



Object Oriented Programming with JAVA			Credits	4
Course Code	25CSIP320	Course type	IPCC	
Teaching Hours/Week (L:T:P)	3:0:2	Total Hours of Pedagogy L:T:P:TW&SL:TH	42:0:28:50:120	
Type of examination (IPCC)	Theory- CIE+SEE and Lab- CIE only	CIE Marks	50	
SEE Examination type	Theory	SEE Marks	50	
SEE Exam Hours	3	Total Marks	100	
Course outcome (Course Skill Set)				
At the end of the course, the student will be able to:				
1. Demonstrate proficiency in writing simple programs involving branching and looping structures. 2. Construct object-oriented solutions using classes, objects, and methods. 3. Develop programs using inheritance, packages, and interfaces. 4. Make use of exception handling and multithreading in solving complex problems 5. Apply concepts of enumerations, type wrappers and strings in program development. 6. Choose a suitable IDE tool to design, develop, and debug Java programs.				
Module-1			Number of hours: 9	
The History and Evolution of Java: How Java Impacted the Internet, Java’s Magic: The ByteCode Java Buzzwords.				
An Overview of Java: Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords).				
Data Types, Variables, and Arrays: Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays.				
Operators: Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses.				
Control Statements: Java’s Selection Statements (if, The Traditional switch), Iteration Statements (while, do-while, for, The For-Each Version of the for Loop, Local Variable Type Inference in a for Loop, Nested Loops), Jump Statements (Using break, Using continue, return).				
Chapter 1, 2, 3, 4, 5				
Lab Program: 1,2				
Module-2			Number of hours: 8	
Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection.				
Methods and Classes: Overloading Methods, Overloading Constructors, Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes,				
Chapter 6, 7				
Lab Program: 3,4				
Module-3			Number of hours: 9	
Inheritance: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance.				



Packages: Packages, Packages and Member Access, Importing Packages.

Interfaces: Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods.

Chapter 8, 9

Lab Program: 5,6

Module-4

Number of hours: 8

Exceptions: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses.

Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming, and Stopping Threads,

Chapter 10, 11

Lab Program: 7,8

Module-5

Number of hours: 8

Enumerations: Enumerations (Enumeration Fundamentals, The values() and valueOf Methods), Type Wrappers (Character, Boolean, The Numeric Type Wrappers)

String Handling: The String Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion using valueOf(), Changing the case of Characters within a String, Joining Strings.

Chapter 12, 18

Lab Program: 9, 10

Practical Component of IPCC

Number of hours: 28

Programming Experiments (Suggested and are not limited to)

1. Develop a Java program to add TWO matrices of suitable order N.
2. Develop a stack class to hold a maximum of 10 integers with suitable methods. Develop a Java main method to illustrate Stack operations.
3. Create a Java class called Student with the USN, Name, Branch, Phone details as variables within it. Write a Java program to create N Student objects and print the USN, Name, Branch, and Phone of these objects with suitable headings.
4. Develop a Java program to create a class named Shape. Create three sub classes, namely: circle, triangle and square. Each class has two member functions named draw and erase. Demonstrate the concepts of polymorphism by developing appropriate methods, defining member data, and implementing the main program.
5. Develop a Java program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.
6. Develop a Java program to create a package named mypack and import & implement it in a suitable class.

7. Develop a Java program to raise a custom exception (user-defined exception) for DivisionByZero using try, catch, throw and finally.

8. Develop a program to create a class MyThread with a class constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.

9. Develop a Java program using an enumeration called Department containing CSE, ISE, ECE, and MECH. Accept a department name as input and display the selected department using valueOf().

10. Develop a Java program to perform basic string-handling operations on a given string. The program should determine and display the length of the string, convert the string into both uppercase and lowercase forms, and reverse the string to check whether the given string is a palindrome and write **JUnit test cases** to verify the correctness of the string length, case conversion, string reversal, and palindrome-checking operations using suitable test inputs.

11. **Micro Project:** A team of two to three students shall develop a project and submit a brief report for a specific task assigned by the course instructor based on the concepts covered in the syllabus.

Note: Each experiment (1 to 10) will be evaluated for 30 marks (total 300), and the micro-project for 120 marks. Total marks (420) shall be scaled down to 15. Make use of the most recent version of the IDE tool for program development.

Suggested Learning Resources: (Textbook/ Reference Book/ Manuals):

Textbook: Herbert Schildt and Danny Coward, Java: The Complete Reference, 13th Edition, McGraw-Hill, 2024.

Reference Books / Manuals:

1. Cay S. Horstmann, Core Java Volume I – Fundamentals, 14th Edition, Oracle Press, 2026.
2. E. Balagurusamy, Programming with Java, 7th Edition, McGraw-Hill Education, 2024.

Web links and e-Resources:

1. Official Java documentation (Oracle): <https://docs.oracle.com/javase/tutorial/>
2. Modern Java learning resources (Dev Java): <https://dev.java/learn/>
3. Java Tutorial: <https://www.w3schools.com/java/>
4. Programming examples and practice (GeeksforGeeks):
<https://www.geeksforgeeks.org/java/java/>
5. Infosys Springboard Course on Java Programming Fundamentals:
<https://infyspringboard.onwingspan.com/web/en/login>
6. HackerRank platform: <https://www.hackerrank.com/domains/java>
7. Structured video lectures (NPTEL): (<https://nptel.ac.in/courses/106105191/>)



Teaching-Learning Process (Innovative Delivery Methods):

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

1. Flipped Classroom.
2. Problem-Based Learning
3. ICT Enabled Teaching
4. Programming Assignments
5. Tool-Based Learning using Eclipse/IntelliJ IDEA

Assessment Structure:

The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying **50% weightage** (i.e., 50 marks each). The CIE Theory component will be **25 marks** and CIE Practical component will be **25 marks**.

The CIE Theory component consists of IA tests for 25 marks. The CIE Practical component for continuous assessments will be for 15 marks through rubrics and for lab tests will be for 10 marks.

- To qualify and become eligible to appear for SEE, in the **CIE theory component**, a student must score at least **40% of 25 marks, i.e., 10 marks**.
- To qualify and become eligible to appear for SEE, in the **CIE Practical component**, a student must secure a **minimum of 40% of 25 marks, i.e., 10 marks**.
- To pass the **SEE**, a student must secure a **minimum of 35% of 50 marks, i.e., 18 marks**.
- A student is deemed to have **successfully completed the course** if the **combined total of CIE and SEE is at least 40 out of 100 marks**.

CIE for the practical component:

The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report using a defined set of rubrics. Each experiment report can be evaluated for 30 marks. The summation of all the experiments marks to be scaled down to 15 marks.

