td></tr>
<tr><td><b>Course Content:</b></td><td colspan="4"></td></tr>
<tr><td><b>Module 1</b></td><td>Introduction to Python Programming for AI</td><td>Assignment</td><td>Implementation</td><td><b>10 Sessions</b></td></tr>
<tr><td colspan="5">Topics:
Python Basics: Variables, Data Types, Operators, and Control Flow Functions, Loops, and Conditionals statements, Data Structures: Lists, Tuples, Dictionaries, Sets ,Introduction to Libraries: NumPy and Pandas for data manipulation, Basic Input/Output and File Handling
Introduction to Python for AI: Libraries and Frameworks Overview</td></tr>
<tr><td><b>Module 2</b></td><td>Data Processing, Visualization</td><td>Assignment</td><td>Implementation</td><td>10 Sessions</td></tr>
<tr><td colspan="5">Topics:
cleaning and preprocessing with Pandas,Handling missing data, outliers, and duplicates, Data transformation (Normalization, Encoding), Introduction to Matplotlib and Seaborn for Data Visualization, Exploratory Data Analysis (EDA), Visualizing datasets to understand patterns and relationships.</td></tr>
<tr><td><b>Module 3</b></td><td>Introduction to Machine Learning</td><td>Mini - Project</td><td>Implementation</td><td>10 Sessions</td></tr>
<tr><td colspan="5">Topics:
What is Machine Learning? Types of ML algorithms Supervised Learning: Regression, Classification, Unsupervised Learning: Clustering, Key ML Algorithms: Linear Regression, Decision Trees, K-Means ,Introduction to Scikit-learn library
Model evaluation (Accuracy, Precision, Recall, Confusion Matrix)</td></tr>
<tr><td><b>Module 4</b></td><td>Neural Networks and Deep Learning</td><td>Quiz</td><td>Implementation</td><td>10 Sessions</td></tr>
<tr><td colspan="5">Topics:
Introduction to Neural Networks and Deep Learning, Perceptron Model and Backpropagation
Deep Neural Networks and Activation Functions, Introduction to TensorFlow and Keras, Building and Training Neural Networks for Image and Text Classification, Overview of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)</td></tr>
<tr><td colspan="5"><b>Targeted Application & Tools that can be used:</b>
<b>Applications</b>:
1. <b>Data Preprocessing</b>: Clean and manipulate data from various sources such as CSV, Excel, SQL databases, and APIs.
2. <b>Exploratory Data Analysis (EDA)</b>: Gain insights into datasets by identifying trends, patterns, and outliers.
3. <b>Predictive Modeling</b>: Build models for classification (e.g., spam detection) and regression (e.g., house price prediction).
4. <b>Clustering</b>: Group data into clusters for unsupervised learning tasks (e.g., customer segmentation).
5. <b>Model Evaluation</b>: Assess model performance using appropriate metrics such as accuracy,</td></tr>
</table>

precision, recall, and F1-score.

Tools:

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

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

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

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

Text Book(s):

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

Reference(s):

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

Course Code: CSE2265	**Course Title: Essentials of AI Lab**	**L- T-P- C**	0	0	2	1
Version No.	2.0					
Course Prerequisites						
Anti-requisites	**NIL**					
Course Description	This course introduces students to the essential concepts and techniques of Artificial Intelligence (AI) with a focus on practical implementation using Python. Students will explore core AI topics such as search algorithms, knowledge representation, machine learning, and neural networks, while gaining proficiency in using popular Python libraries like NumPy, pandas, scikit-learn, and TensorFlow. Through a series of lab exercises and projects, students will apply AI principles to solve real-world problems, develop intelligent applications, and understand how AI systems function at a foundational level.					

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

Module 1	Introduction to AI and Python for AI	ssignment	nplementation	**8 Sessions**

Lab Assignment 1: Setting Up the Python Environment

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

Lab Assignment 2: Basic Python Programming for AI

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

Lab Assignment 3: Data Exploration and Preprocessing

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

Module 2	Data Processing, Visualization	Assignment	Implementation	8 Sessions

Lab Assignment 1: Data Preprocessing with Pandas

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

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

Lab Assignment 2: Data Aggregation and Grouping with Pandas

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

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

Lab Assignment 3: Data Visualization with Matplotlib and Seaborn

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

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

Lab Assignment 4: Visualizing Relationships and Feature Importance

Objective:
Understand how to visualize relationships between features and evaluate feature importance for predictive models.
Tasks:

1. **Scatter Plot Matrix:**
   - Use **Seaborn's** pairplot() to create a scatter plot matrix to visualize the relationships between multiple features.
   - Analyze the pairwise relationships between features and identify any patterns or correlations.
2. **Heatmap of Correlation Matrix:**
   - Use **Pandas** to calculate the correlation matrix of numeric features.
   - Visualize the correlation matrix using **Seaborn's** heatmap() to understand feature correlations and multicollinearity.
3. **Feature Importance from Models:**
   - Train a decision tree or random forest model using **scikit-learn** on a dataset (e.g., **Iris** or **Titanic**).
   - Visualize feature importance using a bar chart to understand which features have the most impact on the model.
4. **Visualizing Predictions vs. Actual Values:**
   - For regression tasks, visualize the predicted values against the actual values using a scatter plot.
   - For classification tasks, visualize the classification results with a confusion matrix.

Lab Assignment 5: Time Series Data Visualization and Processing

Objective:
Learn how to process and visualize time series data, which is common in AI applications like forecasting and trend analysis.
Tasks:

1. **Load and Preprocess Time Series Data:**
    - Load a time series dataset (e.g., stock market data, weather data).
    - Parse dates properly and set the date column as the index using pd.to_datetime() and .set_index().
2. **Plot Time Series Data:**
    - Plot a time series line chart using **Matplotlib** to visualize trends over time.
    - Create rolling averages (e.g., 7-day, 30-day) to smooth out short-term fluctuations in the time series data.
3. **Seasonal Decomposition of Time Series:**
    - Use **statsmodels** to decompose a time series into seasonal, trend, and residual components.
    - Visualize the decomposed components to understand seasonal variations.
4. **Forecasting with Simple Models:**
    - Use simple forecasting models (e.g., moving average, ARIMA) to predict future values.
    - Visualize the forecasted data along with actual historical data.

Module 3	Introduction to Machine Learning	Assignments	Implementation	8 Sessions

Lab Assignment 3: Implementing Linear Regression

- **Tasks:**
    1. Load a real-world dataset (e.g., **Boston Housing Price** dataset).
    2. Train a **Linear Regression** model using LinearRegression() from scikit-learn.
    3. Evaluate the model using **Mean Squared Error (MSE)** and **R-squared Score**.
    4. Visualize the regression line using Matplotlib.

Lab Assignment 4: Logistic Regression for Classification

- **Tasks:**
    1. Load the **Iris** or **Breast Cancer** dataset.
    2. Preprocess the dataset (handle missing values, encode categorical variables, scale data).
    3. Train a **Logistic Regression** model using LogisticRegression().
    4. Evaluate performance using **Accuracy, Precision, Recall, F1-score**.
    5. Plot the **Confusion Matrix** and **ROC Curve**.

Lab Assignment 5: Implementing K-Nearest Neighbors (KNN)

- **Tasks:**

1. Load the **Iris dataset** and split it into training and testing sets.
2. Train a **KNN classifier** using KNeighborsClassifier().
3. Experiment with different values of **K** and evaluate performance.
4. Visualize decision boundaries using a **scatter plot**.

Lab Assignment 6: Decision Trees and Random Forests

- **Tasks:**
  1. Train a **Decision Tree classifier** on the Titanic dataset.
  2. Visualize the tree structure using plot_tree().
  3. Train a **Random Forest classifier** and compare performance with the decision tree.
  4. Determine the **feature importance** using feature_importances_.

Module 4	Neural Networks and Deep Learning	Quiz	Implementation	6 Sessions

Lab Assignment 7: Introduction to Perceptron and Activation Functions

Tasks:

1. Implement a **single-layer perceptron** using NumPy.
2. Train the perceptron to classify **AND, OR, XOR** gates.
3. Experiment with different **activation functions** (Sigmoid, ReLU, Tanh).
4. Visualize decision boundaries.

Lab Assignment 8: Building a Simple Neural Network with Keras

Tasks:

1. Load the **MNIST dataset** from keras.datasets.
2. Preprocess the data (normalize pixel values, reshape input).
3. Create a **fully connected neural network** using Sequential API.
4. Train and evaluate the model using **categorical cross-entropy loss** and **accuracy**.

Lab Assignment 9: Implementing CNN from Scratch

Tasks:

1. Load the **CIFAR-10 dataset**.
2. Build a CNN with **Conv2D, MaxPooling2D, Flatten, Dense, Dropout** layers.
3. Use **Adam optimizer** and **categorical cross-entropy loss**.
4. Train and visualize loss/accuracy curves.

Lab Assignment 10: Image Augmentation & Regularization

Tasks:

1. Apply **data augmentation** (rotation, zoom, flipping) using ImageDataGenerator.
2. Add **dropout and batch normalization** to prevent overfitting.
3. Compare model performance with and without augmentation.

Lab Assignment 11: Transfer Learning with Pre-trained Models

Tasks:

1. Use **VGG16 or ResNet50** pre-trained on ImageNet.
2. Replace the output layer to classify **new images**.
3. Freeze earlier layers and fine-tune deeper layers.
4. Evaluate the model on a custom **dataset (e.g., Cats vs. Dogs)**.

Lab Assignment 12: Implementing RNN for Text Classification

Tasks:

1. Load **IMDB movie reviews dataset** from keras.datasets.
2. Preprocess text (tokenization, padding sequences).
3. Build an **RNN** with **Embedding, SimpleRNN, Dense** layers.
4. Train and evaluate the model.

Lab Assignment 13: Building an LSTM for Time Series Prediction

Tasks:

1. Load a **time series dataset** (e.g., stock prices, temperature data).
2. Preprocess the data (normalize, reshape).
3. Build an **LSTM-based model**.
4. Predict future values and visualize trends.

Targeted Application & Tools that can be used:

Applications:

1. **Data Preprocessing**: Clean and manipulate data from various sources such as CSV, Excel, SQL databases, and APIs.

- **Exploratory Data Analysis (EDA)**: Gain insights into datasets by identifying trends, patterns, and outliers.
- **Predictive Modeling**: Build models for classification (e.g., spam detection) and regression (e.g., house price prediction).
- **Clustering**: Group data into clusters for unsupervised learning tasks (e.g., customer segmentation).
- **Model Evaluation**: Assess model performance using appropriate metrics such as accuracy, precision, recall, and F1-score.

Tools:

- **Pandas**: For data manipulation and cleaning (e.g., handling missing values, merging datasets).
- **NumPy**: For numerical operations and working with arrays and matrices.

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

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

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

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

Text Book(s):

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

Reference(s):

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

Course Code: CBC2504	**Block Chain Architecture Design**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	CBC 1701 Introduction to Block Chain Platforms					
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 Objectives	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 Out Comes	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 – Analyze the architectural differences and scalability approaches of leading blockchain platforms like 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**		**Understand**	**No. of Sessions: 10**
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**		**Analyse**	**No. of Sessions: 12**

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**			**No. of Sessions: 11**

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**			**No. of Sessions: 12**

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.

Textbook(s):

1. Daniel Drescher, "Blockchain Basics: A Non-Technical Introduction in 25 Steps", Apress, 2017.

2. 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.

Weblinks

IBM Blockchain Architecture Overview

📎 https://www.ibm.com/blockchain/architecture

Ethereum Whitepaper (Blockchain Design Insights)

📎 https://ethereum.org/en/whitepaper/

Hyperledger Fabric Documentation (Blockchain Architecture Focus)
📎 https://hyperledger-fabric.readthedocs.io/en/release-2.2/

Blockchain Technology Overview – NIST Report
📎 https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

Mastering Blockchain GitHub Resources by Imran Bashir
📎 https://github.com/PacktPublishing/Mastering-Blockchain-Third-Edition

Blockchain Hub – Educational Resources on Blockchain
📎 https://blockchainhub.net/

CoinDesk – Blockchain 101 Guides
📎 https://www.coindesk.com/learn/blockchain-101

Course Code: **CDV2503**	**Software Testing**		**L-T-P-C**	3-0-0-3
Version No.	1.0			
Course Pre-requisites	CDV2000 DevOps Foundations			
Anti-requisites	NIL			
Course Description	This course introduces students to fundamental principles and practices of software testing. It emphasizes various testing techniques including black-box and white-box testing, test case design, test automation, debugging, and reliability assessment. The course provides insights into the role of testing in software quality assurance and equips students to apply systematic testing strategies in real-world software projects.			
Course Objectives	To understand the importance of software testing in the software development lifecycle. To learn various testing techniques and their applications in software quality assurance. To analyze testing outcomes and debugging strategies. To explore tools and methodologies for automated testing and test management.			
Course Out Comes	CO1: Understand various software testing levels, techniques, and models used in industry. (Understand Level) CO2: Analyze software testing strategies and defect tracking mechanisms. (Analyze Level) CO3: Apply white-box and black-box testing techniques to evaluate software functionality and performance. (Apply Level) CO4: Implement and manage automated test scripts using industry tools. (Apply Level)			
Module 1	**Fundamentals of Software Testing**		**Understand**	**No. of Sessions: 11**
Software Development Life Cycle, Role of Testing in SDLC, Verification vs Validation, Static vs Dynamic Testing, Levels of Testing, Types of Testing – Unit, Integration, System, Acceptance, Regression.				
Module 2	**Test Case Design Techniques and Defect Management**		**Analyse**	**No. of Sessions: 11**
Black Box Testing: Equivalence Class Partitioning, Boundary Value Analysis, Decision				

Tables;
White Box Testing: Statement, Branch, Path, and Condition Coverage; Defect Lifecycle, Bug Tracking Systems, Test Reporting.

Module 3	**Test Management and Automation**		**Apply**	**No. of Sessions: 11**

Test Planning, Test Metrics, Traceability Matrix, Test Environment Setup; Automation Testing Concepts, Selenium, JUnit/NUnit, Script Writing, Continuous Testing in CI/CD.

Module 4	**Advanced Testing and Tools**		**Apply**	**No. of Sessions: 12**

Performance Testing (JMeter), Load Testing, Security Testing (OWASP overview), Mutation Testing, Test Data Generation, Model-Based Testing, Introduction to AI-driven Testing Tools.

Textbooks

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

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

Reference Books

R1: Ron Patton, *Software Testing*, Pearson Education, 2nd Edition, 2005.
R2: Aditya P. Mathur, *Foundations of Software Testing*, Pearson Education, 2008.
R3: Ilene Burnstein, *Practical Software Testing*, Springer, 2003.
R4: Boris Beizer, *Software Testing Techniques*, Dreamtech Press, 2nd Edition, 2003.

Course Code: CDV2503	**Software Testing Lab**	**L-T-P-C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	CDV 2000					
Anti-requisites	NIL					

Course Description	This course provides practical exposure to various software testing techniques and tools used in industry. It enables students to design, develop, and execute test cases for functional and non-functional software systems. The lab sessions emphasize both manual and automated testing environments using open-source tools and frameworks.
Course Objectives	To introduce the fundamental concepts of software testing life cycle and test management. To provide hands-on training in writing effective test cases and executing them on various software systems. To expose students to automated testing tools like Selenium, JUnit, and bug-tracking tools. To develop skills in identifying, documenting, and reporting software defects and ensuring software quality.
Course Out Comes	Upon successful completion of this course, the students will be able to: CO1: Understand various software testing techniques, strategies, and levels of testing. CO2: Apply manual and automated testing tools to evaluate the functionality and performance of software applications. CO3: Analyze test outcomes and identify bugs using defect tracking systems. CO4: Design and implement comprehensive test plans and interpret results to improve software quality.

List of Tools:

1. Manual Testing Tools

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

🧱 2. Unit Testing Tools

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

🌐 3. Automation Testing Tools

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

📱 4. Mobile Application Testing

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

🔄 5. Integration / CI Tools

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

🔍 6. Performance and Load Testing Tools

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

✅ 7. Code Coverage Tools

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

🖌 8. Code Quality & Static Analysis

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

🌐 9. Cross-Browser Testing

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

List of Experiments

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

13. **Performance testing using Apache JMeter**

14. **Preparing test plans, test suites, and test summary reports**

15. **Mini-project: Complete test life cycle on a web/mobile application including documentation**

Textbooks

T1: William Stallings, Cryptography and Network Security: Principles and Practice, Pearson, 7th Edition, 2017.

T2: Behrouz A. Forouzan, Cryptography and Network Security, McGraw-Hill Education, 2nd Edition, 2011.

Reference Books

R1: Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 2nd Edition, 1996.

R2: Douglas R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 4th Edition, 2018.

R3: Charlie Kaufman, Radia Perlman, and Mike Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall, 2nd Edition, 2002.

R4: Christof Paar, Jan Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer, 2010.

Web Resources

W1: https://cryptography.io/ – Python cryptographic library and resources.

W2: https://nvlpubs.nist.gov/ – NIST publications on cryptographic standards.

W3: https://www.owasp.org/ – Open Web Application Security Project for best practices.

W4: https://www.tutorialspoint.com/cryptography/index.htm – Cryptography basics and tutorials.

W5: https://www.coursera.org/learn/crypto – Stanford's free online cryptography course on Coursera.

Course Code: **CSE2508**	**Course Title:** Mobile Applications and Development **Type of Course:** Theory	**L- T-P-C**	2	**0**	0	**2**

Version No.	2.0
Course Pre-requisites	CSE3514 Object Oriented Programming Using Java
Anti-requisites	**NIL**
Course Description	The course deals with the basics of android platform and application life cycle. The goal of the course is to develop mobile applications with Android containing at least one of the following phone material components: GPS, accelerometer or phone camera, use simple GUI applications and work with database to store data locally or in a server. Topics include user interface design; user interface building; input methods; data handling; network techniques and URL loading; GPS and motion sensing. Android application framework and deployment. Power management, Screen resolution, Touch interface, Store data on the device.
Course Objective	The objective of the course is to familiarize the learners with the concepts of Mobile Applications and Development as mentioned above and attain Employability Skills through Experiential Learning Techniques.
Course Outcomes	On successful completion of the course the students shall be able to: 1. Discuss the fundamentals of mobile application development and its architecture. (Comprehension) 2. Illustrate mobile applications with appropriate android view. (Application) 3. Demonstrate the use of services, broadcast receiver, Notifications and content provider.(Application) 4. Apply data persistence techniques, to perform CRUD operations. (Application) 5. Use advanced concepts for mobile application development. (Application)
Course Content:	

Module 1	Introduction and Architecture of Android	Assignment	Simulation/Data Analysis	5 Sessions
Topics: Android: History and features, Architecture, Development Tools, Android Debug Bridge (ADB), and Life cycle.				
Module 2	User Interfaces, Intent and Fragments	Term paper/Assignment	Simulation/Data Analysis	6 Sessions
Topics: Views, Layout, Menu, Intent and Fragments.				

<table>
<tr><td><b>Module 3</b></td><td>Components of Android</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>6 Sessions</td></tr>
<tr><td colspan="5">Topics:
Activities, Services, Broadcast receivers, Content providers, User Navigation</td></tr>
<tr><td><b>Module 4</b></td><td>Notifications and Data Persistence</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>6 Sessions</td></tr>
<tr><td colspan="5">Topics:
Notification, Shared Preferences, SQLite database, Android Room with a View, Firebase.</td></tr>
<tr><td><b>Module 5</b></td><td>Advance App Development</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>7 Sessions</td></tr>
<tr><td colspan="5">Topics:
Graphics and Animation, App Widgets, Sensors, Performance, Location, Places, Mapping, Custom Views, Canvas.</td></tr>
<tr><td colspan="5"><b>Targeted Application & Tools that can be used:</b>
Applications:
Native Android Applications
Native iOS Applications
Cross Platform mobile Apps
Mobile web Applications</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td colspan="5"><b>Text Book(s):</b>
T1. Pradeep kothari “Android Application Development - Black Book”, dreamtechpress
T2. Barry Burd (Author), “Android Application Development” ALL – IN – ONE FOR Dummies
T3. Jeff Mcherter (Author),Scott Gowell (Author), “Professional mobile Application Development” paperback, Wrox - Wiley India Private Limited
T4. Wei-Meng Lee (Author) “Beginning Android Application Development” Wrox – Wiley India Private Limited</td></tr>
<tr><td colspan="5"><b>Reference(s):</b>
1. Bill Phillips, Chris Stewart, and Kristin Marsicano (Author) “Android Programming” 3rd edition,
2017.The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 5. The Big Nerd Ranch Guide, by”
2. Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd,
2014.
3. Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O’Reilly SPD
Publishers, 2015.
4. J F DiMarzio, “Beginning Android Programming with Android Studio”, 4th Edition, Wiley India Pvt
Ltd, 2016. ISBN-13: 978-8126565580</td></tr>
</table>

5. Anubhav Pradhan, Anil V Deshpande, “ Composing Mobile Apps” using Android, Wiley 2014,
ISBN: 978-81-265-4660-2
6. Reto Meier “Professional Android Application Development”
E-Resources: https://puniversity.informaticsglobal.com/login Or http://182.72.188.193/

Course Code: CBC2503 | Course Title: Distributed Ledger Technology Laboratory | L:T:P:C = 0:0:2:1

| Version 1.0 |

Course Pre-requisites:
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 Objectives:
1. To provide practical understanding of blockchain platforms like Ethereum and Hyperledger.
2. To develop and test smart contracts using Solidity and Chaincode.
3. To simulate peer-to-peer blockchain networks and transaction flows.
4. To integrate DLT with applications for real-world solutions.

Course Outcomes:
CO1: Understand the setup and configuration of blockchain environments.
CO2: Apply skills to write, deploy, and test smart contracts on Ethereum.
CO3: Implement and manage blockchain networks using Hyperledger Fabric.
CO4: Apply integration and debugging techniques to test and deploy decentralized applications.

Lab Experiments (15 Weeks):

1. Installation and setup of blockchain development environment
   (Tools: Node.js, Ganache, MetaMask, Remix IDE)

2. Creating a local Ethereum blockchain using Ganache CLI
   (Tool: Ganache CLI)

3. Creating Ethereum wallets and sending Ether between accounts
   (Tools: MetaMask, Web3.js)

4. Writing and deploying a basic smart contract in Solidity
   (Tool: Remix IDE)

5. Solidity programming: control structures, data types, modifiers
   (Tool: Remix IDE)

6. Deploying smart contracts on Ganache and interacting with Web3.js
   (Tools: Ganache, Web3.js, Node.js)

7. Smart contract functions: Events, state variables, and mappings
   (Tool: Remix IDE)

8. Testing smart contracts using Truffle framework
   (Tools: Truffle, Mocha/Chai for test cases)

9. Setup of a Hyperledger Fabric blockchain network using Docker
   (Tools: Hyperledger Fabric, Docker)

10. Writing and deploying Chaincode (smart contracts) on Fabric
    (Languages: Go or Node.js)

11. Invoking transactions and querying the ledger in Fabric
    (Tools: Fabric CLI or SDK)

12. Developing a basic asset transfer DApp on Ethereum
    (Tools: Solidity, Web3.js, HTML/JS frontend)

13. Smart contract security: Detecting vulnerabilities (e.g., reentrancy)
    (Tools: MythX, Slither)

14. End-to-end DApp project integration and deployment
    (Tools: Truffle/Hardhat + Web3.js + frontend)

15. Mini Project Demo: Real-world use case using DLT (group work)
    (Students choose Ethereum or Fabric for implementation)

Textbooks:
T1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020
T2: Melanie Swan, *Blockchain: Blueprint for a New Economy*, O’Reilly Media, 2015

Reference Books: 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
Web Resources: W1: https://ethereum.org/en/developers/ W2: https://hyperledger.org/use/fabric W3: https://www.ibm.com/blockchain W4: https://www.blockchain-council.org/ W5: https://developer.algorand.org/ +==+

Course Code: CSE2509	**Course Title:** Mobile Applications and Development Lab **Type of Course:** Lab	**L- T-P-C**	0	**0**	4	**2**
Version No.	2.0					
Course Pre-requisites	CSE1514 Object Oriented Programming using Java					
Anti-requisites	**NIL**					
Course Description	The course provides hands-on experience in designing, developing, and deploying mobile applications for Android and iOS platforms. Students will work with native development frameworks such as Android Studio (Java/Kotlin) and Xcode (Swift), as well as explore cross-platform tools like Flutter or React Native.					
Course Objective	The objective of the course is to develop Native and Cross-Platform Mobile Applications, design Interactive and Responsive User Interfaces, integrate Backend Services and APIs, implement State Management and Performance Optimization, ensure Mobile App Security and Data Protection					
Course Outcomes	On successful completion of the course the students shall be able to: 1. Develop Functional Mobile Applications 2. Design and Implement Interactive UIs 3. Integrate Cloud Services and APIs 4. Integrate Backend Systems and Data Management 5. Deploy, Publish, and Maintain advanced Mobile Application					

<table>
<tr><td><b>Course Content:</b></td><td colspan="4"></td></tr>
<tr><td><b>Module 1</b></td><td>Introduction and Architecture of Android</td><td>Assignment</td><td>Simulation/Data Analysis</td><td>8 Sessions</td></tr>
<tr><td colspan="5">1.a. Design an app to read user inputs using edit text and display the result of arithmetic operations using toast message.
1.b. Create an android app to calculate the current age of yourself, select your DOB using date picker.
2. Design an app to input your personal information. Use an autocomplete text view to select your place of birth.</td></tr>
<tr><td><b>Module 2</b></td><td>User Interfaces, Intent and Fragments</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>13 Sessions</td></tr>
<tr><td colspan="5">3. a. Design an app to select elective course using spinner view and on click of the display button, toast your ID and selected elective course.
3. b. Design a restaurant menu app to print the total amount of orders.</td></tr>
<tr><td><b>Module 3</b></td><td>Components of Android</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>13 Sessions</td></tr>
<tr><td colspan="5">4. Develop an android app that uses intent to maintain the following scenario.
Check the eligibility criteria for voting. Input the Aadhar no., Name & age in the first activity. If the age is above 18, display the voter's detail in the second activity. Else, display, “You are not eligible to vote” in the second Activity.
5. Demonstrate the use of fragment with list of buttons representing various colors, and on click of these buttons, the appropriate color is filled in the next fragment. Create an Android application to input the vitals of a person (temperature, BP). If the vitals are abnormal, give proper notification to the user.
6. Create an android app to for movie ticket booking. Save the user name of the customer using shared preferences. After completion of booking, retrieve the username from the shared preferences and print the ticket details.</td></tr>
<tr><td><b>Module 4</b></td><td>Notifications and Data Persistence</td><td>Term paper/Assignment</td><td>Simulation/Data Analysis</td><td>13 Sessions</td></tr>
<tr><td colspan="5">7. Create an android application to manage the details of students' database using SQLite.Use necessary UI components, which perform the operations such as insertion, modification, removal and view.Presidency University needs an APP for Admission eligibility checking for students, for that you need to take the following information from the Student: registration ID, physics, chemistry and mathematics marks (PCM), fees is allotted as below criteria.
PCM (Total marks %) Fee concession</td></tr>
</table>

90 above 80 %
70 to 89 60 %
Below 69 % no concession
On click on the button "Registration" details should be stored in the database using SQLite.
Create button DISPLAY ALL (full students list) on click on the button it should display the students list per the fee
concession.
8. A company need to design an app that plays soft music automatically in the background. Create an app to achieve this functionality.
9. Create an android application such that your view object in the Activity can be Animated with fade-in effect. Create an appropriate XML file named fade-in and write the application to perform the property animation.

Module 5	Advance App Development	Term paper/Assignment	Simulation/Data Analysis	13 Sessions

10. Demonstrate how to send SMS and email.
11. Create an android application to transfer a file using WiFi. Create an android application "Where am I" with an Activity that uses the GPS Location provider to find the device's last known location.

Targeted Application & Tools that can be used:
Applications:

1. Native Android Applications (Java/Kotlin)

- Android Mobile Apps built for Android smartphones and tablets using Java or Kotlin programming languages.
- Target audience: Android users.

2. Native iOS Applications (Swift)

- iOS Mobile Apps designed for iPhone and iPad using Swift.
- Target audience: iOS users (Apple ecosystem).

3. Cross-Platform Mobile Apps (Flutter, React Native)

- Cross-platform apps designed to run on both Android and iOS from a single codebase using frameworks like Flutter or React Native.
- Target audience: Users on both Android and iOS platforms.

4. Mobile Web Applications (Progressive Web Apps - PWA)

- Mobile-optimized web applications using HTML5, CSS3, and JavaScript that run in a browser with native-like functionality (offline support, push notifications).
- Target audience: Users accessing apps via mobile browsers.

Development Tools and Frameworks

1. Integrated Development Environments (IDEs)

- Android Studio (for Android): The official IDE for Android development, supporting Java, Kotlin, and Android SDK.
- Xcode (for iOS): The official IDE for iOS development with Swift and Objective-C, providing a comprehensive suite of development tools for iPhone/iPad applications.
- Visual Studio Code (VS Code): Lightweight IDE for working with Flutter, React Native, and web development projects.

2. Cross-Platform Development Frameworks

- Flutter: Open-source UI framework by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.
- React Native: Open-source framework developed by Facebook for building cross-platform apps with JavaScript and React.

3. Backend & Cloud Tools

- Firebase: Google's backend-as-a-service (BaaS) platform offering authentication, real-time databases, cloud storage, and push notifications for mobile apps.
- AWS Amplify: Cloud platform for backend services (API, storage, authentication) and mobile deployment.
- SQLite / Realm: Local storage solutions for mobile apps to manage data storage and retrieval on-device.

4. Mobile App Testing and Debugging Tools

- Android Emulator (for Android): A virtual device to run and test Android apps without needing physical devices.
- Xcode Simulator (for iOS): A tool to simulate different iOS devices and test apps during development.
- Appium: Open-source tool for automated testing across native, hybrid, and mobile web applications.

5. Version Control and Collaboration

- Git: Version control system for managing code changes and collaborating with teams.
- GitHub / GitLab / Bitbucket: Online platforms for hosting Git repositories, collaboration, and version control management.

6. Mobile App Deployment Tools

- Google Play Console: For managing Android app publishing, distribution, and monitoring.
- Apple App Store Connect: For managing iOS app submissions, reviews, and releases on the Apple App Store.

7. UI/UX Design Tools

- Figma / Adobe XD: Tools for UI/UX design and wireframing to create the visual elements of mobile applications before development.
- Sketch: Vector-based design tool for iOS UI design and prototyping

Text Book(s):
T1. Pradeep kothari "Android Application Development - Black Book", dreamtechpress
T2. Barry Burd (Author), "Android Application Development" ALL – IN – ONE FOR Dummies
T3. Jeff Mcherter (Author),Scott Gowell (Author), "Professional mobile Application Development" paperback, Wrox - Wiley India Private Limited
T4. Wei-Meng Lee (Author) "Beginning Android Application Development" Wrox – Wiley India Private Limited

Reference(s):
1. Bill Phillips, Chris Stewart, and Kristin Marsicano (Author) "Android Programming" 3rd edition,
2017.The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 5. The Big Nerd Ranch Guide, by"
2. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd,
2014.
3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O'Reilly SPD
Publishers, 2015.
4. J F DiMarzio, "Beginning Android Programming with Android Studio", 4th Edition, Wiley India Pvt
Ltd, 2016. ISBN-13: 978-8126565580
5. Anubhav Pradhan, Anil V Deshpande, " Composing Mobile Apps" using Android, Wiley 2014,
ISBN: 978-81-265-4660-2
6. Reto Meier "Professional Android Application Development"
E-Resources: https://puniversity.informaticsglobal.com/login Or http://182.72.188.193/

5.

Course Code: CBC2509
Course Title: Consensus Algorithms and Network Design
L:T:P:C – 3:0:0:3
Total Hours: 45
Prerequisite: CBC2000 – Foundations of Blockchain Technology

Course Description

This course offers an in-depth exploration of consensus mechanisms and blockchain network design. It focuses on understanding the theoretical and practical aspects of various consensus protocols used in decentralized systems. The course also examines how blockchain networks are structured and how performance, scalability, and security are achieved through architectural design.

Course Objectives

- Analyze the principles and significance of consensus in decentralized systems

- Explore different consensus algorithms and their applications
- Understand network topology, node behavior, and communication models in blockchain networks
- Evaluate the trade-offs between security, scalability, and decentralization
- Design secure and scalable blockchain networks

Course Outcomes

- **CO1 (Understand)**: Describe various consensus algorithms and their theoretical foundations
- **CO2 (Apply)**: Implement and simulate selected consensus mechanisms in a blockchain context
- **CO3 (Analyze)**: Examine the scalability, performance, and security aspects of blockchain networks
- **CO4 (Design)**: Design and configure basic blockchain network architectures

Course Content (Total: 45 Hours)

Module 1: Introduction to Consensus in Blockchain (Understand) (9 Sessions)

Role of consensus in decentralized systems, Requirements: safety, liveness, fault tolerance, Classification: permissioned vs permissionless networks, Overview of consistency models

Module 2: Classical and Proof-Based Consensus Algorithms (Apply) (12 Sessions)

Proof-of-Work (PoW): Bitcoin model, Proof-of-Stake (PoS): Variants and Ethereum 2.0, Delegated Proof-of-Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Federated Consensus (e.g., Stellar Consensus Protocol), Hybrid consensus models

Module 3: Advanced Consensus Mechanisms (Analyze) (12 Sessions)

Proof-of-Authority (PoA), Raft and Paxos algorithms, Directed Acyclic Graph (DAG)-based consensus (e.g., IOTA, Hedera), Sharding and Layer 2 solutions (e.g., Optimistic Rollups, ZK-Rollups), Consensus in consortium and private blockchains

Module 4: Blockchain Network Design (Design) (12 Sessions)

Node types and roles: full nodes, light nodes, miners, validators, Peer-to-peer (P2P) network architecture, Network topology and message propagation, Latency, throughput, and fault tolerance, Case studies: Ethereum, Hyperledger Fabric, and Tendermint

Textbooks

- **T1**: Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020
- **T2**: Arvind Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press

Reference Books

- **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)*

Web Resources

- [W1] https://ethereum.org
- [W2] https://bitcoin.org

- [W3] https://hyperledger.org
- [W4] https://stellar.org

Course Code: CBC2501	**Course Title: Smart Contract and Solidity Lab** **Type of Course: LAB**	**L- T-P- C**	0-0-2-1	
Version No.	1			
Course Pre-requisites	CBC2000			
Anti-requisites	NONE			
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	The objective of the course is to familiarize the learners with the concepts of **Smart Contract andSolidity** and attain **EMPLOYABILITY** through **Experiential Learning Techniques.**.			
Course Out Comes	**On successful completion of the course the students shall be able to:** **CO 1 :**Understand the fundamentals of computational Element of the Blockchain Technology **C.O 2:** Implementuser-defined operations of arbitrary complexity that are not possible through plaincryptocurrency protocols **C.O 3:** Exhibitbest practices for designing solutions with smart contracts using Solidity and Remix IDE			
Module 1	Introduction to Smart Contract	TEST-1	Fundaments of Smart Contractand Solidity	**12Sessions**
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	TEST-1	Case studies / Case let	**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	Endterm lab Exam	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

List of 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**

🛠 **Open-Source Tools Overview**

- **CrypTool – Educational tool for learning cryptography visually**
- **Python (PyCryptodome, hashlib, cryptography) – For programmatic**

encryption/decryption

- **OpenSSL – Command-line utility for SSL, AES, RSA operations**
- **GnuPG – For creating and verifying digital signatures and PGP**
- **Wireshark – Packet sniffer for analyzing network-level security**

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

Reference Books:
R1: Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 2nd Edition, 1996
R2: Douglas R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 4th Edition, 2018
R3: Charlie Kaufman 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

Web Resources:
W1: https://cryptography.io
W2: https://nvlpubs.nist.gov
W3: https://www.owasp.org
W4: https://www.tutorialspoint.com/cryptography/index.htm
W5: https://www.coursera.org/learn/crypto

Course Code: CSE2266	**Course Title: Theory of Computation** **Type of Course: Theory Only**	**L- T-P- C**	3	0	0	3
Version No.	2.0					
Course Pre-requisites	NIL					
Anti-requisites	Nil					
Course Description	The course deals with introduction of formal languages and the correspondence between language classes and the automata that recognize them. Topics include: Formal definitions of grammars and acceptors, Deterministic and Nondeterministic systems, Grammar ambiguity, finite state and push-down automata; normal forms; Turing machines and its relations with algorithms.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Theory of Computation as mentioned above** and attain **Skill Development** through **Problem Solving** Methodologies.					

Course Out Comes	On successful completion of the course the students shall be able to: 1. Describe various components of Automata. (Knowledge) 2. Illustrate Finite Automata for the given Language. (Application) 3. Distinguish between Regular grammar and Context free grammar. (Comprehension) 4. Construct Push down Automata. (Application) 5. Construct Turing machine for a Language. (Application)			
Course Content:				
Module 1	**Introduction to automata theory**	Assignment	Problems on Strings and Language operations	**06 Sessions**
Topics: Introduction to Automata Theory, Applications of Automata Theory, Alphabets, Strings, Languages & operations on languages, Representation of automata, Language recognizers, Finite State Machines (FSM): Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs				
Module 2	**Finite Automata**	Assignment	Problems on DFA, NFA's	**13 Sessions**
Topics: Basic concepts of Finite automata, DFA- definitions of DFA, Deterministic Accepters Transition Graphs and Languages and DFA's, Regular Languages, NFA- Definition of a Nondeterministic Accepter, Languages and NFA's Why Non-determinism? Equivalence of Deterministic and Nondeterministic Finite Accepters, Reduction of the Number of States in Finite Automata.				
Module 3	**Regular Expressions & Context Free Grammar**	Assignment	Problems on RE, CFG, PT, PL and Ambiguity	**12 Sessions**
Topics: Formal Definition of a Regular Expression, Languages Associated with Regular Expressions, Languages, Regular Languages (RL) and Non-regular Languages: Closure properties of RLs, to show some languages are not RLs, Closure Properties of Regular Context Free Grammars-Examples of Context-Free Languages, Leftmost and Rightmost Derivations, Derivation Trees, Relation Between Sentential Forms and Derivation Trees, Ambiguity in Grammars and Languages: Ambiguous Grammars, Removing Ambiguity, Chomsky Normal Form, Gribiche Normal Form.				
Module 4	**Push down Automata**	Assignment	Problems on pushdown Automaton	**08 Sessions**
Topics: Definition of a Pushdown Automaton, Language Accepted by a Pushdown Automaton, Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack Equivalence of PDA's and CFG's: From Grammars to Pushdown Automata.				
Module 5	**Turing Machine**	Assignment	Problems on Turning Machine	**07 Sessions**
Topics: Definition of a Turing Machine, Turing Machines as Language Accepters, Example Languages to construct Turing machine, Turing Machines as Transducers, Halting Programming Techniques for Turing Machines				

Targeted Application & Tools that can be used:

Targeted Application:

1. Text Processing
2. Compilers
3. Text Editors
4. Robotics Applications
5. Artificial Intelligence

Tools:

1. JFLAP (Java Formal Language and Automata Package) Software simulation tool. It's interactive educational software written in Java to experiment topics in automata theory.
2. Turing machine Online simulators.