The laboratory test (duration 03 hours) at the end of the last week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and scaled down to 10 marks. For the laboratory test, the student is required to conduct one experiment (excluding the micro project).

SEE Pattern:

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), should have a mix of topics under that module.
3. The students have to answer 5 full questions, selecting one full question from each module.
4. Marks scored by the student shall be proportionally scaled down to 50 Marks

The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have a CIE component only. Questions mentioned in the SEE paper may include questions from the practical component.

Rubrics for CIE of Practical Component:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Fundamental Knowledge: Understanding the problem statement [C01] [P01, P02]	The student has in depth knowledge of the topics related to the problem. Student is able to completely understand the problem definition.	Student has good knowledge of some of the topics related to problem. Student is able to understand the problem definition.	Student is capable of narrating the answer but not capable to show in depth knowledge and the problem definition.	Student has not understood the concepts partially. Student is able to partially understand the problem definition.	Student has not understood the concepts and the problem definition clearly.
Design of program [C02-6] [P02, P03]	Capable of discussing multiple designs and selecting the best with justification.	Discusses multiple designs but lacks proper selection criteria.	Discusses a single design with merits and demerits.	Explains only the basic design.	Unable to explain the design.
Implementation using Java [C02-6] [P03, P08, P05]	Implements optimal solution using suitable Java constructs and OOP features.	Correct implementation with good explanation.	Correct implementation with partial explanation.	Implementation contains errors.	Unable to implement the solution.
Program debugging and testing with suitable tools [C02-6] [P05, P08]	Compiles, debugs, and tests without errors; demonstrates full understanding.	Corrects errors with good understanding.	Corrects errors with partial understanding.	Corrects errors with assistance.	Unable to debug and test independently.
Results & interpretation /analysis [C01-6] [P04, P05]	Executes for multiple test cases and performs detailed analysis.	Executes correctly for all test cases.	Executes for a few test cases and analyzes results.	Executes but cannot analyze results.	Unable to verify correctness of results.
Demonstration and documentation [C01-6] [P08, P09, P011]	Demonstration and lab record is well-organized, with clear sections. The record is well structured with suitable formatting.	Demonstration and lab record is organized, with clear sections, but some sections are not well-defined. The record is structured with formatting.	Demonstration and lab record lacks clear organization or structure. Some sections are unclear or incomplete. The record is partially structured with formatting.	Demonstration and lab record is poorly organized, with missing or unclear sections. The record is not properly structured with suitable formatting.	Demonstration and lab record is poorly organized, with missing sections. Record not submitted on time. The record is not structured with minimum formatting.

Rubrics for CIE Practical Test:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Fundamental Knowledge (2) [C01] [P01]	The student has well depth knowledge of the topics related to the problem & course.	Student has good knowledge of some of the topics related to problem & course.	Student has average knowledge of some of the topics related to problem & course.	Student is capable of narrating the answer but not capable to show in depth knowledge.	Student has not understood the concepts clearly.
Understanding of problem definition (1) [C02-6] [P02]	Student is able to completely understand the problem definition.	Student is able to understand the problem definition but not clearly.	Student has a basic understanding of the problem definition that is partial or superficial.	Student is able to show minimal or unclear understanding of the problem definition.	Student is not able to understand the problem definition.
Design and Implementation (3) [C02-6] [P03, P05]	Student is capable of design and implementing with best suitable java construct for the given problem definition.	Student is capable of design and implementing with some construct for the given problem definition.	Student is capable of design and implementing the core part of the construct for the given problem definition .	Student is partially capable of design and implementing with some algorithm for the given problem definition.	Student is not capable of design and implementing.
Result & Analysis (2) [C02-6] [P04, P05]	Student is able to run the program on various data inputs and compare the result with proper inference.	Student will be able to run the program on various data inputs and fair knowledge in comparing the result with proper inference.	Student will be able to run the code for few data/datasets and analyze the output.	Student will be able to run the code for few data inputs but not analyze the output.	Student will be not able to run the program and not able to analyze the result.
Communication (Viva voce) (2) [C01-6] [P08, P09]	Good Verbal & nonverbal communicatio n skills with precise and correct terminologies/ answers.	Good verbal Communication skills with precise and correct terminologies/a ns wers.	Average Communicatio n but with precise and correct terminologies/ ans wers.	Average Communication but with imprecise and incorrect terminologies/ans wers.	Poor Communication (Minimal interaction/ans wers)



Term work (TW) and Self-learning (SL) components:

Term work (includes assignments, seminars, micro projects, industrial visits, any other student activities, etc)

Module No.	Topic	Term work (TW) Activity	Resource / Link	No. of Hours / Semester
1	Installation	Installation and use of IDE for Java. Writing simple Java programs to understand Java program development and execution model	-----	04
2	Java Fundamentals : Data Types, Arrays, Operators and Control Statements	Develop a Student Result Analysis System that accepts marks of students, computes total, average, grade, highest marks, and generates a report using arrays, operators, control statements, and command-line arguments.	https://www.geeksforgeeks.org/java/	-do-
3	Classes, Objects, Constructors and Method Overloading	Develop an Employee Management System implementing classes, objects, constructors, constructor/method overloading, encapsulation, access specifiers, static and final members.	https://www.javatpoint.com/java-classes-objects	-do-

SL: Self-Learning, MOOCs, spoken tutorials, online educational resources etc.

Module No.	Self-Learning Activity	No. of Hours / Semester
1	Springboard course on: Java Programming Fundamentals	36
2	Solve problems on HackerRank using Java programming https://www.hackerrank.com/domains/java	10



Continuous Internal Evaluation (CIE)

(i) Theory component

The CIE marks of 25 shall be earmarked for two tests.

The first test shall be conducted after completing two modules of the syllabus and the second one after completing the rest of the three modules. Each test shall be conducted for 25 marks. The average of the two tests scaled down to 25 marks shall constitute the test marks for a maximum of 25 marks.

(ii) Laboratory component

Out of 25 marks, 15 marks shall be assigned for assessment as per the rubrics listed in the course syllabus (30 marks scaled down to 15). The remaining 10 marks shall be based on the practical test conducted by two Internal examiners appointed by the HoD. The allotment of marks shall be as per the rubrics defined for the practical test (50 marks scaled down to 10).

(i) To qualify and become eligible to appear for the SEE of the IC/IPCC, a student shall secure at least 40 % of 25 marks, i.e., 10 marks, in the CIE theory component, and at least 40 % of 25 marks, i.e., 10 marks in the CIE Practical component.

Passing requirement in SEE

For a pass in the Semester End Examination (SEE), a student shall secure a minimum of 35 marks out of 100. For the declaration of a pass grade, the sum of the Continuous Internal Evaluation (CIE) marks (out of 50) marks and the SEE marks scaled down to 50 shall be greater than or equal to 40 marks. If the total (CIE + SEE) is less than 40 marks, the student shall be awarded the grade 'F' (Fail).



Digital Design and Computer Organization			
Course Code	25CSCC330	Scheme	2025
Type of Course	PCC	Semester	3
Teaching Hours/Week (L:T:P)	4:0:0	CIE Marks	50
Total Hours of Pedagogy per semester L /T/P/SL&TW:	56:0:0:64:120	SEE Marks	50
Credits	4	Total Marks	100
Examination type (SEE)	Theory	Exam Hours	03
Course outcome (Course Skill Set) At the end of the course, the student will be able to: 1. Apply Boolean algebra principles to simplify digital logic circuits using logic gates, K-maps, and NAND/NOR implementations. 2. Apply the concepts of combinational logic and sequential logic in implementation of Digital circuits. 3. Describe the organization and operation of computer systems. 4. Illustrate memory and input/output organization techniques, used in development of semiconductor memory systems. 5. Analyze computer architecture components, and pipelining techniques through experimentation and case studies to assess system performance and efficiency.			
Module-1			
Boolean Algebra and Logic Gates: Basic Theorems And Properties of Boolean Algebra, Boolean Functions, Digital Logic Gates, Integrated Circuits. Gate-Level Minimization: Introduction, The Map Method, Four-Variable K-Map, Product of sums simplification, Don't-Care Conditions, NAND and NOR Implementation. Textbook-1: Chapter- 2 (2.4, 2.5, 2.8,2.9), Chapter-3 (3.1, 3.2,3.3,3.4, 3.5, 3.6) Number of Hours: 12			
Module-2			
Combinational Logic: Introduction, Combinational Circuits, Analysis of Combinational Circuits, Design Procedure, Binary Adder – Subtractor, Decoders, Encoders, Multiplexers. Synchronous Sequential Logic: Introduction, Sequential Circuits, Storage Elements: Latches, Flip-Flops. Textbook-1: Chapter- 4 (4.1, 4.2, 4.3, 4.4, 4.5, 4.9, 4.10, 4.11), Chapter-5 (5.1, 5.2, 5.3, 5.4) Number of Hours: 12			
Module-3			
Basic Structure of Computers: Functional Units, Basic Operational Concepts, Bus structure, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Instruction Set: CISC an RISC, Performance Measurement. Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instruction and Instruction sequencing, Addressing Modes. Textbook-2: Chapter -1 (1.2, 1.3, 1.4, 1.6), Chapter -2 (2.2, 2.3, 2.4, 2.5) Number of Hours: 10			



Module-4
Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Direct Memory Access, Buses. The Memory System: Some Basic Concepts, Semiconductor RAM Memories, Read Only Memory, Speed, size and Cost of memory systems. Cache Memories – Mapping Functions. Textbook-2: Chapter – 4 (4.1, 4.2.1, 4.2.2, 4.2.3, 4.4, 4.5), Chapter-5 (5.1, 5.2, 5.3, 5.4, 5.5.1) Number of Hours: 12
Module-5
Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization. Hardwired Control. Pipelining: Basic concepts, Role of Cache memory, and Pipeline Performance. Textbook-2: Chapter -7 (7.1, 7.2, 7.3,7.4), Chapter – 8 (8.1) Number of Hours: 10
Suggested Learning Resources: (Text Book/ Reference Book/ Manuals): <u>Textbooks:</u> 1. M. Morris Mano & Michael D. Ciletti, Digital Design, 6th edition, Pearson Education. 2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, 5th Edition, Tata McGraw Hill. <u>Reference books / Manuals:</u> 1. Charles H Roth and Larry L Kinney, Analog and Digital Electronics, Cengage Learning 2. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015.
Web links and Video Lectures (e-Resources): 1. NPTEL Video Course: Digital Computer Organization by Prof. P. K. Biswas, IIT Kharagpur - https://nptel.ac.in/courses/117105078 2. MIT OpenCourseWare: Computation Structures (6.004) by Prof. Chris Terman, MIT – https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/ 3. Computer Organisation Virtual Lab, IIIT Hyderabad (Virtual Labs) – https://cse11-iiith.vlabs.ac.in/ 4. Infosys Springboard Courses: https://infyspringboard.onwingspan.com/
Teaching-Learning Process (Innovative Delivery Methods): The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes. 1. Flipped Classroom 2. Problem-Based Learning (PBL) 3. Case-Based Teaching 4. Simulation and Virtual Labs 5. Think-Pair-Share: Students solve a Boolean expression individually, discuss with a partner, then present the solution. 6. One-Minute Paper: At the end of each class, students write the most important concept learned and one remaining doubt.



7. **Flipped Classroom:** Students study course content before class, enabling interactive discussions and hands-on learning during class.
8. **Instruction Execution Mapping:** Students map the complete instruction execution cycle (Fetch → Decode → Execute → Memory → Write Back) to visualize data flow through the ALU, registers, memory, and buses.

Assessment Structure:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage.

- To qualify and become eligible to appear for SEE, in the **CIE**, a student must score at least **40% of 50 marks**, i.e., **20 marks**.
- To pass the **SEE**, a student must score at least **35% of 50 marks**, i.e., **18 marks**.
- Notwithstanding the above, a student is considered to have **passed the course**, provided the combined total of **CIE and SEE** is at least **40 out of 100 marks**.



Continuous Comprehensive Assessments (CCA)+ Internal Assessment Test (IA) = Continuous Internal Evaluation CIE [25+25=50 marks]

A minimum of **two Internal Assessment Tests (IA)** shall be conducted, carrying a total of **25 marks**.

In addition, **Continuous Comprehensive Assessment (CCA)** shall be conducted for a total of **25 marks**.

It is recommended to include **a maximum of two learning activities** as part of the CCA to foster the **holistic development of students**. These activities shall be:

Sl. No.	Name of the Learning activity	Maximum Marks
1.	Develop Verilog HDL programs for the given problem statement	25

List of Problem Statements:

Sl. No.	Problem Description
1.	Simplify a given 4-variable Boolean expression using appropriate minimization techniques and implement the simplified function using basic logic gates through simulation.
2.	Design and simulate a 4-bit adder-Subtractor circuit using basic logic gates, and verify its functionality for different input combinations.
3.	Design Verilog HDL to implement Binary Adder-Subtractor – Half and Full Adder, Half and Full Subtractor.
4.	Design Verilog program to implement types of De-Multiplexer.
5.	Design Verilog program for implementing various types of Flip-Flops such as SR, JK and D.
6.	Design and implement Verilog programs for SR and JK Flip-Flops. Verify their functionality using simulation.
7.	Identify and document the processor type, clock speed, cache memory, RAM capacity, storage devices, and operating system details of a desktop computer using system information tools.
8.	Observe and record CPU utilization, memory usage, disk activity, and running processes using Task Manager/System Monitor before and after executing multiple applications, and compare the changes in system resource usage.
9.	Measure file transfer rates and access times using desktop computers and discuss their impact on overall system performance and memory hierarchy.
10.	Compare the features, instruction formats, and performance characteristics of CISC and RISC architectures through the study of modern processors, and summarize the findings in a comparative table.