Text Book

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

References

1. Aho, Ullman and Hopcroft, "Theory of Computation", Pearson India 3rd Edition 2008.
2. Michael Sipser, "Theory of Computation", Cengage India 3rd Ed, 2014.

E-Resources

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

Topics relevant to "SKILL DEVELOPMENT": Deterministic and Non-Deterministic Automaton, Regular Expressions, CFGs, Turning Machine and Pushdown automaton for Skill Development through Problem Solving methodologies. This is attained through assessment component mentioned in course handout.

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 Hours**		
	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 Hours**		
	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: 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: CO.1 Recognize the importance of Value Education through the process of self-exploration CO.2 Explain the human being as the co-existence of the self and the body in harmony.					

	CO.3 Describe the role of foundational values in building harmonious relationships. CO.4 Summarize the importance of a holistic perspective in developing ethical professional behavior.			
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

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

Reference Books

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

Resources:

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

Topics relevant to Skill Development:

1. An attitude of enquiry.
2. Write reports

The topics related to Human values and Professional ethics:

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

Course Code: CSE7000	**Course Title: Internship** **Type of Course:**	**L- T-P-C**	-	-	-	2
Version No.	**1.0**					

Course Pre-requisites	**Knowledge and Skills related to all the courses studied in previous semesters.**
Anti-requisites	**NIL**
Course Description	Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems.
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learning techniques.
Course Outcomes	On successful completion of this course the students shall be able to: 1. Identify the engineering problems related to local, regional, national or global needs. (Understand) 2. Apply appropriate techniques or modern tools for solving the intended problem. (Apply) 3. Design the experiments as per the standards and specifications. (Analyze) 4. Interpret the events and results for meaningful conclusions. (Evaluate)

Semester: 6

Course Code: CBC2507 **Course Title**: Token Standards and Creation **L:T:P:C** – 3:0:0:3 **Total Hours**: 45 **Prerequisite**: CBC2000 – Foundations of Blockchain Technology
Course Description

This course provides a comprehensive understanding of tokenization in blockchain systems. It covers various token standards (like ERC and BEP series), token creation processes, smart contract deployment, and real-world applications such as NFTs and utility tokens. The course emphasizes hands-on learning by building and deploying tokens on blockchain networks.

Course Objectives

- Understand the principles and need for tokenization in blockchain systems
- Explore various token standards across public blockchains
- Learn to create and deploy tokens using smart contracts
- Analyze use-cases of fungible and non-fungible tokens in real-world ecosystems

Course Outcomes

- **CO1 (Understand)**: Explain the concept of tokenization and differentiate between fungible and non-fungible tokens
- **CO2 (Analyze)**: Compare various token standards and evaluate their use cases
- **CO3 (Apply)**: Develop and deploy ERC-20 and BEP-20 tokens on Ethereum and BNB chains
- **CO4 (Apply)**: Create and deploy NFTs using ERC-721/1155 and demonstrate their usage in real-world applications

Course Content (Total: 45 Hours)

Module 1: Introduction to Tokenization (10 Sessions)

Definition and types of tokens (utility, security, governance), Fungible vs Non-Fungible Tokens (NFTs), Role of tokens in DeFi and Web3, Token economics (tokenomics), Token lifecycle and distribution models

Module 2: Token Standards Overview (12 Sessions)

Ethereum Token Standards: ERC-20, ERC-721, ERC-777, ERC-1155, Binance Smart Chain Token Standards: BEP-2, BEP-20, Comparison of standards, use cases in games, metaverse, finance

Module 3: Smart Contract-Based Token Creation (11 Sessions)

Writing smart contracts for ERC-20/ERC-721 using Solidity, Contract structure and metadata, Minting, burning, transferring, access control, Testing, deploying tokens using Remix, Truffle, and Hardhat

Module 4: Token Integration and Case Studies (12 Sessions)

Wallets and token interaction (MetaMask, Trust Wallet), Interfacing tokens in DApps, NFT marketplaces and royalties, Security considerations in token contracts, Case studies: Uniswap token, Bored Ape NFTs, Chainlink, Axie Infinity

Textbooks

- **T1**: Andreas M. Antonopoulos & Gavin Wood, *Mastering Ethereum*, O'Reilly Media
- **T2**: Ritesh Modi, *Solidity Programming Essentials*, Packt Publishing

Reference Books

- **R1**: Imran Bashir, *Mastering Blockchain*, Packt Publishing
- **R2**: Salman A. Baset & Matthew E. Taylor, *Hands-On Blockchain for Ethereum*, Packt
- **R3**: Smart Contract official documentation (Solidity docs)

Web Resources

- [W1] https://ethereum.org
- [W2] https://soliditylang.org
- [W3] https://remix.ethereum.org
- [W4] https://opensea.io/blog/guides/what-are-nfts

Course Code: CBC2508
Course Title: Token Standards and Creation Lab
L:T:P:C – 0:0:2:1
Duration: 15 Weeks (1 Lab per Week)
Prerequisite: CBC2000 – Foundations of Blockchain Technology, CBC2507 – Token Standards and Creation

Course Objective

To provide hands-on experience in developing, deploying, and managing blockchain-based tokens (fungible and non-fungible) using industry-standard tools and smart contracts.

Course Outcomes

- **CO1 (Understand)**: Demonstrate understanding of token creation standards and smart contract structures
- **CO2 (Apply)**: Develop ERC-20 and BEP-20 fungible tokens using Solidity
- **CO3 (Apply)**: Create and deploy NFTs using ERC-721 and ERC-1155 standards
- **CO4 (Analyze)**: Integrate deployed tokens with wallets and DApps, and test for real-world interaction

Lab Experiments (15 Weeks)

Week	**Title of Experiment**
1	Introduction to Remix IDE and MetaMask Wallet setup
2	Create and deploy a simple smart contract with basic state variables (Solidity)
3	Develop a basic ERC-20 Token (Token name, symbol, supply, and decimals)
4	Implement transfer and approve functions for ERC-20 Token
5	Add Mint and Burn functionality to ERC-20 Token
6	Deploy ERC-20 Token on a testnet (e.g., Goerli or BSC Testnet)
7	Create a basic BEP-20 Token and deploy on Binance Smart Chain Testnet
8	Develop and deploy a basic ERC-721 (NFT) smart contract
9	Add metadata (image, name, description) to an ERC-721 token
10	Implement batch minting using ERC-1155 standard
11	Integrate tokens with MetaMask and simulate transfers
12	Test token interaction in a sample frontend DApp using Web3.js or Ethers.js
13	Explore and interact with token marketplaces (e.g., OpenSea Testnet)
14	Conduct security audit basics: test for reentrancy, overflow, and access control flaws in token contracts
15	Final Demo: Students present their own token project with code, deployment, and interaction flow

Lab Tools and Platforms

- **Smart Contract IDE**: Remix Ethereum, Visual Studio Code
- **Languages**: Solidity
- **Test Networks**: Goerli, Sepolia (Ethereum), BSC Testnet
- **Wallets**: MetaMask
- **Frameworks**: Hardhat / Truffle
- **Optional Tools**: Pinata/IPFS for NFT metadata, OpenSea Testnet, Alchemy, Infura

Recent Reference Books

- Imran Bashir, *Mastering Blockchain*, Packt Publishing, 4th Edition, 2023
- Andreas M. Antonopoulos, *Mastering Ethereum*, O'Reilly Media, 2018
- Ritesh Modi, *Solidity Programming Essentials*, Packt Publishing, 2nd Edition, 2022
- Narayan Prusty, *Building Blockchain Projects*, Packt Publishing, 2021

Web Resources with Links

- Hardhat Development Framework
- Alchemy Blockchain Infrastructure
- Infura – Ethereum API Suite

Course Code: CBC2504
Course Title: Blockchain Security and Performance
L:T:P:C – 3:0:0:3
Preqrequisite: CBC1700

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 Objectives

— 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

CO1 (Understand): Identify and describe security challenges in blockchain platforms
CO2 (Analyze): Evaluate consensus attacks and performance issues in different blockchain frameworks
CO3 (Apply): Implement security-aware smart contracts and mitigation techniques
CO4 (Apply): Analyze and optimize blockchain performance under diverse network loads

Course Content (45 Hours Total)

Module 1: Blockchain Security Foundations – 11 Sessions

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 – 11 Sessions
Ethereum vulnerabilities, DAO attack analysis, Reentrancy, Arithmetic overflow/underflow, Front-running, Best practices in secure smart contract design, Tools for smart contract auditing

Module 3: Blockchain Performance Metrics – 11 Sessions
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 – 12 Sessions
Layer-2 scaling (State Channels, Rollups), Sharding, DAG-based architectures, ZKPs, Mixers and Privacy coins, Secure multi-party computation, Formal verification of performance

Textbooks
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

Reference Books
R1: Antonopoulos & Wood, *Mastering Ethereum*, O'Reilly Media
R2: Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press
R3: Koshy & Arvind, *Blockchain Security and Performance*, Springer, 2021
R4: Bonneau et al., *Security Analysis of Blockchain Protocols*, ACM Reviews

Web Resources
W1: https://consensys.net
W2: https://ethereum.org/en/developers/docs/security/
W3: https://chainsecurity.com/
W4: https://cryptozombies.io
W5: https://bitcoin.org/en/security

Course Code: CBC2505
Course Title: Blockchain Security and Performance Lab
L:T:P:C – 0-0-2-1
Prerequisite: CBC2000

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 Objectives
— 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
CO1 (Understand): Identify and describe security challenges in blockchain platforms
CO2 (Analyze): Evaluate consensus attacks and performance issues in different blockchain frameworks
CO3 (Apply): Implement security-aware smart contracts and mitigation techniques
CO4 (Apply): Analyze and optimize blockchain performance under diverse network loads

Course Content (45 Hours Total)
Module 1: Blockchain Security Foundations – 11 Sessions
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 – 11 Sessions
Ethereum vulnerabilities, DAO attack analysis, Reentrancy, Arithmetic overflow/underflow, Front-running, Best practices in secure smart contract design, Tools for smart contract auditing

Module 3: Blockchain Performance Metrics – 11 Sessions
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 – 12 Sessions
Layer-2 scaling (State Channels, Rollups), Sharding, DAG-based architectures, ZKPs, Mixers and Privacy coins, Secure multi-party computation, Formal verification of performance

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

Textbooks

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

Reference Books

R1: Antonopoulos & Wood, *Mastering Ethereum*, O'Reilly Media
R2: Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press
R3: Koshy & Arvind, *Blockchain Security and Performance*, Springer, 2021
R4: Bonneau et al., *Security Analysis of Blockchain Protocols*, ACM Reviews

Web Resources

W1: https://consensys.net
W2: https://ethereum.org/en/developers/docs/security/
W3: https://chainsecurity.com/
W4: https://cryptozombies.io
W5: https://bitcoin.org/en/security

Course Code: CBC2506	**BlockChain Architecture Design**	**L-T-P-C**	3	0	0	3

Version No.	1.0
Course Pre-requisites	CBC 2001 Introduction to Block Chain Platforms
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 Objectives	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 Out Comes	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 – Analyze the architectural differences and scalability approaches of leading blockchain platforms like 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**		**Understand**	**No. of Sessions: 10**
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**		**Analyse**	**No. of Sessions: 12**
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**			**No. of Sessions: 11**
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**			**No. of Sessions: 12**
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.				
Textbook(s): Daniel Drescher, "Blockchain Basics: A Non-Technical Introduction in 25 Steps", Apress, 2017. 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. **Weblinks** IBM Blockchain Architecture Overview 🖇 https://www.ibm.com/blockchain/architecture Ethereum Whitepaper (Blockchain Design Insights) 🖇 https://ethereum.org/en/whitepaper/ Hyperledger Fabric Documentation (Blockchain Architecture Focus)				

📎 https://hyperledger-fabric.readthedocs.io/en/release-2.2/

Blockchain Technology Overview – NIST Report
📎 https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

Mastering Blockchain GitHub Resources by Imran Bashir
📎 https://github.com/PacktPublishing/Mastering-Blockchain-Third-Edition

Blockchain Hub – Educational Resources on Blockchain
📎 https://blockchainhub.net/

CoinDesk – Blockchain 101 Guides
📎 https://www.coindesk.com/learn/blockchain-101

Course Code: **CSE2274**	**Course Title:** Competitive Programming and Problem Solving	L-T-P-C	0	0	4	2
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	**NIL**					
Course Description	The **Competitive Programming and Problem Solving** course equips students with efficient problem-solving skills for coding competitions and real-world challenges. Starting with brute-force solutions, students learn to optimize time and space complexity using advanced techniques like dynamic programming, greedy algorithms, and backtracking. Hands-on practice on platforms like CodeChef and Codeforces helps tackle problems involving number theory, data structures, and algorithmic paradigms. By understanding CP constraints and fostering a strategic mindset, students gain the confidence to excel in competitions, technical interviews, and practical applications.					
Course Out Comes	On successful completion of the course the students shall be able to: CO1 : Understanding the issues of online platforms and Competitive Programming (CP) and developing brute force coding for commonly asked CP problems.					

	CO2 : Analyzing the space and time complexity of brute force solutions and designing efficient solutions. CO3 : Evaluating the applicability of suitable algorithmic approaches to solve relevant CP problems. CO4: Creating efficient solutions of CP problems using the learnt algorithmic approaches.
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Competitive Programming and Problem Solving** and attain **Skill Development** through **Experiential Learning** techniques.

Module 1: Introduction to Competitive Programming

Overview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforces etc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL

Module 2: Number Theory for Problem-Solving

Use of Number Theory for problem-solving: reducing time/space complexity of brute force coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient coding for Permutation Combination; XORing based and pattern-based solutions.

Module 3: Optimizing Time & Space Using Sequential Storage

Coding for Optimizing time and Space using Sequential Storage: two pointer approach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string matching algorithms; Kadane's algo, stacks, priority-queues and hashing based efficient coding; median based problems and alternate solutions.

Module 4: Non-Linear Data Structures

Applying Non-Linear Data Structures for real-life problems: design of efficient solutions for problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path algos for CP problems with reduced time/space complexity.

Module 5: Problem Solving using Advanced Topics

CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

List of Laboratory Tasks:

1. You are given the finishing times of 'N' runners in a marathon. Write a program to find the runner who finished in the third position. **Focus:** Basic data structures (arrays), sorting algorithms (e.g., insertion sort, selection sort), and basic input/output.

2. In the same marathon, you are given the finishing times of 'N' runners and their bib numbers. Write a program to efficiently find the top 10 runners and their corresponding bib numbers. **Focus:** Efficient sorting algorithms (e.g., merge sort, quick sort), data structures like priority queues, and optimizing for large datasets.
3. A library maintains a list of books with their unique IDs. Write a program to check if a given book ID is present in the library. Focus: Searching algorithms (linear search), basic data structures (arrays or lists).
4. The library wants to implement a system to quickly find books by their titles. Suggest an efficient data structure (e.g., a hash table or a trie) and explain how to implement it to achieve fast book lookups. Focus: Understanding the trade-offs between different data structures, choosing the most appropriate data structure for a specific problem, and implementing efficient search operations.
5. An online store sells products with different prices. Write a program to calculate the total cost of a given list of products. **Focus:** Basic arithmetic operations, working with arrays or lists to store product prices.
6. The online store offers discounts based on the total purchase amount. Design an algorithm to efficiently calculate the final cost of an order, considering different discount rules (e.g., percentage discounts, fixed amount discounts, tiered discounts). Focus: Algorithmic design, conditional statements, handling complex scenarios with multiple rules, and potentially using dynamic programming techniques for optimization.
7. You are given two integers, 'a' and 'm'. Calculate 'a' raised to the power 'm' modulo a large prime number 'p'. **Focus:** Basic modular arithmetic operations (modular exponentiation), understanding the modulo operator.
8. In a secure communication system, you need to efficiently compute the modular exponentiation for very large values of 'm'. Implement and analyze the efficiency of the binary exponentiation algorithm for this task. **Focus:** Efficient algorithms for modular exponentiation (binary exponentiation), time complexity analysis, and understanding the importance of efficient algorithms in cryptography.
9. You have a deck of 'N' cards. Calculate the total number of possible hands of size 'K' that can be drawn from the deck. **Focus:** Basic combinatorics (combinations), factorial calculations.
10. In a card game, you need to calculate the probability of drawing certain combinations of cards (e.g., a pair, a three-of-a-kind) from a shuffled deck. Design an efficient algorithm to calculate these probabilities. **Focus:** Advanced combinatorics (permutations and combinations with repetitions), probability calculations, and optimizing calculations to avoid overflows.
11. You are given a network of devices represented as a graph. Determine if there is a path between two given devices in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search).
12. In a secure network, you need to detect and isolate compromised devices. Design an algorithm that efficiently identifies devices that exhibit anomalous behavior (e.g., unusual traffic patterns) using XOR-based techniques for data comparison and pattern matching. **Focus:** Applying XOR operations for data comparison and pattern recognition, understanding the properties of XOR (e.g., commutative, associative), and designing algorithms for network anomaly detection.

13. You are given an array representing the speeds of cars on a highway. Find the minimum time required for all cars to pass a certain point. **Focus:** Basic array traversal, finding the minimum element in an array.
14. In a more realistic scenario, cars have different lengths. Implement a two-pointer approach to simulate the movement of cars and determine the minimum time for all cars to pass a given point. **Focus:** Two-pointer technique, simulating real-world scenarios with arrays, optimizing time complexity.
15. Given a string, find the number of occurrences of a specific substring within the string. **Focus:** Basic string manipulation, string matching (brute-force approach).
16. Implement the KMP (Knuth-Morris-Pratt) string matching algorithm to efficiently find all occurrences of a given pattern within a large text document. **Focus:** Advanced string matching algorithms, understanding the concept of the "next" array in KMP, optimizing for large input sizes.
17. An online auction platform receives bids for different items. Implement a data structure (e.g., a priority queue) to efficiently track the highest bid for each item. **Focus:** Priority queues, insertion and extraction operations on priority queues, basic implementation of a priority queue using an array or a suitable library.
18. The auction platform needs to handle a large number of bids concurrently. Design and implement a system that efficiently processes bids, updates the highest bid for each item, and handles potential race conditions. **Focus:** Concurrent data structures and algorithms, thread safety, handling race conditions, optimizing for high-throughput scenarios.
19. A social network can be represented as a graph where users are nodes, and connections between users are edges. Write an algorithm to find if two given users are connected in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search), basic graph representation (adjacency list or adjacency matrix).
20. In a large social network, efficiently finding the shortest path between two users is crucial. Implement Dijkstra's algorithm to find the shortest paths between users in the network, considering edge weights (e.g., representing the strength of connections). **Focus:** Shortest path algorithms (Dijkstra's algorithm), graph algorithms with weighted edges, optimizing for large graphs.
21. A file system can be modeled as a tree structure. Implement a function to traverse the file system and print the names of all files and directories. **Focus:** Tree traversal algorithms (depth-first search or breadth-first search), basic tree representation (using nodes and pointers).
22. Design and implement a file system that supports efficient operations like creating directories, deleting files, and finding files based on their names or paths. Consider using a combination of tree structures and hash tables for efficient indexing and searching. **Focus:** Designing and implementing file system structures, using multiple data structures together, optimizing for common file system operations.
23. An online shopping cart can be represented as a tree, where each node represents an item or a category of items. Write an algorithm to calculate the total price of all items in the shopping cart. **Focus:** Tree traversal, calculating sums within a tree structure.
24. Implement a system that allows customers to apply discounts and coupons to their shopping carts. Consider using a combination of trees and other data structures (e.g., hash tables) to efficiently apply discounts and calculate the final price. **Focus:** Applying discounts and promotions to tree-like structures, efficient implementation of discount rules, optimizing for complex pricing scenarios.