11.	Study of semiconductor RAM and ROM organization with practical examples by determining memory capacity, address lines, data lines, and memory locations, and compare their characteristics.	
12.	Identify the major hardware components of a computer system through hands-on inspection, and demonstrate the ability to safely disassemble and reassemble the system while maintaining its functionality.	

NOTE: Make use of simulation packages like Xilinx Simulator Interface , Multisim, Modalism, PSpice or any other relevant. At the end of the course, the students need to submit the report on conduction of the activity with relevant data to the course instructor.



Rubrics for Learning Activity:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Understanding of Digital Logic / Computer Organization Concepts(CO1, CO2 / PO1, PO2)	Demonstrates complete understanding of the underlying digital logic/computer organization concepts and correctly explains the design approach.	Demonstrates very good understanding with negligible errors.	Demonstrates adequate understanding with minor misconceptions.	Shows limited understanding and requires guidance.	Unable to understand the concepts.
Circuit Design / Verilog Program Development(CO2, CO3, CO4 / PO2, PO3, PO5)	Designs correct logic circuits/Verilog programs with efficient implementation and proper coding practices.	Design/program is correct with minor improvements possible.	Design/program is mostly correct but contains moderate errors.	Design/program is partially correct with logical or coding errors.	Unable to develop a functional circuit/program.
Simulation, Testing and Verification(CO2, CO3, CO4 / PO3, PO4, PO5)	Successfully simulates the design, verifies all test cases, and accurately interprets the results.	Simulation is correct for most cases with minor errors in verification.	Simulation works for standard cases with limited analysis.	Simulation is incomplete or verification contains significant errors.	Unable to simulate or verify the design.
Analysis of Results / Hardware Observation(CO3, CO4, CO5 / PO2, PO4)	Accurately analyzes outputs, compares results, and provides meaningful observations and conclusions.	Good analysis with minor omissions.	Adequate analysis with limited conclusions.	Incomplete analysis and incorrect observations.	No meaningful analysis or conclusions.
Documentation and Record Maintenance(CO3-CO5 / PO2, PO4)	Record is complete, neatly organized, includes circuit diagrams/programs, simulation outputs, observations, and follows documentation standards.	Well documented with minor formatting issues.	Adequately documented with some missing details.	Poor documentation and incomplete record.	No proper documentation submitted.



Term Work (TW) and Self Learning (SL) components in Number of Hours per semester		
Term work (includes assignments, seminars, micro projects, industrial visits, any other student learning activities etc.)		
Sl. No.	Term Work (TW) Activity	Number of Hours / Semester
1.	Learning Activity	30
SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.		
Sl. No.	Self-Learning (SL) Activity	Number of Hours / Semester
1	Infosys springboard course on: Digital Logic Design Overview	04
2	Infosys springboard course on: Computer Organization	12
3	Infosys springboard course on: VLSI Digital Design - Chip Design and Verilog Programming	10
4	Virtual lab simulation: Computer Organization https://cse11-iiith.vlabs.ac.in OR Simulation Enabled Experiential Learning (SEEL) https://seel.vtu.ac.in/	08

Continuous Internal Evaluation (CIE)

1. The CIE marks shall be decided based on internal tests and other outcome-based activities. The continuous internal evaluation shall be carried out by the teacher handling the course.
2. Out of 50 marks earmarked for CIE, 25 marks shall be assigned for internal tests. The first test shall be conducted after completing two modules of the syllabus and the second one after completing the remaining three modules.
3. The remaining 25 marks shall be assigned for Continuous Course Assessment (CCA), (Learning Activity - Term works) conducted as per the rubrics listed in the assessment section of the respective syllabus.
4. A student shall obtain a minimum of 40% marks or 20 out of 50 marks allotted to CIE to become eligible to appear for the SEE.

Passing requirement in SEE

1. For a pass in the Semester End Examination (SEE), a student shall secure a minimum of 35 marks out of 100.
2. For the declaration of a pass grade, the sum of the Continuous Internal Evaluation (CIE) marks (out of 50) marks and the SEE marks scaled down to 50 shall be greater than or equal to 40 marks.
3. If the total (CIE + SEE) is less than 40 marks, the student shall be awarded the grade 'F' (Fail).



OPERATING SYSTEMS			Credits	03
Course Code	25CSCC340	Course type	PCC	
Teaching Hours/Week (L:T:P:S)	3:0:0:4	Total Hours of Pedagogy	42:0:0:48:90	
CIE type	Theory	CIE Marks	50	
SEE Examination type	Theory	SEE Marks	50	
SEE Exam Hours	3	Total Marks	100	
Course Objectives: • To Demonstrate the need for OS and different types of OS • To discuss suitable techniques for management of different resources • To demonstrate different APIs/Commands related to processor, memory, storage and file system management.				
Teaching–Learning Process (General Instructions) These are sample strategies that teachers can use to accelerate the attainment of the various course outcomes. 1. Lecturer methods (L) need not to be only traditional lecture methods, but alternative effective teaching methods could be adopted to attain the outcomes. 2. Use of Video/Animation to explain functioning of various concepts. 3. Encourage collaborative (Group Learning) Learning in the class. 4. Role play for process scheduling. 5. Demonstrate the installation of any one Linux OS on VMware/Virtual Box				
Module-1		Number of hours: 8:0:0:0		
Introduction to operating systems: System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management. Operating System Services: User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure.				
Textbook 1: Chapter — 1 (1.1-1.8), 2 (2.2-2.7)				
Module-2		Number of hours: 9:0:0:0		
Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication. Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling, Multiple processor scheduling.				
Textbook 1: Chapter — 3 (3.1-3.4), 4 (4.1,4.3,4.4,4.6), 5 (5.1 -5.5)				
Module-3		Number of hours: 9:0:0:0		
Process Synchronization: Background; The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.				

**Textbook 1: Chapter — 6 (6.1-6.4,6.6-6.7), 7 (7.1 -7.7)****Module-4**

Number of hours: 8:0:0:0

Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation: Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

Textbook 1: Chapter -8 (8.1-8.6), 9 (9.1-9.6)**Module-5**

Number of hours: 8:0:0:0

File System, Implementation of File System: File system: File concept; Access methods; Directory and Disk structure; File system mounting; File sharing; Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

Mass Storage Structure: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management.

Textbook 1: Chapter — 10 (10.1-10.5), 11 (11.1-11.5), 12 (12.1-12.5), 14 (14.1-14.4)**Course outcome (Course Skill Set)**

At the end of the course, the student will be able to:

1. Explain the structure and functionality of the operating system.
2. Demonstrate process management using appropriate CPU scheduling algorithms.
3. Experiment with various process synchronization and Deadlock handling Techniques.
4. Build various Techniques for memory management.
5. Apply file organization techniques and disk scheduling strategies to solve storage management problems.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

CIE for the theory component of the PCC (maximum marks 50)**Continuous Internal Evaluation:**

- For the Assignment component of the CIE, there are 25 marks and for the Internal Assessment
- The first test will be administered after 40-50% of the syllabus has been covered, and the second test will be administered after 85-90% of the syllabus has been covered
- Any two assignment methods mentioned in the 22OB2.4, if an assignment is project-based then only one assignment for the course shall be planned. The teacher should not conduct two assignments at the end of the semester if two assignments are planned.



- For the course, CIE marks will be based on a scaled-down sum of two tests and other methods of assessment.

SEE for PCC

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (**duration 03 hours**)

Suggested Learning Resources:**Textbook**

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Reference Books

2. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
3. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
4. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
5. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

Web links and e-Resources:

- <https://youtu.be/vBURTt97EkA>
- https://www.youtube.com/watch?v=783KABtuE4&list=PLIemF3uozcAKTgsCljS2voMK3TMROYE_f
- <https://www.youtube.com/watch?v=3ITLMMeeXY&list=PL3pGy4HtqwD0n7bQfHjPnsWzkeRn6mkO>

Team work (TW) and Self-learning (SL) components

Team work (includes assignments, seminars, micro projects, industrial visits, any other student activities)

Module No.	Topic	Term work Activity	Resource / Link	No. of Hours	Assessment Method
1	Comparative Study of Operating Systems	Compare Windows, Linux, Android, and macOS based on architecture, scheduling, memory management, security, and file systems.	https://www.geeksforgeeks.org/operating-systems/different-operating-systems/	8	Quiz
2	Process Scheduling Simulation	Design and simulate FCFS, SJF, Priority, and	https://github.com/SorawitChok/Process-Scheduling-Simulation-Project	10	Presentation, Viva and Report



		Round Robin scheduling algorithms using C, Java, or Python. Compare waiting time, turnaround time, and CPU utilization.			submission
4	Page Replacement Algorithms	Implement FIFO, LRU and other page replacement algorithms and compare page faults.	https://www.youtube.com/watch?v=8rcUs5RutX0	10	Presentation, Viva and Report submission

Self-Learning (includes MOOCs, spoken tutorials, online educational resources etc.)

Module No.	Topic	Self-Learning Activity	Resource / Link	No. of Hours	Assessment Method
3	Real-Life Deadlock Examples	Identify deadlock situations in daily life (traffic intersections, railway crossings, resource sharing) and explain how they relate to the four necessary conditions of deadlock.	https://dev-aditya.medium.com/deadlocks-a-real-world-example-2e6269795c6a	10	Presentation and Report
5	Virtualization Concepts	Learn differences between virtualization and containerization.	https://www.youtube.com/watch?v=OT8HNz3kKSE	10	Quiz

**Rubrics for Learning Activity:**

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Understanding of OS Concept (CO1, CO2 /PO1,PO2)	Demonstrates complete understanding of the OS concept and correctly identifies the appropriate algorithm/ technique.	Demonstrates very good understanding with negligible errors.	Demonstrates adequate understanding with minor misconceptions	Shows limited understanding; requires significant guidance.	Unable to understand the OS concept.
Algorithm Design (CO2, CO3, CO4 /PO2,PO3)	Designs an efficient and logically correct algorithm/flow for the problem.	Algorithm is correct with minor improvements possible.	Algorithm is mostly correct but lacks optimization.	Algorithm is partially correct with logical errors.	Unable to design a suitable algorithm.
C Program Implementation (CO1-CO5 /PO1,PO3)	Implements the solution completely in C with correct syntax, modular programming, and appropriate data structures.	Program executes correctly with minor coding issues.	Program works for most cases with moderate errors.	Program executes partially with several errors.	Unable to implement the solution in C.
Output Correctness and Analysis (CO3, CO4, CO5 /PO2, PO4)	Produces correct results for all test cases and accurately analyzes the outcome.	Correct output for most test cases with minor analysis errors.	Produces acceptable output for standard cases.	Output is partially correct with incorrect analysis.	Incorrect output or no meaningful analysis.



Code Quality and Documentation (CO3- CO6 /PO2,PO4)	Code is well-structured, properly indented, commented, and follows coding standards.	Good coding style with few documentati on issues.	Adequate documentatio n and readability.	Poor documentatio n and inconsistent coding style.	No documentatio n; difficult to understand code.
--	--	---	--	--	--



DATA STRUCTURES AND APPLICATIONS			Credits	3
Course Code	25CSCC350	Course type	PCC	
Teaching Hours/Week (L:T:P)	3:0:0	Total Hours of Pedagogy in a semester L:T:P:SL:TW&SL:TH	(42:0:0:48:90)	
CIE type	IA tests + CCA	CIE Marks	50	
SEE Examination type	Theory	SEE Marks	50	
SEE Exam Hours	3	Total Marks	100	
Course outcome (Course Skill Set)				
At the end of the course, the student will be able to:				
1. Recall the fundamental concepts, terminology, and characteristics of various data structures.				
2. Explain the principles, representation, and operations of linear and non-linear data structures.				
3. Implement different primitive data structures, their operations, and primitive applications.				
4. Choose a suitable data structure to solve computational and real-world problems.				
5. Make use of hashing techniques and leftist trees to organize the data effectively.				
Module-1			Number of hours: 8	
Introduction: Data Structures, Classifications (Primitive & Non-Primitive). Pointers and Dynamic Memory Allocation, Arrays, Dynamically Allocated Arrays, Structures and Unions. Sparse Matrices: Sparse Matrix Representation and Transposing a Matrix.				
Textbook: Chapter-1 (1.2), Chapter-2 (2.1-2.3, 2.5.2, 2.5.3), Reference Book-1 (1.3)				
Module-2			Number of hours: 8	
Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to Postfix Conversion, Evaluation of Postfix Expression. Queues: Definition, Array Representation, Queue Operations.				
Textbook: Chapter-3 (3.1-3.3, 3.6)				
Module-3			Number of hours: 9	
Circular Queues, Circular queues using Dynamic arrays.				
LINKED LISTS: Singly Linked Lists and Chains, Representing Chains in C, Additional List Operations, Sparse Matrices: Sparse Matrix Representation, Doubly Linked List.				
Textbook: Chapter-3 (3.4), Chapter-4 (4.1 to 4.2, 4.5, 4.7.1, 4.8)				