25. In a social network, users can form groups. Given a list of friendships, determine if all users in a specific group are connected (directly or indirectly) through friendships. **Focus:** Disjoint set union (DSU) data structure, basic connectivity checks.
26. Design an efficient algorithm to find the minimum number of new friendships needed to connect all users in the social network into a single, connected component. **Focus:** Applying DSU for finding connected components, greedy algorithms, optimization for minimizing connections.
27. A treasure hunt involves a series of clues leading to the final treasure. Given a list of possible paths and their associated costs, find the cheapest path to reach the treasure. **Focus:** Greedy algorithms (e.g., Dijkstra's algorithm for shortest paths), basic graph representation.
28. In a more complex treasure hunt, there are time constraints associated with each path. Design an algorithm to find the fastest path to the treasure while considering both path costs and time constraints.

    Focus: Combining greedy approaches with other techniques (e.g., priority queues), handling multiple constraints, optimizing for time-critical scenarios.

29. In a simplified chess game with only rooks, determine the minimum number of moves required for a rook to reach a specific target square on an empty board. **Focus:** Breadth-first search (BFS) on a graph (the chessboard), basic graph traversal.
30. In a more realistic chess game with multiple pieces and obstacles, implement a minimax algorithm with alpha-beta pruning to determine the best move for a player. **Focus:** Game tree search, minimax algorithm, optimization techniques like alpha-beta pruning, handling complex game states.

Targeted Application & Tools that can be used:

1. C or C++ Compiler (g++): The standard compiler for CP. Familiarize students with compilation flags (e.g., -O2 for optimization).
2. IDE (Integrated Development Environment): Code:: Blocks, Visual Studio, CLion, or similar IDEs. These provide debugging capabilities, code completion, and other helpful features.
3. Online Judges (CodeChef, Codeforces, LeetCode, HackerRank): Essential for practicing and submitting solutions.
4. Debugger (gdb): Crucial for understanding code execution and finding bugs. Origin, excel and Mat lab soft wares for programming and data analysis.
5. Number Theory Libraries: Some libraries provide pre-built functions for number theory operations (though often it's better to implement them yourself for learning).
6. Wolfram Alpha: A useful tool for verifying number theory calculations and exploring concepts.
7. **String Libraries:** Familiarize students with the string manipulation functions available in C++.
8. **Graph Visualization Tools:** Tools like Graphviz can be helpful for visualizing graphs and understanding graph algorithms.
9. **DP Debugging Techniques:** Practice debugging DP solutions, as they can be complex. Visualizing the DP table can be helpful.

Text Books:

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

2 "Data Structures and Algorithms in Java: A Project-Based Approach" – *Dan S. Myers, Cambridge University Press*

Reference Books:

1. Data Structures and Algorithmic Thinking with Python/C++/Java", *Narasimha Karumanchi, 5th Edition, Career Monk, 2017.*
2. Introduction to Algorithms, Thomas H. Cormen (Author), Charles E. Leiserson (Author), Ronald L. Rivest , fourth edition April 2022

Web Resources

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

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

Assessment Type

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

Course Code: **LAW7601**	**Indian Constitution Type of Course: MOOC course**	**L- T- P- C**	**-**	**-**	**-**	**0**
		Contact hours	**-**	**-**	**-**	**-**
Course Pre-requisites	NIL					
Anti-requisites	**NIL**					
Course Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamentals of Indian Constitution concepts and their relevance to 75+ Years of Republic of India (https://constitution75.com/) as well as **#AzaadiKaAmrutMahotsav** / **Azadi Ka Amrit Mahotsav** (https://amritmahotsav.nic.in). It is designed to equip students with the knowledge about the Constitution of India. This course aims to introduce the constitutional law of India to students from all walks of life and help them understand the constitutional principles as applied and understood in everyday life. The objective of making the					

	Constitution of India, familiar to all students, and not only to law students, this course aims and objectifies legal understanding in the simplest of forms. **This course is designed to cater to Constitutional Studies.**
Course Objective	The objective of the course is **‘SKILL DEVELOPMENT’** of the student by using **‘PARTICIPATIVE LEARNING’** techniques
Course Outcomes	On successful completion of this course the students shall be able to: 1. Describe the basic understanding of the Indian Constitution and the concepts and issues relevant to day-to-day life of the nation and to equip the Citizen with the zeal of capacity building. Recognizing and identify the values of the Constitution of India. 2. Enabling the Citizen-centric Awareness of Rights and Responsibilities of the State 3. Explain the role of the State actors in building India. 4. Understanding the Gandhian vision over the power of the LSG (Local Self-Governance)
Course Content:	
Module 1	**Understanding the Making of the Constitution: The Constituent Assembly & The Constitution of India**
Topics: Historical Context of Constituent Assembly - Compositions & Functions of Constituent Assembly What is a Constitution? – Why have a Constitution? – Constitutional Change - Features of Indian Constitution – Preamble of Indian Constitution	
Module 2	**Citizen’s Fundamental Rights and State’s Responsibilities (Directive Principles)**
Topics: Introduction to Fundamental Rights - Right to Equality – Facets of Right to Equality - Right to Freedom - Constitutional Position of Some Democratic Rights - Right Against Exploitation - Right to Freedom of Religion - Right to Constitutional Remedies Directive Principles of the State Policy	
Module 3	**Organs Of the Government**
Topics: **Executive**: The President of India - Powers and Functions of President of India - Emergency Powers and the Position of the President **Legislature**: Union Council of Ministers - Prime Minister - The Rajya Sabha - The Lok Sabha - Relation between the Lok Sabha & Rajya Sabha - Office of the Speaker – Important Parliamentary Committees **Judiciary**: The Structure and Organization of the Judiciary & the High Court - The Supreme Court - Role of The Supreme Court - Judicial Activism in India - Basic Structure Doctrine & PIL	
Module 4	**Federalism & Decentralization**
Topics: What is Federalism? - Centre-State Legislative Relations - Centre-State Administrative Relations - Centre-State Financial Relations	

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

Targeted Application & Tools that can be used:

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

Tools: Online Tools – NPTEL and Swayam.

Project work/Assignment:

Assessment Type

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

Online Link*:

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

* Other source links are available in below Resources link.

Text Book

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

Reference Books

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

Resources:

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

Topics relevant to Skill Development:

1. An attitude of inquiry.
2. Write reports

The topics related to Constitutional Studies and its application :

All topics in theory component are relevant to Indian Constitution.

Course Code: CSE2272	**Course Title: Cloud Computing** **Type of Course: Theory**	**L- T-P- C**	2	0	0	2
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	Nil					
Course Description	Cloud Computing provides a hands-on comprehensive study of Cloud concepts and capabilities across the various Cloud service models including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). It dives into all of the details that a student needs to know in order to plan for developing applications on the cloud and what to look for when using applications or services hosted on a cloud.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of **CLOUD COMPUTING** and is designed to improve the learners' **SKILL DEVELOPMENT** through **PARTICIPATIVE LEARNING TECHNIQUES.**					
Course Out Comes	On successful completion of the course the students shall be able to: 1. Describe the fundamental components and layers of Cloud Computing Architecture. [Remember] 2. Identify appropriate Virtualization techniques to virtualize infrastructures [Understand] 3. Summarize various Cloud mechanisms to optimize the QoS parameters [Understand] 4. Apply cloud platforms to develop various applications [Apply]					
Course Content:						
Module 1	**Introduction to Cloud services**	Assignment	Theory			**L: 10**
Evolution of cloud computing, Computing Platforms and Technologies, Cloud Computing Architecture, IaaS, PaaS, SaaS, Types of Clouds, Cloud Computing Environments. [Understanding]						
Module 2	**Virtualization Techniques**	Assignment	Theory			**L: 10**
Basics of Virtualization - Types of Virtualizations, Taxonomy of Virtualization Techniques, Implementation Levels of Virtualization. [Understanding]						
Module 3	**Cloud QoS and Management**	Assignment	Theory			**L: 10**
Cloud Infrastructure Mechanisms- Logical Network Perimeter, Virtual Server, Cloud Storage Device, Cloud Usage Monitor, Ready-Made Environment, SLAs, Specialized Cloud Mechanisms- Automated Scaling Listener, Load Balancer, SLA Monitor, Pay-Per-Use Monitor, Audit Monitor, Cloud Security Mechanisms. [Understanding]						

Module 4	**Cloud Application development in Cloud**	Assignment	Theory		**L: 10**

Programming Models for Cloud Computing – MapReduce, CGL Mapreduce, Cloud Haskell, Development environments for
service development (Demonstration using AWS Cloud/Saturn Cloud); Dockers and Containers. [Apply]

Targeted Application & Tools that can be used :

Applications:

Cloud Platform, Use of cloud technology in different applications like healthcare, agriculture etc.

Tools:

1. Google App Engine
2. AWS, Saturn Cloud etc.

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

- **Students can design and implement dynamic resource allocation for virtual machine using cloud computing environment.**
- **Design and Implementation of a Scalable Cloud-Based Data Storage System**
- **Development of a Multi-Cloud Management Platform**

Text Book

1. **Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "Mastering Cloud Computing", McGraw Hill Education, 2013 edition.**

2. **John Rittinghouse and James Ransome, "Cloud Computing, Implementation, Management and Security", CRC Press, 2010 edition.**

References

1. **Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, "Cloud Computing Concepts, Technology & Architecture", PHI publisher 2013 edition.**

2. **K. Chandrasekaran, "Essentials of CLOUD COMPUTING", CRC Press, 2015 edition.**

3. **David E.Y. Sarna, "Implementing and Developing Cloud Applications", CRC Press, 2018 edition.**

4. **Manvi, Sunilkumar, and Gopal K. Shyam. "Cloud Computing: Concepts and Technologies". CRC Press, 2021.**

Web Based Resources and E-books:

W1. IEEE Transactions on Cloud Computing- https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6245519 W2. International Journal of Cloud Computing- https://www.inderscience.com/jhome.php?jcode=ijcc

W3. CloudSim Resources https://javadoc.io/doc/org.cloudsimplus/cloudsim-plus/latest/org/cloudbus/cloudsim/resources/class-use/Resource.html

W4. Journal of Network and Computer Networking- https://www.journals.elsevier.com/journal-of-network-and-computer- applications

Topics relevant to "Skill Development": AWS, Azure, APIs, Aneka Cloud Platform, Virtualization, Cloud Platforms in Industry, EC2, Installation of VM Workstation, Cloud Infrastructure and Challenges for Skill Development through
Participative Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: CSE2273	Course Title: Cloud Computing Lab	L- T-P- C	0	0	2	1

Version No.	1.0
Course Pre-requisites	**Nil**
Anti-requisites	Nil
Course Description	Cloud Computing provides a hands-on comprehensive study of Cloud concepts and capabilities across the various Cloud service models including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). It dives into all of the details that a student needs to know in order to plan for developing applications on the cloud and what to look for when using applications or services hosted on a cloud.
Course Objectives	The objective of the course is to familiarize the learners with the concepts of **CLOUD COMPUTING** and is designed to improve the learners' **SKILL DEVELOPMENT** through **PARTICIPATIVE LEARNING TECHNIQUES.**
Course Content:	

Targeted Application & Tools that can be used :

Applications:

Cloud Platform, Use of cloud technology in different applications like healthcare, agriculture etc.

Tools:

1. Google App Engine
2. AWS, Saturn Cloud etc.

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

- **Students can design and implement dynamic resource allocation for virtual machine using cloud computing environment.**
- **Design and Implementation of a Scalable Cloud-Based Data Storage System**
- **Development of a Multi-Cloud Management Platform**

List of Laboratory Tasks:

Experiments:

1. **Create a simple cloud software application and provide it as a service using any Cloud Service Provider to demonstrate Software as a Service (SaaS).**
2. **Create a Virtual Machine with 1 vCPU, 2GB RAM and 15GB storage disk using a Type 2 Virtualization Software**
3. **Create a Virtual Hard Disk and allocate the storage using VM ware Workstation**
4. **Create a Snapshot and Cloning of a VM and Test it by loading the Previous Version/Cloned VM**
5. **Demonstrate Infrastructure as a Service (IaaS) by Creating a Virtual Machine using a Public Cloud Service Provider (Azure/GCP/AWS), configure with minimum CPU, RAM, and Storage and Launch the VM image.**
6. **Create a Simple Web Application using Java or Python and host it in any Public Cloud Service Provider (Azure/GCP/AWS) to demonstrate Platform as a Service (PaaS)**

7. Create a Storage service using any Public Cloud Service Provider (Azure/GCP/AWS) and check the public accessibility of the stored file to demonstrate Storage as a Service
8. Create a SQL storage service and perform a basic query using any Public Cloud Service Provider (Azure/GCP/AWS) to demonstrate Database as a Service (DaaS)
9. Perform the basic configuration setup for Installing Hadoop 2.x like Creating the HDUSER and SSH localhost
10. Install Hadoop 2.x and configure the Name Node and Data Node.
11. Launch the Hadoop 2.x and perform MapReduce Program for a Word Count problem

Text Book

3. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "Mastering Cloud Computing", McGraw Hill Education, 2013 edition.

4. John Rittinghouse and James Ransome, "Cloud Computing, Implementation, Management and Security", CRC Press, 2010 edition.

References

Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, "Cloud Computing Concepts, Technology & Architecture", PHI publisher 2013 edition.

K. Chandrasekaran, "Essentials of CLOUD COMPUTING", CRC Press, 2015 edition.

David E.Y. Sarna, "Implementing and Developing Cloud Applications", CRC Press, 2018 edition.

Manvi, Sunilkumar, and Gopal K. Shyam. "Cloud Computing: Concepts and Technologies". CRC Press, 2021.

Web Based Resources and E-books:

W1. IEEE Transactions on Cloud Computing-
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6245519 W2. International Journal of Cloud Computing- https://www.inderscience.com/jhome.php?jcode=ijcc
W3. CloudSim Resources
https://javadoc.io/doc/org.cloudsimplus/cloudsim-plus/latest/org/cloudbus/cloudsim/resources/class-use/Resource.html
W4. Journal of Network and Computer Networking- https://www.journals.elsevier.com/journal-of-network-and-computer- applications

Topics relevant to "Skill Development": AWS, Azure, APIs, Aneka Cloud Platform, Virtualization, Cloud Platforms in Industry, EC2, Installation of VM Workstation, Cloud Infrastructure and Challenges for Skill Development through
Participative Learning techniques. This is attained through assessment component mentioned in course handout.

Course Code: **CSE2274**	**Course Title:** Competitive Programming and Problem Solving	L-T-P-C	0	0	4	2
Version No.	1.0					

Course Pre-requisites	NIL
Anti-requisites	**NIL**
Course Description	The **Competitive Programming and Problem Solving** course equips students with efficient problem-solving skills for coding competitions and real-world challenges. Starting with brute-force solutions, students learn to optimize time and space complexity using advanced techniques like dynamic programming, greedy algorithms, and backtracking. Hands-on practice on platforms like CodeChef and Codeforces helps tackle problems involving number theory, data structures, and algorithmic paradigms. By understanding CP constraints and fostering a strategic mindset, students gain the confidence to excel in competitions, technical interviews, and practical applications.
Course Out Comes	On successful completion of the course the students shall be able to: CO1 : Understanding the issues of online platforms and Competitive Programming (CP) and developing brute force coding for commonly asked CP problems. CO2 : Analyzing the space and time complexity of brute force solutions and designing efficient solutions. CO3 : Evaluating the applicability of suitable algorithmic approaches to solve relevant CP problems. CO4: Creating efficient solutions of CP problems using the learnt algorithmic approaches.
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Competitive Programming and Problem Solving** and attain **Skill Development** through **Experiential Learning** techniques.

Module 1: Introduction to Competitive Programming

Overview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforces etc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL

Module 2: Number Theory for Problem-Solving

Use of Number Theory for problem-solving: reducing time/space complexity of brute force coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient coding
for Permutation Combination; XORing based and pattern-based solutions.

Module 3: Optimizing Time & Space Using Sequential Storage

Coding for Optimizing time and Space using Sequential Storage: two pointer approach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string
matching algorithms; Kadane's algo, stacks, priority-queues and hashing based efficient coding; median based problems and alternate solutions.

Module 4: Non-Linear Data Structures

Applying Non-Linear Data Structures for real-life problems: design of efficient solutions for problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path
algos for CP problems with reduced time/space complexity.

Module 5: Problem Solving using Advanced Topics

CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

List of Laboratory Tasks:

31. You are given the finishing times of 'N' runners in a marathon. Write a program to find the runner who finished in the third position. **Focus:** Basic data structures (arrays), sorting algorithms (e.g., insertion sort, selection sort), and basic input/output.
32. In the same marathon, you are given the finishing times of 'N' runners and their bib numbers. Write a program to efficiently find the top 10 runners and their corresponding bib numbers. **Focus:** Efficient sorting algorithms (e.g., merge sort, quick sort), data structures like priority queues, and optimizing for large datasets.
33. A library maintains a list of books with their unique IDs. Write a program to check if a given book ID is present in the library. Focus: Searching algorithms (linear search), basic data structures (arrays or lists).
34. The library wants to implement a system to quickly find books by their titles. Suggest an efficient data structure (e.g., a hash table or a trie) and explain how to implement it to achieve fast book lookups. Focus: Understanding the trade-offs between different data structures, choosing the most appropriate data structure for a specific problem, and implementing efficient search operations.
35. An online store sells products with different prices. Write a program to calculate the total cost of a given list of products. **Focus:** Basic arithmetic operations, working with arrays or lists to store product prices.
36. The online store offers discounts based on the total purchase amount. Design an algorithm to efficiently calculate the final cost of an order, considering different discount rules (e.g., percentage discounts, fixed amount discounts, tiered discounts). Focus: Algorithmic design, conditional statements, handling complex scenarios with multiple rules, and potentially using dynamic programming techniques for optimization.
37. You are given two integers, 'a' and 'm'. Calculate 'a' raised to the power 'm' modulo a large prime number 'p'. **Focus:** Basic modular arithmetic operations (modular exponentiation), understanding the modulo operator.
38. In a secure communication system, you need to efficiently compute the modular exponentiation for very large values of 'm'. Implement and analyze the efficiency of the

binary exponentiation algorithm for this task. **Focus:** Efficient algorithms for modular exponentiation (binary exponentiation), time complexity analysis, and understanding the importance of efficient algorithms in cryptography.