Module-4	Number of hours: 9
Trees: Introduction-Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals – preorder, Inorder, postorder. Binary Search Trees – Definition, Searching, Insertion, Deletion, Counting Binary Trees. Textbook: Chapter-5 (5.1.1, 5.2, 5.3.1-5.3.3, 5.7.1- 5.7.4, 5.11.1)	
Module-5	Number of hours: 8
GRAPHS: The Graph Abstract Data Types, Elementary Graph Operations. HASHING: Introduction, Static Hashing, Dynamic Hashing. PRIORITY QUEUES: Single and double ended Priority Queues, Leftist Trees. Textbook: Chapter 6 (6.1, 6.2), Chapter 8 (8.1 to 8.3), Chapter 9 (9.1, 9.2)	
Suggested Learning Resources: (Textbook/ Reference Book/ Manuals): Textbooks: 1. Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2023. Reference Books: 1. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed., McGraw-Hill, 2014. 2. Aaron M Tenenbaum, Data Structures using C, Pearson Education, 2019. 3. Reema Thareja, Data Structures Using C, 3rd Ed., Oxford Press, 2012. Web links and e-Resources: 1. Introduction to Data Structures: https://www.youtube.com/watch?v=zWg7U0OEAOE&list=PLBF3763AF2E1C572F&index=1 2. Stacks: https://www.youtube.com/watch?v=g1USSZVWDsY&list=PLBF3763AF2E1C572F&index=2 3. Queues and Linked Lists: https://www.youtube.com/watch?v=PGWZUgzDMYI&list=PLBF3763AF2E1C572F&index=3 4. Trees: https://www.youtube.com/watch?v=tORLeHHtazM&list=PLBF3763AF2E1C572F&index=6 5. Hashing: https://www.youtube.com/watch?v=KW0UvOW0Xlo&list=PLBF3763AF2E1C572F&index=5 6. Infosys Springboard: Data Structures & Algorithms using C++, C and Python – 2026: https://infyspringboard.onwingspan.com/ 7. Virtual Labs: Computer Science and Engineering - Data Structures-1: https://ds1-iiith.vlabs.ac.in/	



Teaching-Learning Process (Innovative Delivery Methods):

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

1. Flipped Classroom
2. Problem-Based Learning (PBL)
3. Case-Based Teaching
4. Simulation and Virtual Labs

5. ICT-Enabled Teaching

Assessment Structure:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage.

- To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks.
- To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks.
- Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

Continuous Comprehensive Assessments (CCA) + Internal Assessment Test (IA) = Continuous Internal Evaluation CIE [25 + 25 = 50 marks]

A minimum of **two Internal Assessment Tests (IA)** shall be conducted, carrying a total of **25 marks**. In addition, **Continuous Comprehensive Assessment (CCA)** shall be conducted for a total of **25 marks**.

It is recommended to include **a maximum of two learning activities** in the CCA to foster students' **holistic development**. These activities shall be:

Sl. No.	Name of the Learning activity	Maximum Marks
1.	Problem-Solving: Competitive Programming Assignments (HackerRank/HackerEarth/LeetCode)	15
2.	Problems with stack applications, trees, tree traversals, threaded binary trees, BST, graphs, and traversals, Hashing and Leftist trees.	10

**Rubrics for Learning Activity -1:**

Evaluated for 25 marks and scale down to 15. The students need to submit the report on the solutions to the problems with relevant data to the course instructor.

Criteria/Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Problem Understanding and Analysis (CO2, CO3 /PO1, PO2)	Clearly understands the problem statement, identifies constraints, and selects an effective approach independently.	Understands the problem well and identifies most constraints with minor guidance.	Understands the basic requirements but misses some constraints or cases.	Has partial understanding and requires significant guidance	Unable to interpret the problem requirements .
Selection and Application of Data Structures (CO3, CO5 /PO1, PO2)	Selects the most suitable data structure (Stack, Queue, List, Tree, Hashing, etc.) and justifies the choice based on problem requirements.	Selects an appropriate data structure with minor inefficiencies.	Uses a suitable data structure but lacks optimization or justification.	Chooses a partially suitable data structure with limitations.	Unable to select an appropriate data structure.



Algorithm Design and Problem-Solving Skills (CO3, CO5 /PO2, PO3)	Designs an efficient algorithm with correct logic, optimal complexity, and handles all edge cases.	Develops a correct algorithm with minor logical issues.	Develops a basic algorithm that solves standard cases.	Algorithm is partially correct and requires improvements.	Unable to develop a workable algorithm
Implementation and Coding Skills (CO4/PO4)	Implements error-free, modular, readable code that passes all test cases on the platform.	Implements correct code with minor errors; passes most test cases.	Code works for normal cases with limited errors.	Code has multiple errors and passes limited test cases.	Code is incomplete or does not execute successfully.
Code is incomplete or does not execute successfully. (CO4, CO5 /PO4)	Performs extensive testing, handles edge cases, debugs effectively, and optimizes time and space complexity.	Tests adequately and improves solution efficiency with minor issues.	Performs basic testing and achieves acceptable efficiency.	Performs limited testing with inefficient solutions.	No proper testing or optimization is performed.

**Rubrics for Learning Activity -2:**

Evaluated for 25 marks and scale down to 10. The students need to submit the report on problem & solutions to the course instructor.

Criteria/Scale	Superior (5)	Good (4)	Fair (3)	Needs Improve ment (2)	Unacceptable (1)
Understanding of Data Structure Concepts through Certification Modules (CO1, CO2 /PO1,PO2)	Demonstrates comprehensive understanding of fundamental and advanced data structure concepts, terminology, and applications through successful completion of certification modules.	Demonstrate s good understandin g of concepts with minor gaps.	Understan ds basic concepts and terminolo gy with limited depth.	Shows partial understa nding and requires additiona l support.	Unable to demonstrate understand of fundamental concepts.
Application of Data Structures and Algorithms in Problem Solving (CO3, CO5 /PO1,PO2)	Effectively applies suitable data structures and algorithms to solve complex computatio nal problems and certification assessments.	Applies appropriate concepts to solve problems with minor errors.	Applies basic data structure concepts to solve standard problems.	Applies concepts with significan t guidance and limited accuracy.	Unable to apply concepts for problem solving.