39. You have a deck of 'N' cards. Calculate the total number of possible hands of size 'K' that can be drawn from the deck. **Focus:** Basic combinatorics (combinations), factorial calculations.
40. In a card game, you need to calculate the probability of drawing certain combinations of cards (e.g., a pair, a three-of-a-kind) from a shuffled deck. Design an efficient algorithm to calculate these probabilities. **Focus:** Advanced combinatorics (permutations and combinations with repetitions), probability calculations, and optimizing calculations to avoid overflows.
41. You are given a network of devices represented as a graph. Determine if there is a path between two given devices in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search).
42. In a secure network, you need to detect and isolate compromised devices. Design an algorithm that efficiently identifies devices that exhibit anomalous behavior (e.g., unusual traffic patterns) using XOR-based techniques for data comparison and pattern matching. **Focus:** Applying XOR operations for data comparison and pattern recognition, understanding the properties of XOR (e.g., commutative, associative), and designing algorithms for network anomaly detection.
43. You are given an array representing the speeds of cars on a highway. Find the minimum time required for all cars to pass a certain point. **Focus:** Basic array traversal, finding the minimum element in an array.
44. In a more realistic scenario, cars have different lengths. Implement a two-pointer approach to simulate the movement of cars and determine the minimum time for all cars to pass a given point. **Focus:** Two-pointer technique, simulating real-world scenarios with arrays, optimizing time complexity.
45. Given a string, find the number of occurrences of a specific substring within the string. **Focus:** Basic string manipulation, string matching (brute-force approach).
46. Implement the KMP (Knuth-Morris-Pratt) string matching algorithm to efficiently find all occurrences of a given pattern within a large text document. **Focus:** Advanced string matching algorithms, understanding the concept of the "next" array in KMP, optimizing for large input sizes.
47. An online auction platform receives bids for different items. Implement a data structure (e.g., a priority queue) to efficiently track the highest bid for each item. **Focus:** Priority queues, insertion and extraction operations on priority queues, basic implementation of a priority queue using an array or a suitable library.
48. The auction platform needs to handle a large number of bids concurrently. Design and implement a system that efficiently processes bids, updates the highest bid for each item, and handles potential race conditions. **Focus:** Concurrent data structures and algorithms, thread safety, handling race conditions, optimizing for high-throughput scenarios.
49. A social network can be represented as a graph where users are nodes, and connections between users are edges. Write an algorithm to find if two given users are connected in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search), basic graph representation (adjacency list or adjacency matrix).
50. In a large social network, efficiently finding the shortest path between two users is crucial. Implement Dijkstra's algorithm to find the shortest paths between users in the network,

considering edge weights (e.g., representing the strength of connections). **Focus:** Shortest path algorithms (Dijkstra's algorithm), graph algorithms with weighted edges, optimizing for large graphs.

51. A file system can be modeled as a tree structure. Implement a function to traverse the file system and print the names of all files and directories. **Focus:** Tree traversal algorithms (depth-first search or breadth-first search), basic tree representation (using nodes and pointers).
52. Design and implement a file system that supports efficient operations like creating directories, deleting files, and finding files based on their names or paths. Consider using a combination of tree structures and hash tables for efficient indexing and searching. **Focus:** Designing and implementing file system structures, using multiple data structures together, optimizing for common file system operations.
53. An online shopping cart can be represented as a tree, where each node represents an item or a category of items. Write an algorithm to calculate the total price of all items in the shopping cart. **Focus:** Tree traversal, calculating sums within a tree structure.
54. Implement a system that allows customers to apply discounts and coupons to their shopping carts. Consider using a combination of trees and other data structures (e.g., hash tables) to efficiently apply discounts and calculate the final price. **Focus:** Applying discounts and promotions to tree-like structures, efficient implementation of discount rules, optimizing for complex pricing scenarios.
55. In a social network, users can form groups. Given a list of friendships, determine if all users in a specific group are connected (directly or indirectly) through friendships. **Focus:** Disjoint set union (DSU) data structure, basic connectivity checks.
56. Design an efficient algorithm to find the minimum number of new friendships needed to connect all users in the social network into a single, connected component. **Focus:** Applying DSU for finding connected components, greedy algorithms, optimization for minimizing connections.
57. A treasure hunt involves a series of clues leading to the final treasure. Given a list of possible paths and their associated costs, find the cheapest path to reach the treasure. **Focus:** Greedy algorithms (e.g., Dijkstra's algorithm for shortest paths), basic graph representation.
58. In a more complex treasure hunt, there are time constraints associated with each path. Design an algorithm to find the fastest path to the treasure while considering both path costs and time constraints.

    Focus: Combining greedy approaches with other techniques (e.g., priority queues), handling multiple constraints, optimizing for time-critical scenarios.
59. In a simplified chess game with only rooks, determine the minimum number of moves required for a rook to reach a specific target square on an empty board. **Focus:** Breadth-first search (BFS) on a graph (the chessboard), basic graph traversal.
60. In a more realistic chess game with multiple pieces and obstacles, implement a minimax algorithm with alpha-beta pruning to determine the best move for a player. **Focus:** Game tree search, minimax algorithm, optimization techniques like alpha-beta pruning, handling complex game states.

Targeted Application & Tools that can be used:

10. C or C++ Compiler (g++): The standard compiler for CP. Familiarize students with compilation flags (e.g., -O2 for optimization).
11. IDE (Integrated Development Environment): Code:: Blocks, Visual Studio, CLion, or similar IDEs. These provide debugging capabilities, code completion, and other helpful features.
12. Online Judges (CodeChef, Codeforces, LeetCode, HackerRank): Essential for practicing and submitting solutions.
13. Debugger (gdb): Crucial for understanding code execution and finding bugs. Origin, excel and Mat lab soft wares for programming and data analysis.
14. Number Theory Libraries: Some libraries provide pre-built functions for number theory operations (though often it's better to implement them yourself for learning).
15. Wolfram Alpha: A useful tool for verifying number theory calculations and exploring concepts.
16. **String Libraries:** Familiarize students with the string manipulation functions available in C++.
17. **Graph Visualization Tools:** Tools like Graphviz can be helpful for visualizing graphs and understanding graph algorithms.
18. **DP Debugging Techniques:** Practice debugging DP solutions, as they can be complex. Visualizing the DP table can be helpful.

Text Books:

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

4 "Data Structures and Algorithms in Java: A Project-Based Approach" – ***Dan S. Myers, Cambridge University Press***

Reference Books:

3. Data Structures and Algorithmic Thinking with Python/C++/Java", ***Narasimha Karumanchi, 5th Edition, Career Monk, 2017.***

4. Introduction to Algorithms, Thomas H. Cormen (Author), Charles E. Leiserson (Author), Ronald L. Rivest , fourth edition April 2022

Web Resources

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

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

Assessment Type

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

Course Code: LAW7601	**Indian Constitution Type of Course: MOOC course**	**L- T- P- C**	-	-	-	0
		Contact hours	-	-	-	-
Course Pre-requisites	NIL					
Anti-requisites	**NIL**					
Course Description	This course is designed to improve the learners' SKILL DEVELOPMENT by using PATICIPATIVE LEARNING techniques. This course aims to familiarize students with fundamentals of Indian Constitution concepts and their relevance to 75+ Years of Republic of India (https://constitution75.com/) as well as **#AzaadiKaAmrutMahotsav** / **Azadi Ka Amrit Mahotsav** (https://amritmahotsav.nic.in). It is designed to equip students with the knowledge about the Constitution of India. This course aims to introduce the constitutional law of India to students from all walks of life and help them understand the constitutional principles as applied and understood in everyday life. The objective of making the Constitution of India, familiar to all students, and not only to law students, this course aims and objectifies legal understanding in the simplest of forms. **This course is designed to cater to Constitutional Studies.**					
Course Objective	The objective of the course is **'SKILL DEVELOPMENT'** of the student by using **'PARTICIPATIVE LEARNING'** techniques					
Course Outcomes	On successful completion of this course the students shall be able to: 5. 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. 6. Enabling the Citizen-centric Awareness of Rights and Responsibilities of the State 7. Explain the role of the State actors in building India. 8. Understanding the Gandhian vision over the power of the LSG (Local Self-Governance)					
Course Content:						
Module 1	**Understanding the Making of the Constitution: The Constituent Assembly & The Constitution of India**					
Topics: Historical Context of Constituent Assembly - Compositions & Functions of Constituent Assembly What is a Constitution? – Why have a Constitution? – Constitutional Change - Features of Indian Constitution – Preamble of Indian Constitution						
Module 2	**Citizen's Fundamental Rights and State's Responsibilities (Directive Principles)**					
Topics: Introduction to Fundamental Rights - Right to Equality – Facets of Right to Equality - Right to Freedom - Constitutional Position of Some Democratic Rights - Right Against Exploitation - Right to Freedom of Religion - Right to Constitutional Remedies Directive Principles of the State Policy						

Module 3	Organs Of the Government

Topics:

Executive: The President of India - Powers and Functions of President of India - Emergency Powers and the Position of the President
Legislature: Union Council of Ministers - Prime Minister - The Rajya Sabha - The Lok Sabha - Relation between the Lok Sabha & Rajya Sabha - Office of the Speaker – Important Parliamentary Committees
Judiciary: The Structure and Organization of the Judiciary & the High Court - The Supreme Court - Role of The Supreme Court - Judicial Activism in India - Basic Structure Doctrine & PIL

Module 4	Federalism & Decentralization		

Topics:

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

Targeted Application & Tools that can be used:

Application areas to familiarize students with fundamentals of Indian Constitutional concepts.
Tools: Online Tools – NPTEL and Swayam.

Project work/Assignment:

Assessment Type

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

Online Link*:

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

* Other source links are available in below Resources link.

Text Book

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

Reference Books

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

Resources:

7. https://onlinecourses.nptel.ac.in/noc20_lw03/course?&force_user=true
8. https://onlinecourses.swayam2.ac.in/cec19_hs13/course?&force_user=true
9. https://nptel.ac.in/courses/129106003
10. https://nptel.ac.in/courses/129106411
11. https://nptel.ac.in/courses/129105608
12. https://nptel.ac.in/courses/129106002

Topics relevant to Skill Development:

3. An attitude of inquiry.
4. Write reports

The topics related to Constitutional Studies and its application :

All topics in theory component are relevant to Indian Constitution.

| Course Code: CBC2000 | Course Title: Blockchain Technology and Applications | L:T:P:C = 3:0:0:3

| Version 1.0 |

Course Pre-requisites: Basics of Computer Networks, Database Systems

Anti-requisites: NIL

Course Description:
This course introduces the core concepts of blockchain technology, including its structure, consensus algorithms, and cryptographic foundations. It explores the various applications of blockchain in sectors such as finance, healthcare, supply chain, and digital identity. Students will also understand the legal, ethical, and scalability challenges of blockchain systems.

Course Objectives:
1. Understand the foundational principles of blockchain technology.
2. Explore blockchain architecture, cryptographic elements, and consensus mechanisms.
3. Analyze real-world blockchain use cases and evaluate their effectiveness.
4. Examine privacy, scalability, legal, and governance issues in blockchain ecosystems.

Course Outcomes:
CO1 (Understand): Describe blockchain architecture, key components, and cryptographic elements.
CO2 (Analyze): Compare different types of blockchains and consensus mechanisms.
CO3 (Apply): Demonstrate blockchain use cases across various industries.

CO4 (Apply): Analyze the social, legal, and ethical implications of blockchain adoption.

Course Content:

Module 1: Introduction to Blockchain and Cryptographic Foundations | No. of Sessions: 11
Distributed systems, Introduction to Blockchain, Properties of blockchain, Cryptographic hash functions, Merkle trees, Digital signatures, Public key infrastructure (PKI), Blockchain transactions

----

Module 2: Blockchain Types and Consensus Mechanisms | No. of Sessions: 11
Permissioned vs Permissionless blockchains, Public vs Private vs Consortium blockchains, Bitcoin architecture, Ethereum overview, Consensus algorithms: PoW, PoS, DPoS, PBFT

Module 3: Blockchain Platforms and Applications | No. of Sessions: 12
Smart contracts and dApps, Ethereum use cases, Hyperledger Fabric, Blockchain in finance (DeFi, tokenization), Healthcare, Supply chain, Identity verification, Voting systems

----

Module 4: Challenges, Governance, and Future Trends | No. of Sessions: 11
Scalability and interoperability, Energy consumption, Privacy challenges, Legal and regulatory frameworks, Blockchain governance models, Emerging trends: CBDCs, NFTs, Web3, DAOs

----

Textbooks:
T1: Melanie Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly Media, 2015
T2: Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020

Reference Books:
R1: Arshdeep Bahga, Vijay Madisetti, *Blockchain Applications: A Hands-on Approach*, VPT, 2017
R2: Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, 2nd Edition, 2017
R3: Bettina Warburg, *Basics of Blockchain*, LTS Publishing, 2019
R4: Nitin Gaur et al., *Blockchain for Business*, Pearson Education, 2021

Web Resources:

W1: https://ethereum.org
W2: https://hyperledger.org
W3: https://bitcoin.org/en/
W4: https://blockchainhub.net
W5: https://web3.foundation

Course Code: CBC2505
Course Title: Blockchain Security and Performance
L:T:P:C – 3:0:0:3

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 Objectives

— 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

CO1 (Understand): Identify and describe security challenges in blockchain platforms
CO2 (Analyze): Evaluate consensus attacks and performance issues in different blockchain frameworks
CO3 (Apply): Implement security-aware smart contracts and mitigation techniques
CO4 (Apply): Analyze and optimize blockchain performance under diverse network loads

Course Content (45 Hours Total)
Module 1: Blockchain Security Foundations – 11 Sessions

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 – 11 Sessions

Ethereum vulnerabilities, DAO attack analysis, Reentrancy, Arithmetic overflow/underflow, Front-running, Best practices in secure smart contract design, Tools for smart contract auditing

Module 3: Blockchain Performance Metrics – 11 Sessions
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 – 12 Sessions
Layer-2 scaling (State Channels, Rollups), Sharding, DAG-based architectures, ZKPs, Mixers and Privacy coins, Secure multi-party computation, Formal verification of performance

Textbooks
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

Reference Books
R1: Antonopoulos & Wood, *Mastering Ethereum*, O'Reilly Media
R2: Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press
R3: Koshy & Arvind, *Blockchain Security and Performance*, Springer, 2021
R4: Bonneau et al., *Security Analysis of Blockchain Protocols*, ACM Reviews

Web Resources
W1: https://consensys.net
W2: https://ethereum.org/en/developers/docs/security/
W3: https://chainsecurity.com/
W4: https://cryptozombies.io
W5: https://bitcoin.org/en/security

Course Code: CBC2505
Course Title: Blockchain Security and Performance
L:T:P:C – 3:0:0:3

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 Objectives

— 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

CO1 (Understand): Identify and describe security challenges in blockchain platforms
CO2 (Analyze): Evaluate consensus attacks and performance issues in different blockchain frameworks
CO3 (Apply): Implement security-aware smart contracts and mitigation techniques
CO4 (Apply): Analyze and optimize blockchain performance under diverse network loads

Course Content (45 Hours Total)

Module 1: Blockchain Security Foundations – 11 Sessions
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 – 11 Sessions
Ethereum vulnerabilities, DAO attack analysis, Reentrancy, Arithmetic overflow/underflow, Front-running, Best practices in secure smart contract design, Tools for smart contract auditing

Module 3: Blockchain Performance Metrics – 11 Sessions
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 – 12 Sessions
Layer-2 scaling (State Channels, Rollups), Sharding, DAG-based architectures, ZKPs, Mixers and Privacy coins, Secure multi-party computation, Formal verification of performance

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

Textbooks

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

Reference Books

R1: Antonopoulos & Wood, *Mastering Ethereum*, O'Reilly Media

R2: Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press

R3: Koshy & Arvind, *Blockchain Security and Performance*, Springer, 2021

R4: Bonneau et al., *Security Analysis of Blockchain Protocols*, ACM Reviews

Web Resources

W1: https://consensys.net

W2: https://ethereum.org/en/developers/docs/security/

W3: https://chainsecurity.com/

W4: https://cryptozombies.io

W5: https://bitcoin.org/en/security

Course Code:	**Course Title:** Competitive Programming					

CSE2510	and Problem Solving **Type of Course: Program Core**	L-T-P-C	0	0		

Version No.	1.0
Course Pre-requisites	NIL
Anti-requisites	**NIL**
Course Description	The **Competitive Programming and Problem Solving** course equips students with efficient problem-solving skills for coding competitions and real-world challenges. Starting with brute-force solutions, students learn to optimize time and space complexity using advanced techniques like dynamic programming, greedy algorithms, and backtracking. Hands-on practice on platforms like CodeChef and Codeforces helps tackle problems involving number theory, data structures, and algorithmic paradigms. By understanding CP constraints and fostering a strategic mindset, students gain the confidence to excel in competitions, technical interviews, and practical applications.
Course Out Comes	On successful completion of the course the students shall be able to: CO1 : Understanding the issues of online platforms and Competitive Programming (CP) and developing brute force coding for commonly asked CP problems. CO2 : Analyzing the space and time complexity of brute force solutions and designing efficient solutions. CO3 : Evaluating the applicability of suitable algorithmic approaches to solve relevant CP problems. CO4: Creating efficient solutions of CP problems using the learnt algorithmic approaches.
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Competitive Programming and Problem Solving** and attain **Skill Development** through **Experiential Learning** techniques.

Module 1: Introduction to Competitive Programming

Overview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforces etc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL

Module 2: Number Theory for Problem-Solving

Use of Number Theory for problem-solving: reducing time/space complexity of brute force coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient

coding
for Permutation Combination; XORing based and pattern-based solutions.

Module 3: Optimizing Time & Space Using Sequential Storage

Coding for Optimizing time and Space using Sequential Storage: two pointer approach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string
matching algorithms; Kadane's algo, stacks, priority-queues and hashing based efficient coding; median based problems and alternate solutions.

Module 4: Non-Linear Data Structures

Applying Non-Linear Data Structures for real-life problems: design of efficient solutions for problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path
algos for CP problems with reduced time/space complexity.

Module 5: Problem Solving using Advanced Topics

CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

List of Laboratory Tasks:

61. You are given the finishing times of 'N' runners in a marathon. Write a program to find the runner who finished in the third position. **Focus:** Basic data structures (arrays), sorting algorithms (e.g., insertion sort, selection sort), and basic input/output.
62. In the same marathon, you are given the finishing times of 'N' runners and their bib numbers. Write a program to efficiently find the top 10 runners and their corresponding bib numbers. **Focus:** Efficient sorting algorithms (e.g., merge sort, quick sort), data structures like priority queues, and optimizing for large datasets.
63. A library maintains a list of books with their unique IDs. Write a program to check if a given book ID is present in the library. Focus: Searching algorithms (linear search), basic data structures (arrays or lists).
64. The library wants to implement a system to quickly find books by their titles. Suggest an efficient data structure (e.g., a hash table or a trie) and explain how to implement it to achieve fast book lookups. Focus: Understanding the trade-offs between different data structures, choosing the most appropriate data structure for a specific problem, and implementing efficient search operations.
65. An online store sells products with different prices. Write a program to calculate the total cost of a given list of products. **Focus:** Basic arithmetic operations, working with arrays or lists to store product prices.
66. The online store offers discounts based on the total purchase amount. Design an algorithm to efficiently calculate the final cost of an order, considering different discount rules (e.g., percentage discounts, fixed amount discounts, tiered discounts). Focus: Algorithmic design, conditional statements, handling complex scenarios with multiple rules, and potentially using dynamic programming techniques for optimization.

67. You are given two integers, 'a' and 'm'. Calculate 'a' raised to the power 'm' modulo a large prime number 'p'. **Focus:** Basic modular arithmetic operations (modular exponentiation), understanding the modulo operator.
68. In a secure communication system, you need to efficiently compute the modular exponentiation for very large values of 'm'. Implement and analyze the efficiency of the binary exponentiation algorithm for this task. **Focus:** Efficient algorithms for modular exponentiation (binary exponentiation), time complexity analysis, and understanding the importance of efficient algorithms in cryptography.
69. You have a deck of 'N' cards. Calculate the total number of possible hands of size 'K' that can be drawn from the deck. **Focus:** Basic combinatorics (combinations), factorial calculations.
70. In a card game, you need to calculate the probability of drawing certain combinations of cards (e.g., a pair, a three-of-a-kind) from a shuffled deck. Design an efficient algorithm to calculate these probabilities. **Focus:** Advanced combinatorics (permutations and combinations with repetitions), probability calculations, and optimizing calculations to avoid overflows.
71. You are given a network of devices represented as a graph. Determine if there is a path between two given devices in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search).
72. In a secure network, you need to detect and isolate compromised devices. Design an algorithm that efficiently identifies devices that exhibit anomalous behavior (e.g., unusual traffic patterns) using XOR-based techniques for data comparison and pattern matching. **Focus:** Applying XOR operations for data comparison and pattern recognition, understanding the properties of XOR (e.g., commutative, associative), and designing algorithms for network anomaly detection.
73. You are given an array representing the speeds of cars on a highway. Find the minimum time required for all cars to pass a certain point. **Focus:** Basic array traversal, finding the minimum element in an array.
74. In a more realistic scenario, cars have different lengths. Implement a two-pointer approach to simulate the movement of cars and determine the minimum time for all cars to pass a given point. **Focus:** Two-pointer technique, simulating real-world scenarios with arrays, optimizing time complexity.
75. Given a string, find the number of occurrences of a specific substring within the string. **Focus:** Basic string manipulation, string matching (brute-force approach).
76. Implement the KMP (Knuth-Morris-Pratt) string matching algorithm to efficiently find all occurrences of a given pattern within a large text document. **Focus:** Advanced string matching algorithms, understanding the concept of the "next" array in KMP, optimizing for large input sizes.
77. An online auction platform receives bids for different items. Implement a data structure (e.g., a priority queue) to efficiently track the highest bid for each item. **Focus:** Priority queues, insertion and extraction operations on priority queues, basic implementation of a priority queue using an array or a suitable library.
78. The auction platform needs to handle a large number of bids concurrently. Design and implement a system that efficiently processes bids, updates the highest bid for each item, and handles potential race conditions. **Focus:** Concurrent data structures and algorithms, thread safety, handling race conditions, optimizing for high-throughput scenarios.

79. A social network can be represented as a graph where users are nodes, and connections between users are edges. Write an algorithm to find if two given users are connected in the network. **Focus:** Graph traversal algorithms (depth-first search or breadth-first search), basic graph representation (adjacency list or adjacency matrix).
80. In a large social network, efficiently finding the shortest path between two users is crucial. Implement Dijkstra's algorithm to find the shortest paths between users in the network, considering edge weights (e.g., representing the strength of connections). **Focus:** Shortest path algorithms (Dijkstra's algorithm), graph algorithms with weighted edges, optimizing for large graphs.
81. A file system can be modeled as a tree structure. Implement a function to traverse the file system and print the names of all files and directories. **Focus:** Tree traversal algorithms (depth-first search or breadth-first search), basic tree representation (using nodes and pointers).
82. Design and implement a file system that supports efficient operations like creating directories, deleting files, and finding files based on their names or paths. Consider using a combination of tree structures and hash tables for efficient indexing and searching. **Focus:** Designing and implementing file system structures, using multiple data structures together, optimizing for common file system operations.
83. An online shopping cart can be represented as a tree, where each node represents an item or a category of items. Write an algorithm to calculate the total price of all items in the shopping cart. **Focus:** Tree traversal, calculating sums within a tree structure.
84. Implement a system that allows customers to apply discounts and coupons to their shopping carts. Consider using a combination of trees and other data structures (e.g., hash tables) to efficiently apply discounts and calculate the final price. **Focus:** Applying discounts and promotions to tree-like structures, efficient implementation of discount rules, optimizing for complex pricing scenarios.
85. In a social network, users can form groups. Given a list of friendships, determine if all users in a specific group are connected (directly or indirectly) through friendships. **Focus:** Disjoint set union (DSU) data structure, basic connectivity checks.
86. Design an efficient algorithm to find the minimum number of new friendships needed to connect all users in the social network into a single, connected component. **Focus:** Applying DSU for finding connected components, greedy algorithms, optimization for minimizing connections.
87. A treasure hunt involves a series of clues leading to the final treasure. Given a list of possible paths and their associated costs, find the cheapest path to reach the treasure. **Focus:** Greedy algorithms (e.g., Dijkstra's algorithm for shortest paths), basic graph representation.
88. In a more complex treasure hunt, there are time constraints associated with each path. Design an algorithm to find the fastest path to the treasure while considering both path costs and time constraints.

    Focus: Combining greedy approaches with other techniques (e.g., priority queues), handling multiple constraints, optimizing for time-critical scenarios.
89. In a simplified chess game with only rooks, determine the minimum number of moves required for a rook to reach a specific target square on an empty board. **Focus:** Breadth-first search (BFS) on a graph (the chessboard), basic graph traversal.

90. In a more realistic chess game with multiple pieces and obstacles, implement a minimax algorithm with alpha-beta pruning to determine the best move for a player. **Focus:** Game tree search, minimax algorithm, optimization techniques like alpha-beta pruning, handling complex game states.

Targeted Application & Tools that can be used:

19. C or C++ Compiler (g++): The standard compiler for CP. Familiarize students with compilation flags (e.g., -O2 for optimization).
20. IDE (Integrated Development Environment): Code:: Blocks, Visual Studio, CLion, or similar IDEs. These provide debugging capabilities, code completion, and other helpful features.
21. Online Judges (CodeChef, Codeforces, LeetCode, HackerRank): Essential for practicing and submitting solutions.
22. Debugger (gdb): Crucial for understanding code execution and finding bugs. Origin, excel and Mat lab soft wares for programming and data analysis.
23. Number Theory Libraries: Some libraries provide pre-built functions for number theory operations (though often it's better to implement them yourself for learning).
24. Wolfram Alpha: A useful tool for verifying number theory calculations and exploring concepts.
25. **String Libraries:** Familiarize students with the string manipulation functions available in C++.
26. **Graph Visualization Tools:** Tools like Graphviz can be helpful for visualizing graphs and understanding graph algorithms.
27. **DP Debugging Techniques:** Practice debugging DP solutions, as they can be complex. Visualizing the DP table can be helpful.

Text Books:

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

6 **"Data Structures and Algorithms in Java: A Project-Based Approach" –** ***Dan S. Myers, Cambridge University Press***

Reference Books:

5. **Data Structures and Algorithmic Thinking with Python/C++/Java",** ***Narasimha Karumanchi, 5th Edition, Career Monk, 2017.***
6. **Introduction to Algorithms, Thomas H. Cormen (Author), Charles E. Leiserson (Author), Ronald L. Rivest , fourth edition April 2022**

Web Resources

3. **https://nptel.ac.in/courses/106106231**
4.

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

Assessment Type

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

- Quiz
- End Term Exam
- Self-Learning

Course Code:CSE 7300	**Course Title: Capstone Project** **Type of Course:**	**L- T-P- C**	0	0	0	10
Version No.	**1.0**					
Course Pre-requisites	**Knowledge and Skills related to all the courses studied in previous semesters.**					
Anti-requisites	**NIL**					
Course Description	Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learning techniques.					
Course Outcomes	On successful completion of this course the students shall be able to: 1. Identify problems based on societal /research needs. (Understand) 2. Apply Knowledge and skill to solve societal problems in a group. (Apply) 3. Develop interpersonal skills to work as member of a group or leader. (Apply) 4. Analyze the inferences from available results through theoretical / Experimental / Simulations. (Analyze) 5. Analyze the impact of solutions in societal and environmental context for sustainable development. (Analyze) 6. Improve in written and oral communication. (Create)					

	7. Demonstrate capabilities of self-learning in a group, which leads to lifelong learning. (Understand)

Course Code: MAT2064	**Course Title: Numerical Computing** **Type of Course:1] School Core**	**L-T- P-C**	**3**	**0**	**0**	**3**
Version No.		1.0				
Course Pre-requisites		Calculus, Linear Algebra, Differential Equations				
Anti-requisites		**NIL**				
Course Description		The course explores mathematical techniques used to approximate solutions to complex problems that are difficult to solve analytically, often utilizing computers to perform calculations, including methods for root finding, interpolation, numerical differentiation and integration, solving systems of linear equations, and approximating solutions to differential equations, with applications across various scientific and engineering fields. It focuses on understanding the theoretical basis behind these methods, their implementation in programming languages, and analyzing their accuracy and stability.				
Course Objective		The objective of the course is to equip students with understanding and ability to apply various numerical techniques to approximate solutions to complex mathematical problems that are difficult or impossible to solve analytically, particularly focusing on areas like solving systems of equations, finding roots of functions, interpolation, numerical differentiation, and integration, often utilizing computational tools to implement these methods.				
Course Out Comes		On successful completion of the course the students shall be able to: CO1 - Calculate errors induced in the values by truncation of a series expansion. CO2 - Demonstrate the applications of numerical methods to find the roots of polynomial equations and eigen values of real symmetric matrices. CO3 - Apply the knowledge of numerical methods in modelling of various physical and engineering phenomena. CO4 - Apply various numerical methods for solving linear Ordinary & Partial differential equations arising in engineering field.				
Course Content:						
Module 1	**Solution of Linear Systems of Equation**		**(12 Classes)**			
Numerical Computation: Motivation and Objectives, Number Representation, Machine Precision, Round-of Error, Truncation Error, Random Number Generation. Solution of algebraic and transcendental equations: Various types of errors - Bisection method, Regula-Falsi method, Newton-Raphson method, Graffe's method - Bairstow's method - Newton's method for solving f(x,y) = 0 and g(x,y) = 0, secant method, Fixed point iteration method, Solution of linear system of equations, Gauss elimination method, Pivoting, Gauss Jordan method, Iterative methods of Gauss Jacobi and Gauss Seidel, Sufficient conditions for convergence - LU decomposition method, Eigenvalues of a matrix by Power method and Jacobi's method for symmetric matrices.						
Module 2	**Interpolation and Approximation**	**Assignment**	**(8 Classes)**			

<table>
<tr><td colspan="4">Interpolation with equal intervals, Newton’s forward and backward difference formulae, Interpolation with unequal intervals, Lagrange’s interpolation, Newton’s divided difference interpolation, Cubic Splines, Difference operators and relations.</td></tr>
<tr><td><b>Module 3</b></td><td><b>Numerical Differentiation and Integration</b></td><td></td><td><b>(10 Classes)</b></td></tr>
<tr><td colspan="4">Numerical differentiation, Approximation of derivatives using interpolation polynomials, Numerical integration using Trapezoidal rule, Simpson’s one-third rule, Simpson’s three-eighth rule, Weddle’s rule, Romberg’s Method, Two point and three point Gaussian quadrature formulae, Evaluation of double integrals by Trapezoidal rule and Simpson’s one-third rule</td></tr>
<tr><td><b>Module 4</b></td><td><b>Initial & Boundary Value Problems for Ordinary & Partial Differential Equations</b></td><td><b>Assignment</b></td><td><b>(15 Classes)</b></td></tr>
<tr><td colspan="4">Single step methods — Taylor’s series method, Modified Euler’s method, Fourth order Runge-Kutta method for solving first order equations, Multi step methods, Milne’s and Adams, Bash forth predictor corrector methods for solving first order equations.
Finite difference methods for solving second order, two-point linear boundary value problems, Finite difference techniques for the solution of two-dimensional Laplace’s and Poisson’s equations on rectangular domain, One-dimensional heat flow equation by explicit and implicit (Crank Nicholson) methods, One-dimensional wave equation by explicit method.</td></tr>
<tr><td colspan="4">Targeted Application & Tools that can be used:
The contents of this course has direct applications in most of the core engineering courses for problem formulations, Problem Solution and system Design.
Tools Used: Python.</td></tr>
<tr><td colspan="4"><b>Assignment:</b></td></tr>
<tr><td colspan="4"><b>1.</b> 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.</td></tr>
<tr><td colspan="4"><b>Text Book</b>
1. C.F.Gerald and P.O.Wheatley", Applied Numerical Analysis", McGraw-Hill, 1981.
2. Cheneg and Kincaid, "Introduction to Numerical Computing", Tata McGraw-Hill, 1998.</td></tr>
<tr><td colspan="4"><b>References:</b>
1. SRK Iyengar & RK Jain, Numerical Methods, New Age Internationals.
2. Erwin Kreyzig, Advanced Engineering Mathematics, John Wiley and sons, Inc.10th Edition
3. B. S. Grewal (2017), Higher Engineering Mathematics by, 44th Edition, Khanna Publishers.

<b>E-resources/ Web links:</b>
1. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_135224
2. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_141727
3. https://presiuniv.knimbus.com/user#/viewDetail?searchResultType=ECATALOGUE_BASED&unique_id=EBSCO95_30102024_217628
4. http://.ac.in/courses.php?disciplineID=111
5. http://www.class-central.com/subject/math(MOOCs)
6. http://academicearth.org/
7. https://www.math.hkust.edu.hk/~maqian/ma006_0607F.html
8. https://www.scu.edu.au/study-at-scu/units/math1005/2022/</td></tr>
<tr><td colspan="4"><b>Topics relevant to SKILL DEVELOPMENT:</b>   The course focuses on the concepts of calculus</td></tr>
</table>

and differential equation with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem solving**.** This is attained through the assessment component mentioned in course handout.

Course Code: CBC3400
Course Title: Cryptography and Security in Blockchain
L:T:P:C – 3:0:0:3
Prerequisite: 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 Objectives

— 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

CO1 (Understand): Explain the role of cryptographic primitives in securing blockchain data and identity
CO2 (Analyze): Evaluate consensus protocols and blockchain attack surfaces
CO3 (Apply): Implement and test cryptographic algorithms and secure wallet mechanisms
CO4 (Apply): Identify and mitigate security flaws in smart contracts and blockchain systems

Course Content (45 Hours Total)

Module 1: Cryptographic Foundations – 11 Sessions (Understand)

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 – 11 Sessions (Analyze)
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 – 11 Sessions (Apply)
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 – 12 Sessions (Apply)
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)

Textbooks
T1: Kelsey Hightower et al., *Blockchain Security and Privacy*, Wiley, **2023**
T2: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **4th Edition, 2023**

Reference Books
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**

Web Resources
W1: https://cryptozombies.io
W2: https://soliditylang.org/security
W3: https://ethereum.org/en/developers/docs/security
W4: https://z.cash/technology
W5: https://slither.readthedocs.io

Course Code: CBC3401
Course Title: Crypto Trading Strategies & Risk Management
L:T:P:C – 3:0:0:3
Prerequisite: 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 Objectives

— 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

CO1 (Understand): Explain cryptocurrency market structures, instruments, and volatility characteristics
CO2 (Analyze): Evaluate and compare crypto trading strategies and their performance indicators
CO3 (Apply): Apply risk management techniques and portfolio diversification methods
CO4 (Apply): Use tools to implement, simulate, or automate crypto trading strategies

Course Content (45 Hours Total)

Module 1: Cryptocurrency Markets & Exchanges – 11 Sessions (Understand)
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 – 11 Sessions (Analyze)
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 – 11 Sessions (Apply)
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 – 12 Sessions (Apply)

Trading bots and scripting basics (Python, Pionex), Backtesting tools (TradingView, CryptoCompare), Trading APIs (Binance, Coinbase), Regulatory risks, KYC/AML, Taxation, Real-world strategy simulation

Textbooks
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**

Reference Books
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**

Web Resources
W1: https://www.tradingview.com
W2: https://coinmarketcap.com
W3: https://pionex.com
W4: https://academy.binance.com
W5: https://cryptoquant.com

Course Code: CBC3402
Course Title: Bitcoin and Ethereum Protocols
L:T:P:C – 3:0:0:3
Prerequisite: 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 Objectives
— 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
CO1 (Understand): Describe the technical components and operation of Bitcoin and Ethereum
CO2 (Analyze): Compare consensus mechanisms and data structures used in both platforms
CO3 (Apply): Trace transaction flow and analyze smart contract execution
CO4 (Apply): Evaluate scaling solutions, forks, and governance in decentralized networks

Course Content (45 Hours Total)

Module 1: Bitcoin Protocol Internals – 11 Sessions (Understand)
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 – 11 Sessions (Analyze)
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 – 11 Sessions (Apply)
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 – 12 Sessions (Apply)
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

Textbooks
T1: Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, **2nd Edition, 2023**
T2: Andreas M. Antonopoulos & Gavin Wood, *Mastering Ethereum*, O'Reilly Media, **2023**

Reference Books
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**

Web Resources
W1: https://bitcoin.org/en/developer-documentation
W2: https://ethereum.org/en/developers/docs
W3: https://soliditylang.org
W4: https://github.com/ethereum/EIPs
W5: https://bitcoin.stackexchange.com

Course Code: CBC3403
Course Title: Blockchain for Digital Identity Management
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description

This course explores how **blockchain technology** is revolutionizing **digital identity management** by enabling decentralized, secure, and privacy-preserving solutions. Topics include identity lifecycle, verifiable credentials, decentralized identifiers (DIDs), self-sovereign identity (SSI), and regulatory compliance. Real-world platforms and government use cases are also covered.

Course Objectives

— 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

CO1 (Understand): Describe the principles and components of blockchain-based digital identity systems
CO2 (Analyze): Compare centralized and decentralized identity models and their implications
CO3 (Apply): Design verifiable credentials and DIDs using standard protocols
CO4 (Apply): Evaluate use cases and compliance aspects of blockchain-enabled identity frameworks

Course Content (45 Hours Total)

Module 1: Digital Identity Concepts & Challenges – 11 Sessions (Understand)
Digital identity basics, Authentication vs authorization, Centralized identity systems, Federated login issues, Identity theft and fraud, Introduction to identity on blockchain

Module 2: Decentralized Identity & Standards – 11 Sessions (Analyze)
Decentralized Identifiers (DIDs), Verifiable Credentials (VCs), Self-Sovereign Identity (SSI), DIDComm, W3C standards, Privacy-preserving principles (ZKP, selective disclosure)

Module 3: Platforms & Architecture – 11 Sessions (Apply)
Sovrin, uPort, Hyperledger Indy, Aries, Trust over IP (ToIP), Blockchain identity layers, Wallets and credential holders, Issuer-verifier-holder architecture

Module 4: Implementation & Regulatory Compliance – 12 Sessions (Apply)
Smart contract-enabled identity, GDPR compliance, KYC/AML on blockchain, Identity in banking and e-governance, Cross-border identity solutions, Use cases: Aadhaar on blockchain, eIDAS, decentralized healthcare ID

Textbooks
T1: Alex Preukschat & Drummond Reed, *Self-Sovereign Identity*, Manning Publications, **2023**
T2: Mearian Lucas, *Blockchain for Digital Identity*, BPB Publications, **2022**

Reference Books
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

Web Resources
W1: https://www.w3.org/TR/did-core/
W2: https://identity.foundation
W3: https://www.hyperledger.org/use/hyperledger-indy
W4: https://www.sovrin.org
W5: https://trustoverip.org

Course Code: CBC3404
Course Title: Cryptocurrency Wallet Development
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description

This course focuses on the **design, development, and security of cryptocurrency wallets**, which are essential tools for interacting with blockchain networks. Students will learn about wallet types, key management, transaction signing, wallet APIs, and user interface design, along with best practices for developing secure and user-friendly crypto wallets.