Completion of Certification and Learning Progress (CO1, CO4 /PO4)	Successfully completes all assigned Infosys Springboard modules, assessments, and certification requirements with excellent Performance	Completes most modules and assessments with good performance.	Completes required modules with satisfactory performance.	Completes only partial modules or assessments.	Does not complete the assigned certification activities.
Implementation Skills and Practical Application (CO4/PO4)	Demonstrates strong programming skills by implementing data structures and algorithms through hands-on exercises and coding activities.	Implements solutions correctly with minor errors.	Implements basic programs with limited optimization.	Requires significant assistance for implementation tasks.	Unable to implement programming solutions effectively.
Self-Learning, Technical Skills, and Continuous Improvement (CO2, CO5 /PO4, PO11)	Demonstrates independent learning, explores advanced concepts, and applies knowledge beyond the certification requirements.	Shows good self-learning ability and explores additional resources.	Completes learning activities with moderate involvement.	Shows limited engagement with learning resources.	Does not demonstrate self-learning improvement

**Term Work (TW) and Self Learning (SL) components in Number of Hours per semester**

Term work (includes assignments, seminars, micro-projects, industrial visits, any other student learning activities etc.)

Sl. No.	Term Work (TW) Activity	Number of Hours / Semester
1.	Learning Activity -1	20
2.	Learning Activity -2	10
SL: Self-learning, MOOCs, spoken tutorials, online educational resources, etc.		
Sl. No.	Self-Learning (SL) Activity	Number of Hours / Semester
1.	Springboard certification on: Data Structures and Algorithms	18

Continuous Internal Evaluation (CIE)

1. The CIE marks shall be decided based on internal tests and other outcome-based activities. The continuous internal evaluation shall be carried out by the teacher handling the course.
2. Out of 50 marks earmarked for CIE, 25 marks shall be assigned for internal tests. The first test shall be conducted after completing two modules of the syllabus and the second one after completing the remaining three modules.
3. The remaining 25 marks shall be assigned for Continuous Course Assessment (CCA), conducted as per the rubrics listed in the assessment section of the respective syllabus.
4. A student shall obtain a minimum of 40% marks or 20 out of 50 marks allotted to CIE to become eligible to appear for the SEE.

Passing requirement in SEE

1. For a pass in the Semester End Examination (SEE), a student shall secure a minimum of 35 marks out of 100.
2. For the declaration of a pass grade, the sum of the Continuous Internal Evaluation (CIE) marks (out of 50) marks and the SEE marks scaled down to 50 shall be greater than or equal to 40 marks.
3. If the total (CIE + SEE) is less than 40 marks, the student shall be awarded the grade 'F' (Fail).



DATA STRUCTURE LABORATORY			Credits	1
Course Code	25CSCL360	Course type	SEE	
Teaching Hours/Week (L: T:P)	0:0:2	Total Hours of Pedagogy L:T:P:SL:TW	0:0:28:2:30	
CIE type	Lab CIE + Lab test	CIE Marks	50	
SEE Examination type	Practical	SEE Marks	50	
SEE Exam Hours	3	Total Marks	100	

Course outcome (Course Skill Set)

At the end of the course, the student will be able to:

1. Develop C programs to implement fundamental data structures and their operations for solving computational problems.
2. Make use of appropriate data structures to organize, store, and manipulate data for solving real-world problems using C.
3. Build a laboratory record consisting of problem statements, algorithm/design, implementation, program outputs, and result with appropriate inferences.

Note:

1. The laboratory syllabus consists of PART-A and PART-B. While PART-A has 6 conventional experiments, PART-B has 6 typical open-ended experiments. The maximum marks for the laboratory course are 100.
2. Both PART-A and PART-B are considered for CIE and SEE.
3. Students have answer 1(one) question from PART-A and 1(one) question from PART-B.
 - a. The questions set for SEE shall be from among the experiments under PART-A. It is evaluated for 70 marks out of the maximum 100 marks.
 - b. The open-ended question set for SEE shall be any other open-ended question and not selected from the experiments under PART-A. It shall be evaluated for 30 marks.
4. For Continuous Internal Evaluation (CIE) during the semester, classwork shall typically include open-ended questions from Part-B, along with other similar questions designed to enhance the students' skills.

PART - A
FIXED SET OF EXPERIMENTS

1. Develop a Program in C for the following:
 - a. Define a structure Book with the following fields: Book_ID, Title, Author, Price, Availability_Status (Available/Issued)
 - b. Dynamically allocate memory to store the details of N books, where N is entered by the user.
 - c. Implement the following functions:
 - i. create() – To allocate memory and read the details of N books and store them in the allocated memory.
 - ii. display() – To display the details of all books available in the library.
 - iii. search() – To search for a book using Book ID and display the book details if found.

- iv. issueBook() – To issue a book by updating its availability status and display an appropriate message.
 - v. returnBook() – To return an issued book by updating its availability status and display an appropriate message.
 - d. Provide a menu-driven interface with the following options:
 - i. Add Book Records
 - ii. Display All Book Records
 - iii. Search Book by Book ID
 - iv. Issue a Book
 - v. Return a Book
 - vi. Exit
- 2. Develop a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)
 - a. Push an Element on to Stack
 - b. Pop an Element from Stack
 - c. Demonstrate Overflow and Underflow situations on Stack
 - d. Display the status of Stack
 - e. Exit

Support the program with appropriate functions for each of the above operations
- 3. Develop a program in C for simulating a Printer Queue using Queue data structure with the following operations:
 - a. Add a print job to the Printer Queue.
 - b. Process (delete) a print job from the Printer Queue.
 - c. Display all pending print jobs in the Queue.
 - d. Demonstrate Queue overflow and underflow conditions.
 - e. Exit.
- 4. Develop a menu-driven Program in C for the following operations on a Singly Linked List of Student Records:
 - a. Create a linked list using front insertion.
 - b. Display all student records and count the number of nodes.
 - c. Insert and delete nodes at the beginning of the list.
 - d. Search for a student record using USN.
- 5. Develop a Program in C for the following:
 - a. Create a Binary Tree by inserting integer elements entered by the user in level order.
 - b. Implement the following functions:
 - i. createTree() – Create a binary tree using dynamic memory allocation.
 - ii. preorder() – Display the elements using Preorder Traversal.
 - iii. inorder() – Display the elements using Inorder Traversal.
 - iv. postorder() – Display the elements using Postorder Traversal.
 - c. Provide a menu-driven interface with the following options:
 - I. Create Binary Tree
 - II. Display Preorder Traversal

- III. Display Inorder Traversal
- IV. Display Postorder Traversal
- V. Exit

6. Develop a Program in C for the following operations on Graph(G) of Cities
- a. Create an Adjacency Matrix representing the graph of N cities
 - b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method.

PART - B
OPEN ENDED EXPERIMENTS

Open-ended experiments are a type of laboratory activity where the outcome is not predetermined and students are given the freedom to explore, design, and conduct the experiment based on the problem statements as per the concepts defined by the course coordinator. It encourages creativity, critical thinking, and inquiry-based learning.