Course Objectives
— 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
CO1 (Understand): Explain the architecture and functionality of cryptocurrency wallets
CO2 (Analyze): Compare wallet types and evaluate their security mechanisms
CO3 (Apply): Design and develop wallets with secure key storage and transaction capabilities
CO4 (Apply): Integrate wallet features with DApps and blockchain networks using APIs and SDKs

Course Content (45 Hours Total)

Module 1: Wallet Architecture & Key Management – 11 Sessions (Understand)
Wallet types (hot, cold, custodial, non-custodial), Mnemonic phrases and HD wallets (BIP32/39/44), Public-private key pairs, Address generation, Key encryption, Backup and recovery mechanisms

Module 2: Wallet Security & Compliance – 11 Sessions (Analyze)
Threats and attack vectors (phishing, malware, keylogging), Secure enclave and HSM, 2FA and biometric protection, Anti-fraud mechanisms, KYC/AML integration, Regulatory frameworks (FATF, GDPR)

Module 3: Wallet Development – 11 Sessions (Apply)
Bitcoin and Ethereum transaction lifecycle, Signing and broadcasting transactions, Wallet SDKs (Web3.js, ethers.js, bitcoinjs-lib), QR code integration, Multi-currency support, UI/UX for wallet apps

Module 4: DApp Integration & Advanced Features – 12 Sessions (Apply)
Interfacing with smart contracts, MetaMask integration, Web3 wallets,

WalletConnect protocol, Hardware wallet integration (Ledger, Trezor), Lightning wallets, Token swap and staking support

Textbooks
T1: Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, **2nd Edition, 2023**
T2: Chris Dannen, *Introducing Ethereum and Solidity*, Apress, **2022**

Reference Books
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)

Web Resources
W1: https://web3js.readthedocs.io
W2: https://docs.ethers.org
W3: https://bitcoin.org/en/developer-guide
W4: https://developer.metamask.io
W5: https://walletconnect.com

Course Code: CBC3405
Course Title: Blockchain Security & Ethical Hacking
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course focuses on **security challenges and ethical hacking techniques** specific to blockchain systems. It explores vulnerabilities in smart contracts, consensus mechanisms, cryptographic exploits, and network-level attacks. Students will gain hands-on knowledge in penetration testing, auditing, and securing decentralized applications in a responsible and ethical manner.

Course Objectives
— 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

CO1 (Understand): Describe blockchain security models, attack surfaces, and ethical hacking methodologies
CO2 (Analyze): Identify and assess vulnerabilities in blockchain components and smart contracts
CO3 (Apply): Simulate blockchain-specific attacks in a controlled environment
CO4 (Apply): Use auditing tools and secure coding techniques to mitigate real-world blockchain threats

Course Content (45 Hours Total)

Module 1: Blockchain Threat Landscape & Security Models – 11 Sessions (Understand)
Security principles in decentralized systems, Attack surfaces in blockchain, Common vulnerabilities (51% attack, Sybil attack, DDoS), Smart contract security flaws, Wallet security risks, Ethics and legalities in hacking

Module 2: Ethical Hacking & Penetration Testing – 11 Sessions (Analyze)
Reconnaissance and scanning, Blockchain-specific enumeration, Network sniffing, Exploit development for smart contracts, Bug bounty frameworks (Immunefi, HackenProof), Legal compliance and scope setting

Module 3: Secure Development & Mitigation – 11 Sessions (Apply)
Secure coding standards (Solidity), Gas optimization and fallback protection, Reentrancy guards, Rate limiting and throttling, Smart contract auditing lifecycle, Continuous security integration (DevSecOps)

Module 4: Tools & Real-World Case Studies – 12 Sessions (Apply)
Tools: Mythril, Slither, Hardhat, Truffle Security, Wireshark, Metasploit, Analysis of DAO hack, Poly Network breach, DeFi protocol exploits, Wallet draining scams, Building a secure DApp from scratch

Textbooks
T1: Rajneesh Gupta, *Blockchain Security and Ethical Hacking*, BPB Publications, **2023**
T2: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **4th Edition, 2023**

Reference Books
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**

Web Resources
W1: https://ethereum.org/en/developers/docs/security
W2: https://consensys.github.io/smart-contract-best-practices/
W3: https://cryptozombies.io
W4: https://immunefi.com/learn
W5: https://github.com/crytic/slither

Course Code: CBC3406
Course Title: Introduction to Artificial Intelligence in Blockchain
L:T:P:C – 3:0:0:3
Prerequisite: 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 Objectives

— 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

CO1 (Understand): Explain the fundamentals of AI and blockchain and their convergence
CO2 (Analyze): Evaluate use cases and architectures of AI-powered blockchain systems
CO3 (Apply): Integrate AI models into smart contracts and decentralized applications
CO4 (Apply): Examine the challenges and ethical concerns of deploying AI on the blockchain

Course Content (45 Hours Total)

Module 1: Fundamentals of AI and Blockchain – 11 Sessions (Understand)
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 – 11 Sessions (Analyze)
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 – 11 Sessions (Apply)
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 – 12 Sessions (Apply)
Data privacy, GDPR, Explainable AI (XAI) on blockchain, Resource constraints, Model verification and updates, Regulatory landscape, Ethical implications, Roadmap to autonomous decentralized AI

Textbooks
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**

Reference Books
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**

Web Resources
W1: https://ai.google
W2: https://ethereum.org/en/developers/docs/oracles
W3: https://deeplearning.ai
W4: https://oceanprotocol.com
W5: https://h2020bonseyes.eu

Course Code: CBC3407
Course Title: Machine Learning for Cyber Threat Detection
L:T:P:C – 3:0:0:3
Prerequisite: 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 Objectives

— 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

CO1 (Understand): Explain the types of cyber threats and the role of machine learning in threat detection
CO2 (Analyze): Evaluate datasets and select appropriate ML models for different security tasks
CO3 (Apply): Build and train ML models for intrusion detection and malware classification
CO4 (Apply): Implement real-time threat detection and evaluate model performance

Course Content (45 Hours Total)

Module 1: Introduction to Cybersecurity and Threat Types – 11 Sessions (Understand)

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 – 11 Sessions (Analyze)

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 – 11 Sessions (Apply)

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 – 12 Sessions (Apply)

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

Textbooks
T1: Emmanuel Tsukerman, *Machine Learning for Cybersecurity Cookbook*, Packt Publishing, **2023**
T2: Clarence Chio & David Freeman, *Machine Learning and Security*, O'Reilly Media, **2023**

Reference Books
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**

Web Resources
W1: https://www.kaggle.com/datasets
W2: https://www.unb.ca/cic/datasets
W3: https://github.com/nu11secur1ty
W4: https://scikit-learn.org/stable/
W5: https://cybersecurity.att.com

Course Code: CBC3408
Course Title: AI-Powered Fraud Detection in Blockchain
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores the application of **Artificial Intelligence (AI)** for detecting **fraudulent activities** in blockchain-based systems. It covers anomaly detection, behavioral analysis, graph-based fraud detection, and ML techniques to identify suspicious patterns in transactions, smart contracts, and decentralized applications. Real-world case studies and tools for fraud prevention are also included.

Course Objectives
— Understand the nature of fraud in blockchain networks and DeFi systems
— Explore AI and ML techniques for fraud detection
— Apply anomaly detection and classification models on blockchain datasets

— Learn to deploy real-time fraud detection pipelines and interpret alerts
Course Outcomes **CO1 (Understand):** Describe types of fraud in blockchain systems and the role of AI in detecting them **CO2 (Analyze):** Evaluate transaction patterns using AI algorithms to identify anomalies **CO3 (Apply):** Implement supervised and unsupervised models for fraud detection in smart contracts and wallets **CO4 (Apply):** Build and test AI-driven fraud analytics dashboards for real-time monitoring
Course Content (45 Hours Total)
Module 1: Blockchain Fraud & Risk Landscape – 11 Sessions (Understand) Types of fraud: double-spending, Sybil attacks, wash trading, flash loan exploits, rug pulls, Ponzi schemes in DeFi, Anti-fraud mechanisms, AML/KYC challenges, Role of explainable AI (XAI)
Module 2: ML Techniques for Fraud Detection – 11 Sessions (Analyze) Supervised learning (logistic regression, decision trees, SVM), Unsupervised methods (k-means, Isolation Forest, PCA), Graph-based approaches (Graph Neural Networks), Transaction feature engineering, Datasets (EtherScan, Elliptic, AMLSim)
Module 3: Smart Contract & Wallet Behavior Analysis – 11 Sessions (Apply) Contract flow analysis, Malicious patterns in Solidity, Detecting honeypots and backdoors, Wallet profiling, Time-series analysis of transfers, API integration for behavioral monitoring
Module 4: Tools, Visualization & Real-Time Analytics – 12 Sessions (Apply) Building fraud dashboards with Kibana/Grafana, Stream analytics for fraud signals, Alerting engines, Case studies (DAO exploit, PolyNetwork hack), Integration with Chainalysis, CipherTrace, AML Bot APIs
Textbooks **T1:** M. Rezaul Karim, *Artificial Intelligence for Blockchain*, Packt Publishing, **2023** **T2:** Sudeep Tanwar et al., *Blockchain and AI for Cybersecurity and Privacy*, Springer, **2022**
Reference Books

R1: Thomas Holt, *Cybercrime Through the Blockchain*, Routledge, **2022**
R2: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **2023**
R3: Ashish Mishra, *AI in Finance and Blockchain*, Wiley, **2022**
R4: Yulia Timofeeva, *Detecting Fraud and Financial Crime Using Machine Learning*, Springer, **2023**

Web Resources
W1: https://www.chainalysis.com
W2: https://ciphertrace.com
W3: https://www.amlbot.com
W4: https://etherscan.io
W5: https://elliptic.co/resources

Course Code: CBC3409
Course Title: Optimizing Blockchain Networks with AI
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description

This course focuses on leveraging **Artificial Intelligence (AI)** to enhance the **performance, scalability, and security of blockchain networks**. It explores optimization techniques for transaction throughput, consensus efficiency, node management, and predictive maintenance using AI models. The course emphasizes both theoretical foundations and practical applications of AI in blockchain infrastructure.

Course Objectives

— Understand performance challenges in blockchain networks
— Explore AI techniques for optimizing consensus, throughput, and latency
— Learn predictive models for network behavior and threat detection
— Apply intelligent algorithms to improve scalability and resource management

Course Outcomes

CO1 (Understand): Explain the performance and scalability limitations of blockchain networks
CO2 (Analyze): Evaluate the role of AI in enhancing consensus mechanisms and throughput
CO3 (Apply): Design and test AI models to predict congestion, detect anomalies, and optimize node behavior
CO4 (Apply): Integrate AI algorithms for energy efficiency, resource allocation, and real-time network tuning

Course Content (45 Hours Total)

Module 1: Blockchain Network Architecture & Bottlenecks – 11 Sessions (Understand)
Node types and network topology, Transaction processing pipeline, Consensus protocol overview (PoW, PoS, PBFT), Throughput and latency issues, Scalability trilemma, Gas fee dynamics

Module 2: AI Techniques for Network Optimization – 11 Sessions (Analyze)
Reinforcement Learning for block size tuning, Neural networks for congestion prediction, Anomaly detection in peer behavior, AI-assisted block propagation, ML models for dynamic fee estimation

Module 3: Resource & Energy Optimization – 11 Sessions (Apply)
AI for miner/validator selection, Load balancing, Smart node clustering, Energy-efficient scheduling, Predictive fault detection in blockchain nodes, Real-time node health monitoring

Module 4: Intelligent Consensus & Future Directions – 12 Sessions (Apply)
AI-enhanced consensus models, Swarm intelligence, Federated learning for decentralized AI, Hybrid optimization architectures (AI + DLT), Use cases in DeFi, supply chain, and CBDC networks

Textbooks
T1: Sudeep Tanwar et al., *Blockchain with AI and Machine Learning*, Springer, **2023**
T2: M. Rezaul Karim, *Artificial Intelligence for Blockchain*, Packt Publishing, **2023**

Reference Books
R1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **4th Edition, 2023**
R2: Zheng Zibin, *Blockchain Technologies for AI Applications*, Springer, **2022**
R3: Md. Sadek Ferdous, *AI and Blockchain for Smart Systems*, CRC Press, **2022**
R4: Qusay H. Mahmoud, *Cognitive Blockchain*, Springer, **2021**

Web Resources
W1: https://ethereum.org/en/developers/docs/scaling
W2: https://deeplearning.ai
W3: https://github.com/OpenMined
W4: https://developer.ibm.com/blogs/blockchain-ai
W5: https://research.ibm.com/blog/ai-blockchain

Course Code: CBC3410
Course Title: Generative AI for Blockchain Applications
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description

This course introduces the integration of **Generative AI** with **Blockchain technology** to develop intelligent, decentralized, and automated applications. It focuses on the use of models like GPT, LLMs, and diffusion networks to enhance smart contracts, autonomous agents, decentralized content creation, NFT generation, and AI transparency using blockchain.

Course Objectives

— Understand the fundamentals of Generative AI and its potential in blockchain ecosystems
— Explore the use of LLMs and generative models for smart contracts, NFTs, and decentralized apps
— Apply decentralized storage and blockchain for provenance and model verification
— Investigate real-world use cases and ethical implications of AI-generated content on-chain

Course Outcomes

CO1 (Understand): Describe the principles of Generative AI and how it enhances blockchain use cases
CO2 (Analyze): Evaluate generative models for decentralized identity, content, and NFTs
CO3 (Apply): Design applications that combine smart contracts with LLM-generated outputs
CO4 (Apply): Implement provenance, audit trails, and ownership models using blockchain for AI-generated assets

Course Content (45 Hours Total)

Module 1: Introduction to Generative AI & Blockchain – 11 Sessions (Understand)

Overview of Generative AI: GPT, GANs, VAEs, Diffusion Models, Blockchain basics: smart contracts, transactions, oracles, Benefits of integrating AI with blockchain (immutability, trust, decentralization)

Module 2: Generative AI for NFTs and Smart Contracts – 11 Sessions (Analyze)

AI-generated NFTs and metadata, Smart contract-driven art/music/token generation, Prompt engineering for blockchain use, Ethereum, IPFS, and Arweave for content storage, Creative commons and licensing on-chain

Module 3: Autonomous Agents & Decentralized AI Apps – 11 Sessions (Apply)
Autonomous AI agents for DeFi and DAO governance, Blockchain-based prompt markets, Agent verification and staking, AI for DAO proposals and decision-making, Combining LLMs with Chainlink oracles

Module 4: Security, Ethics & Use Cases – 12 Sessions (Apply)
AI model validation and IP protection using blockchain, Verifiable credentials, Combatting AI-generated misinformation, Case studies: OpenAI tokens, Alethea AI, Bittensor, Web3 AI marketplaces, Regulatory landscape and ethical concerns

Textbooks
T1: Rex St. John, *Generative AI on Web3: LLMs and Blockchain Integration*, Independently Published, **2023**
T2: M. Rezaul Karim, *Artificial Intelligence and Blockchain for Beginners*, Packt Publishing, **2023**

Reference Books
R1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **2023**
R2: Peter Van Hardenberg, *Understanding Large Language Models*, O'Reilly Media, **2023**
R3: Sudeep Tanwar et al., *Blockchain and AI for Cybersecurity and Privacy*, Springer, **2022**
R4: OpenAI Research Papers & Alethea Whitepapers

Web Resources
W1: https://platform.openai.com/docs
W2: https://ethereum.org/en/developers
W3: https://ipfs.tech
W4: https://alethea.ai
W5: https://bittensor.com

Course Code: CBC3411
Course Title: Quantum Computing & Blockchain-AI Security
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course provides an interdisciplinary overview of **quantum computing** and its implications for the **security of blockchain and AI systems**. It covers quantum algorithms, post-quantum cryptography, quantum-safe blockchain protocols, and the impact of quantum advancements on AI model integrity, smart contracts, and decentralized security mechanisms.

Course Objectives

— Understand the fundamentals of quantum computing and its threat to classical cryptography — Explore quantum-resistant cryptographic techniques and blockchain adaptations — Analyze the vulnerabilities in AI and blockchain systems in a post-quantum world — Apply hybrid security models combining quantum safety, blockchain, and AI
Course Outcomes **CO1 (Understand):** Explain the principles of quantum computing and quantum threats to cryptographic systems **CO2 (Analyze):** Evaluate the impact of quantum algorithms on blockchain consensus and AI model security **CO3 (Apply):** Implement basic post-quantum algorithms and analyze their integration in blockchain **CO4 (Apply):** Propose secure architectures combining AI, blockchain, and quantum-safe techniques
Course Content (45 Hours Total)
Module 1: Foundations of Quantum Computing – 11 Sessions (Understand) Qubits, superposition, entanglement, quantum gates and circuits, Quantum parallelism, Quantum vs classical models, Introduction to quantum programming (Qiskit, Cirq)
Module 2: Quantum Threats to Blockchain and AI – 11 Sessions (Analyze) Shor's algorithm, Grover's algorithm, Breaking RSA/ECC, Threats to Bitcoin and Ethereum, Risks to AI model integrity, Quantum attacks on hashes and signatures
Module 3: Post-Quantum Cryptography & Blockchain – 11 Sessions (Apply) Lattice-based, hash-based, code-based cryptography, NIST PQC candidates, Blockchain modifications for quantum resistance, Quantum key distribution (QKD), Hybrid encryption in smart contracts
Module 4: Secure Architectures & Use Cases – 12 Sessions (Apply) Combining AI with quantum-safe blockchains, Use cases: quantum voting, decentralized identity, healthcare AI security, IBM Q and Google Quantum AI tools, Regulatory and ethical considerations
Textbooks

T1: Mikio Nakahara & Tetsuo Ohmi, *Quantum Computing: An Applied Approach*, Springer, **2023**
T2: Mehran Mozaffari, *Post-Quantum Blockchain*, Springer, **2023**

Reference Books
R1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **2023**
R2: Scott Aaronson, *Quantum Computing Since Democritus*, Cambridge University Press, **2022**
R3: Ronald de Wolf, *Quantum Computing: Lecture Notes*, CWI Amsterdam, **2022**
R4: NIST PQC Project Resources and IBM Q Experience Whitepapers

Web Resources
W1: https://qiskit.org
W2: https://quantumai.google
W3: https://csrc.nist.gov/projects/post-quantum-cryptography
W4: https://quantum-computing.ibm.com
W5: https://ethereum.org/en/developers/docs/security/

Course Code: CBC3412
Course Title: Introduction to Decentralized Finance (DeFi)
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description

This course provides a foundational understanding of **Decentralized Finance (DeFi)**—an open financial ecosystem built on blockchain networks. It covers core DeFi primitives such as lending, staking, token exchanges, liquidity pools, stablecoins, DAOs, and yield farming. Emphasis is placed on smart contract mechanics, platform security, and regulatory considerations.

Course Objectives
— Understand the architecture and protocols that power DeFi systems
— Explore key DeFi applications including lending, trading, and insurance
— Learn how smart contracts automate financial services in DeFi
— Analyze risks, governance, and compliance challenges in decentralized finance

Course Outcomes
CO1 (Understand): Explain the foundational concepts, architecture, and drivers of DeFi
CO2 (Analyze): Evaluate DeFi protocols such as DEXs, lending markets, and stablecoins
CO3 (Apply): Simulate DeFi operations such as liquidity provision and staking on testnets

CO4 (Apply): Analyze risks (smart contract, liquidity, governance) and suggest mitigation strategies

Course Content (45 Hours Total)

Module 1: Introduction to DeFi and Blockchain Infrastructure – 11 Sessions (Understand)
What is DeFi?, DeFi vs TradFi, Ethereum and EVM, Smart contracts in DeFi, Token standards (ERC-20, ERC-721), Role of oracles, Introduction to wallets (MetaMask, Ledger)

Module 2: DeFi Applications & Protocols – 11 Sessions (Analyze)
Decentralized exchanges (Uniswap, Curve), Lending/borrowing protocols (Aave, Compound), Stablecoins (DAI, USDC, algorithmic types), Derivatives and synthetic assets, Cross-chain DeFi

Module 3: Participation in DeFi Ecosystem – 11 Sessions (Apply)
Liquidity provision, Yield farming, Staking and governance tokens, DAO voting, Token bonding curves, Farming simulations on testnets (Ropsten, Mumbai)

Module 4: Security, Risks & Regulation – 12 Sessions (Apply)
Smart contract bugs and audits, Flash loan attacks, Liquidity risks, Rug pulls, DeFi insurance, Regulatory overview (FATF, SEC, MiCA), DeFi compliance tools and trends

Textbooks
T1: Camila Russo, *The Infinite Machine*, Harper Business, **2022**
T2: Lasse Clausen, *Mastering DeFi: Decentralized Finance for Everyone*, Independently Published, **2023**

Reference Books
R1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **2023**
R2: Ashwin Ramachandran et al., *The DeFi Stack: A Guide to Decentralized Finance*, Github Release, **2023**
R3: Rainer Böhme, *Cryptoassets and DeFi*, Springer, **2022**
R4: Binance Academy & CoinGecko Research Papers

Web Resources
W1: https://defillama.com
W2: https://ethereum.org/en/defi
W3: https://aave.com
W4: https://uniswap.org
W5: https://dappradar.com

Course Code: CBC3413
Course Title: Blockchain in Financial Services

L:T:P:C – 3:0:0:3
Prerequisite: 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 Objectives
— 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
CO1 (Understand): Explain how blockchain redefines processes in financial services
CO2 (Analyze): Evaluate blockchain use cases across banking, insurance, and investment sectors
CO3 (Apply): Design secure blockchain-based workflows for financial operations
CO4 (Apply): Assess integration, scalability, and regulatory strategies for blockchain solutions in finance

Course Content (45 Hours Total)

Module 1: Blockchain Basics for Finance – 11 Sessions (Understand)
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 – 11 Sessions (Analyze)
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 – 11 Sessions (Apply)
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 – 12 Sessions (Apply)
Security risks (51%, front-running, reentrancy), Regulatory standards (MiCA, FATF), Auditability and traceability, Integration with core banking systems, Blockchain sandboxing, Interoperability (Polkadot, Cosmos)

Textbooks
T1: Antony Welfare, *Commercializing Blockchain: Strategic Applications in the Real World*, Wiley, **2023**
T2: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **4th Edition, 2023**

Reference Books
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

Web Resources
W1: https://r3.com
W2: https://www.weforum.org/projects/blockchain
W3: https://www.bis.org
W4: https://coincenter.org
W5: https://www2.deloitte.com/global/en/pages/financial-services/articles/blockchain-in-banking.html

Course Code: CBC3414
Course Title: Building Decentralized Applications (DApps) for Finance
L:T:P:C – 3:0:0:3
Prerequisite: Nil

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 Objectives

— Understand DApp architecture and smart contract fundamentals
— Learn to build secure, finance-focused DApps using Ethereum and Web3 tools
— Explore UI/UX design, wallet integration, and backend connectivity
— Analyze security, scalability, and deployment challenges in real-world DApps

Course Outcomes
CO1 (Understand): Explain the architecture and components of decentralized financial applications
CO2 (Analyze): Evaluate DApp frameworks, smart contract interactions, and Web3 interfaces
CO3 (Apply): Build and test secure DApps for financial services using blockchain tools
CO4 (Apply): Deploy and optimize DApps with wallet, API, and decentralized storage integration

Course Content (45 Hours Total)

Module 1: DApp Architecture & Financial Use Cases – 11 Sessions (Understand)
Overview of DApps, Differences between traditional apps and DApps, Financial DApp use cases (wallets, lending, insurance), Blockchain platforms (Ethereum, Polygon), DApp lifecycle

Module 2: Smart Contracts for DApps – 11 Sessions (Analyze)
Solidity fundamentals, Contract structure, Inheritance and modifiers, Events and data flow, Interaction with oracles, Audit practices, Gas optimization, Contract upgradeability

Module 3: DApp Frontend, Wallet & Web3 Integration – 11 Sessions (Apply)
Web3.js/ethers.js setup, MetaMask and WalletConnect integration, DApp UI design principles, React.js frontend, Reading/writing contract data from frontend, Decentralized authentication

Module 4: Deployment, Storage & Real-World DApps – 12 Sessions (Apply)
IPFS and Arweave for decentralized storage, Hardhat/Truffle for testing and deployment, DeFi API integration (Uniswap, Aave), Use cases: DApp lending platform, staking, NFT marketplace, Compliance and user privacy

Textbooks
T1: Nader Dabit, *Full Stack DApp Development*, O'Reilly Media, **2023**
T2: David Hoover, *Decentralized Applications: Harnessing Ethereum & Solidity*, Manning, **2023**

Reference Books

R1: Andreas M. Antonopoulos, *Mastering Ethereum*, O'Reilly Media, **2023**

R2: Greg Lim, *Beginning Ethereum and Solidity Smart Contracts*, Packt, **2022**

R3: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **2023**

R4: OpenZeppelin Docs, Ethereum Dev Community Resources

Web Resources

W1: https://ethereum.org/en/developers

W2: https://docs.metamask.io

W3: https://web3js.readthedocs.io

W4: https://hardhat.org

W5: https://openzeppelin.com

Course Code: CBC3415

Course Title: Smart Contracts for Financial Products

L:T:P:C – 3:0:0:3

Prerequisite: Nil

Course Description

This course focuses on the **design, development, and deployment of smart contracts** specifically for **financial products and services**. It introduces smart contract logic, digital agreements, automated settlements, and programmable finance using platforms like Ethereum and Hyperledger. Use cases include insurance, lending, escrow, tokenized assets, and derivatives.

Course Objectives

— Understand the fundamentals and architecture of smart contracts

— Learn how to model financial instruments using code-based contracts

— Build and deploy secure smart contracts for DeFi and traditional finance applications

— Explore risks, audit strategies, and compliance issues in financial smart contracts

Course Outcomes

CO1 (Understand): Explain the architecture, lifecycle, and legal implications of smart contracts in finance

CO2 (Analyze): Compare financial product types and model them using Solidity or other smart contract languages

CO3 (Apply): Build, test, and deploy smart contracts for loans, insurance, and tokenized securities

CO4 (Apply): Audit and optimize smart contracts for security, gas usage, and regulatory alignment

Course Content (45 Hours Total)

Module 1: Introduction to Smart Contracts – 11 Sessions (Understand)
Concept and origin, Contract lifecycle, Ethereum and EVM, Benefits and risks in finance, Platforms (Solidity, Vyper, DAML), Role of oracles and event logs

Module 2: Modeling Financial Instruments – 11 Sessions (Analyze)
Smart contracts for loans (collateralized, flash), Escrow services, Insurance contracts (parametric, peer-to-peer), Structured products, Tokenized bonds and equities

Module 3: Development, Testing & Deployment – 11 Sessions (Apply)
Solidity contract structure, Events and modifiers, Testing with Truffle/Hardhat, Gas optimization, Upgradable contracts using proxies, Remix and blockchain testnets

Module 4: Security, Audits & Compliance – 12 Sessions (Apply)
Smart contract vulnerabilities (overflow, reentrancy), Audit frameworks (Mythril, Slither), Access control (OpenZeppelin), Compliance considerations (MiCA, SEC), Real-world failures (DAO, Wormhole), Best practices

Textbooks
T1: Ritesh Modi, *Building Smart Contracts and DApps on Ethereum*, BPB Publications, **2023**
T2: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **4th Edition, 2023**

Reference Books
R1: Andreas M. Antonopoulos & Gavin Wood, *Mastering Ethereum*, O'Reilly Media, **2023**
R2: Chris Dannen, *Introducing Ethereum and Solidity*, Apress, **2022**
R3: S. Seijas et al., *Financial Smart Contracts and the Future of Banking*, Springer, **2021**
R4: OpenZeppelin Docs and Solidity Audit Guides

Web Resources
W1: https://soliditylang.org
W2: https://remix.ethereum.org
W3: https://openzeppelin.com
W4: https://consensys.net
W5: https://chain.link

Course Code: CBC3416
Course Title: Decentralized Autonomous Organizations and Risk Management in Finance
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description

This course explores **Decentralized Autonomous Organizations (DAOs)** and their application in **financial risk management**. It delves into governance models, treasury operations, legal status, and smart contract-driven automation of financial processes. It also covers various risk types—operational, smart contract, and market—and their mitigation strategies in decentralized environments.

Course Objectives

— Understand the structure and operational logic of DAOs
— Explore DAO use cases in investment, lending, and decentralized governance
— Learn how risk is modeled and managed within decentralized financial systems
— Analyze legal, regulatory, and technical frameworks for DAO-based finance

Course Outcomes

CO1 (Understand): Describe the architecture, governance, and funding mechanisms of DAOs
CO2 (Analyze): Evaluate DAO-based models for financial services and risk exposure
CO3 (Apply): Build and simulate DAO smart contracts and voting systems
CO4 (Apply): Develop risk mitigation strategies for DAO-managed financial operations

Course Content (45 Hours Total)

Module 1: DAO Fundamentals and Governance – 11 Sessions (Understand)

Definition and DAO structure, Key smart contracts in DAOs, Voting mechanisms (token-based, quadratic), Treasury management, Governance tokens, DAO platforms (Aragon, DAOstack)

Module 2: DAO Use Cases in Finance – 11 Sessions (Analyze)

DAO-based lending platforms, Investment DAOs, Yield aggregators, DAO-managed stablecoins, DAO for insurance and asset management, Real-world case studies (MakerDAO, Yearn Finance, ConstitutionDAO)

Module 3: Risk Management in DAO Ecosystems – 11 Sessions (Apply)
Types of risks: protocol, liquidity, operational, regulatory, Risk scoring systems, DAO treasury risk modeling, Smart contract vulnerabilities, Flash loan attacks, DAO governance attacks

Module 4: Legal Frameworks and Compliance – 12 Sessions (Apply)
Legal entity status of DAOs, Jurisdictions (Wyoming, Malta, Liechtenstein), DAO compliance toolkits, Auditing and reporting in DAO finance, Security vs utility tokens, Regulatory overlaps (SEC, MiCA, FATF)

Textbooks
T1: Oriol Caudevilla et al., *Decentralized Finance and DAOs*, Springer, **2023**
T2: Linda Xie, *A Beginner's Guide to DAOs*, Ethereum Foundation Series, **2023**

Reference Books
R1: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **2023**
R2: Nathan Schneider, *Exit to Community: DAO Governance and the Future of Finance*, Yale University Press, **2022**
R3: Primavera De Filippi et al., *Blockchain and the Law: The Rule of Code*, Harvard Press, **2022**
R4: DAO Research Collective Whitepapers & LexDAO Legal Templates

Web Resources
W1: https://daostack.io
W2: https://aragon.org
W3: https://ethereum.org/en/dao
W4: https://makerdao.com
W5: https://lexdao.substack.com

Course Code: CBC3417
Course Title: Regulatory and Compliance Challenges in DeFi
L:T:P:C – 3:0:0:3
Prerequisite: Nil

Course Description
This course explores the **regulatory, legal, and compliance frameworks** surrounding **Decentralized Finance (DeFi)**. It analyzes how DeFi challenges traditional financial regulation and examines global responses, such as MiCA, FATF, SEC guidance, and KYC/AML implementations. The course includes real-world case studies, compliance tools, and ethical considerations for DeFi innovation.

Course Objectives

— Understand the regulatory landscape for DeFi and decentralized platforms
— Examine compliance obligations related to KYC, AML, and taxation
— Analyze global approaches to DeFi oversight, classification of tokens, and enforcement trends
— Explore solutions for integrating compliance into decentralized systems

Course Outcomes

CO1 (Understand): Identify the key regulatory concerns and frameworks applicable to DeFi systems
CO2 (Analyze): Evaluate compliance challenges in decentralized lending, trading, and asset issuance
CO3 (Apply): Implement KYC/AML procedures and governance models in DeFi applications
CO4 (Apply): Assess risk and compliance strategies using real-world legal and technical tools

Course Content (45 Hours Total)

Module 1: Introduction to DeFi Regulation – 11 Sessions (Understand)

What is DeFi and why it's difficult to regulate, Regulatory bodies (FATF, SEC, EU, BIS), Principles-based vs rules-based regulation, Legal identity and anonymity, Public vs permissioned protocols

Module 2: Global Regulatory Frameworks – 11 Sessions (Analyze)

MiCA (EU), FinCEN (USA), MAS (Singapore), FCA (UK), Taxation of crypto assets, Classifying tokens (security, utility, hybrid), Regulatory sandbox models, DAO regulation

Module 3: Compliance Integration in DeFi Platforms – 11 Sessions (Apply)

On-chain KYC/AML, Decentralized identity (DID, VC), Risk scoring and monitoring tools, AML compliance engines (Chainalysis, TRM Labs), Oracle-based compliance feeds, Smart contract-based compliance controls

Module 4: Case Studies, Ethics & Emerging Trends – 12 Sessions (Apply)

Notable enforcement actions (SEC vs Ripple, Tornado Cash sanctions), DeFi rug pulls and litigation, Data privacy (GDPR in DeFi), Self-regulatory organizations (SROs), Compliance-by-design models, Future of regulated DeFi

Textbooks

T1: Jason Gottlieb, *Crypto Regulation: Navigating Compliance in the Decentralized World*, Wiley, **2023**

T2: Christopher Giancarlo, *CryptoDad: The Fight for the Future of Money*, Wiley, **2023**

Reference Books

R1: Primavera De Filippi & Aaron Wright, *Blockchain and the Law*, Harvard University Press, **2022**

R2: Imran Bashir, *Mastering Blockchain*, Packt Publishing, **2023**

R3: World Economic Forum, *DeFi Policy Toolkit*, WEF, **2022**

R4: Global Financial Stability Reports – IMF, BIS, FATF Recommendations

Web Resources

W1: https://www.fatf-gafi.org

W2: https://www.sec.gov/spotlight/cybersecurity

W3: https://www.european-union.europa.eu

W4: https://www.chainalysis.com

W5: https://trmlabs.com

Course Code: **CSE2511**	**Course Title: Data Analytics** **Type of Course: Program Core**	**L-T-P-C**	2	0	0	2
Version No.	**1.0**					
Course Pre-requisites	**MAT1003 Applied Statistics**					
Anti-requisites	**NIL**					
Course Description	**Fundamentals of Data Analytics is designed for inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, and supports in decision-making. The course begins by covering Data extraction, pre-processing, and transformation. It delivers the basic statistics and taught in an intuitive way to analysis the data. This course will help the students to apply the knowledge on data analysis to a wide range of applications.**					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Fundamentals of Data Analytics** and attain **SKILL DEVELOPMENT** through **PROBLEM SOLVING** Methodologies.					
Course Out Comes	**On successful completion of this course, the students shall be able to:** **CO1**:Describe different types of data and variables. **CO2**: Explain data using appropriate statistical methods. **CO3**: Demonstrate the collection, processing and analysis of data for any given application and illustrate various charts using visualization methods. **CO4**: Apply the Data Analysis techniques by R Programming					
Course Content:						
Module 1	**Introduction to Data Analysis- CO1**	Assignment	Data Collection, data analysis, Programming	**06 classes**		
Topics: Introducing Data, overview of data analysis: Data in the Real World, Data vs. Information, The Many “Vs” of Data, Structured Data and Unstructured Data, Types of Data, Data Analysis Defined, Types of Variables, Central Tendency of Data, Scales of Data, Sources of Data. Data preparation. **R Studio:** Base R-R Studio IDE-Introduction to R Projects and R Markdown. Basic R: R as a calculator-Scripts and Comments-R Variables. Data I/O: Working Directories-Importing Data Exporting Data-More ways to save-Data I/O in Base R.						
Module 2	**Data Analysis and Visualization-CO2**	Case studies	Programming	**10 classes**		
Topics: Data Summarization: One Quantitative and Categorical Variable. Data Classes: One Dimensional Data Classes-Data Frames and Matrices-Lists. Data Cleaning: Dealing with						

Missing Data-Strings and Recoding Variables. Manipulating Data in R: Reshaping Data-Merging Datasets. Data Visualizations: Plotting with ggplot2- Plotting with Base R

Module 3	**Statistical Analysis -CO3**	Case studies	R programming	**7 classes**

Topics: Proportion tests-Chi squared test-Fisher exact test-Correlation-T test-Wilcoxon Rank sum tests-Wilcoxon signed rank test- one-way ANOVA test- Kruskal Wallis test

Module 4	Predictive Analysis-CO4	Case studies	Programming	**7 classes**

Topics: Linear least-squares – implementation – the goodness of fit – testing a linear model – weighted resampling. Regression using Stats models – multiple regression – nonlinear relationships – logistic regression – estimating parameters – accuracy. Time series analysis – moving averages – missing values – serial correlation – autocorrelation. Introduction to survival analysis

Targeted Application & Tools that can be used:

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

Text Books

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

References

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

Online resources:

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

Topics relevant to development of "FOUNDATION SKILLS":

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

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

Course Code: CSE2512	**Course Title: Data Analytics Lab** **Type of Course: Program Core**	**L-P- C**	0	0	2	1
Version No.	**1.0**					
Course Pre-requisites	**MAT1003 Applied Statistics**					
Anti-requisites	**NIL**					
Course Description	**Fundamentals of Data Analytics is designed for inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, and supports in decision-making. The course begins by covering Data extraction, pre-processing, and transformation. It delivers the basic statistics and taught in an intuitive way to analysis the data. This course will help the students to apply the knowledge on data analysis to a wide range of applications.**					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Fundamentals of Data Analytics** and attain **SKILL DEVELOPMENT** through **PROBLEM SOLVING** Methodologies.					
Course Out Comes	**On successful completion of this course, the students shall be able to:** **CO1**:Describe different types of data and variables. **CO2**: Explain data using appropriate statistical methods. **CO3**: Demonstrate the collection, processing and analysis of data for any given application and illustrate various charts using visualization methods. **CO4**: Apply the Data Analysis techniques by R Programming					
Course Content:						

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

List of Laboratory Tasks:

Experiment No. 1: Introduction to R and RStudio

Level 1: Getting Started with R and RStudio

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

Level 2: Working with RStudio

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

Experiment No. 2: Basic Data Handling in R

Level 1: Data Types and Structures in R

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

Level 2: Data Import and Export

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

Level 3: Exploring Datasets

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

Experiment No. 3: Basic Data structure in R

Level 1: a. Demonstrate a program to join columns and rows in a data frame using cbind() and rbind() in R.

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

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

a. Using mathematical functions on console
b. Write an R script, to create R objects for the calculator application

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

Experiment No. 1: Data Cleaning and Preprocessing

Level 1: Handling Missing Data in R

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

Level 2: Data Transformation in R

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

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

Level 1: Descriptive Statistics

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

Experiment No. 3: Data Visualization with ggplot2

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

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

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

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

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

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

Experiment No 2: Hypothesis – Non-Parametric Test

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

Experiment No 3: Correlation and Covariance

Level 1: Using the iris data set in R

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

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

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

Experiment No 1: Regression Model

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

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

Experiment No. 2: Time Series Analysis in R
Level 1: Demonstrate Timeseries analysis using Time Series Data Library at http://robjhyndman.com/TSDL/.

Targeted Application & Tools that can be used:
Application Area are Decision making in business, health care, financial sector, Medical diagnosis etc.

Text Books

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

References

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

Online resources:

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

Topics relevant to development of "FOUNDATION SKILLS":

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

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

Course Code: CSE7101	**Course Title: Mini Project** **Type of Course:**	**L- T-P- C**	0	0	0	5
Version No.	**1.0**					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain Employability Skills through Experiential Learning techniques.					

Course Outcomes	On successful completion of this course the students shall be able to: 1. Identify the engineering problems related to local, regional, national or global needs. (Understand) 2. Apply appropriate techniques or modern tools for solving the intended problem. (Apply) 3. Design the experiments as per the standards and specifications. (Analyze) 4. Interpret the events and results for meaningful conclusions. (Evaluate) 5. Appraise project findings and communicate effectively through scholarly publications. (Create)