1. Develop a C program to:
 - a. Read two sparse matrices of the same order.
 - b. Convert both into 3-tuple representation.
 - c. Perform matrix addition.
 - d. Display the resultant matrix in both normal matrix form and 3-tuple form.
2. Develop a C-based Expression Conversion tool that uses stack data structure to convert arithmetic expressions from infix notation to postfix notation. The tool should support alphanumeric operands, parentheses, and the operators (+, -, *, /, %, ^). Analyze the correctness of the conversion by testing various valid expressions.
3. Develop a menu-driven Program in C for the following operations on Circular Queue (integer/floating point data):
 - a. Insert an element into the Circular Queue.
 - b. Delete an element from the Circular Queue.
 - c. Display the contents of the Circular Queue.
 - d. Demonstrate overflow and underflow conditions.
 - e. Exit.
4. Develop a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo etc.
 - a. Create a DLL of N Employees Data by using end insertion.
 - b. Display the status of DLL and count the number of nodes in it
 - c. Perform Insertion and Deletion at End of DLL
 - d. Perform Insertion and Deletion at Front of DLL
 - e. Exit
5. Develop a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers.
 - a. Create a BST of N Integers
 - b. Traverse the BST in Inorder, Preorder and Post Order
 - c. Search the BST for a given element (KEY) and report the appropriate message

d. Exit

6. Develop a Program in C for an Employee Record Management System using Hashing with the following operations:

- Create a hash table to store employee records using Employee ID as the key.
- Insert employee records into the hash table using the Hash function $H: K \rightarrow L$ as $H(K) = K \bmod m$
- Resolve collisions using Linear Probing.
- Search for an employee record using a given Employee ID and display the details.
- Display the contents of the hash table along with the corresponding memory locations.

Suggested Learning Resources: (Text Book/ Reference Book/ Manuals):

Text books:

- Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2023.

Reference books / Manuals:

- Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2023.
- A M Tenenbaum, Data Structures using C, PHI, 1989
- Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.

Web links and Video Lectures (e-Resources):

- GNU-GCC Documentation: <https://gcc.gnu.org/>
- Virtual Labs on Data Structures-1: <https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html>
- DSA Tutorial: <https://www.geeksforgeeks.org/dsa/dsa-tutorial-learn-data-structures-andalgorithms/>
- NPTEL Course on DSA: <https://nptel.ac.in/courses/106106130>
- NPTEL Course on Introduction to C Programming: <https://nptel.ac.in/courses/106104128>

Teaching-Learning Process (Innovative Delivery Methods):

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

- Engineering tool usage for the conduction of experiment
- Demonstration through ICT tools
- Use of virtual labs

Assessment Structure:

The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying 50% weightage (i.e., 50 marks each).

The CIE marks awarded shall be based on the continuous evaluation of the laboratory report using a defined set of rubrics. Each experiment report can be evaluated for 30 marks. Total

marks for the all report shall be scaled down to 30 marks. The laboratory test (duration 03 hours) at the end of the last week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and Marks scored shall be scaled down to 20 marks. Final average of marks for report and test shall be out of 50 marks.

For both CIE and SEE, the student is required to conduct one experiment each from both Part A and Part B.

Rubrics for CIE – Continuous assessment:

Criteria/Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Unacceptable (1)
Fundamental Knowledge: Understanding the problem statement and selection of appropriate data structure [CO1, CO2] [PO1, PO2]	Demonstrates in-depth understanding of data structure concepts and correctly identifies the most suitable data structure for the given problem.	Demonstrates good understanding of data structure concepts and selects an appropriate data structure with minor guidance.	Demonstrates satisfactory understanding and selects a suitable data structure for the problem.	Demonstrates partial understanding and selects an inappropriate or partially suitable data structure.	Unable to understand the problem or identify an appropriate data structure.
Design of Algorithm and Data Structure [CO1, CO2] [PO2, PO3]	Develops multiple algorithmic solutions, compares them, and justifies the best design with appropriate data structure selection.	Develops more than one design but provides limited justification for selecting the best solution.	Develops a correct algorithm and data structure with basic explanation.	Develops an incomplete algorithm or partially correct design.	Unable to design a suitable algorithm or data structure.
Implementation (Program Coding) [CO1, CO2] [PO5, PO8]	Implements the selected data structure efficiently using appropriate C constructs	Implements the selected data structure correctly and explains	Implements the required data structure with minor coding issues.	Implements the program with significant errors or incomplete operations.	Unable to implement the required data structure program.

	with correct operations and modular programming practices.	the implementation clearly.			
Program Debugging and Testing [CO1, CO2] [PO5, PO8]	Successfully compiles, debugs, and validates the program for all test cases without syntax, semantic, or logical errors.	Identifies and rectifies errors independently with good understanding of debugging.	Corrects most errors with partial understanding of debugging.	Corrects errors only with assistance and limited understanding.	Unable to debug or execute the program successfully
Results and Analysis [CO1, CO2] [PO4]	Executes the program using various test cases, validates all data structure operations, and provides detailed analysis with appropriate conclusions.	Executes all test cases and explains the results correctly.	Executes most test cases and provides basic analysis.	Executes the program but provides limited or incorrect analysis.	Unable to verify or interpret the program output correctly.
Demonstration and Documentation [CO3] [PO8, PO9, PO11]	Demonstration is excellent. Lab record is well-organized with problem statement, algorithm, data structure representation, program, output,	Demonstration is good. Lab record is organized with all major sections and appropriate formatting.	Demonstration is satisfactory. Lab record contains most required sections with acceptable formatting.	Demonstration is incomplete. Lab record has missing sections or inadequate formatting.	Demonstration is poor or absent. Lab record is incomplete, improperly formatted, or not submitted on time.

	observations , and conclusions presented using proper formatting.				
--	---	--	--	--	--

Rubrics for SEE / CIE Test:

Criteria /Scale	Superior (5)	Good (4)	Fair (3)	Needs Improve ment (2)	Unacceptable (1)
Fundamental Knowledge (2) [CO1, CO2] [PO1]	Demonstrates excellent knowledge of data structures, their operations, and applications.	Demonstrates good knowledge of major data structure concepts.	Demonstrates average understanding of basic concepts.	Demonstrates limited conceptual understanding.	Does not understand the fundamental concepts.
Understanding of Problem Definition (1) [CO1, CO2] [PO2]	Clearly understands the problem and identifies the most appropriate data structure.	Understands the problem with minor clarification.	Demonstrates partial understanding of the problem.	Demonstrates minimal understanding of the problem.	Unable to understand the problem statement.
Design and Implementation (3) [CO1, CO2] [PO3]	Designs an efficient algorithm and correctly implements the selected data structure using appropriate C constructs.	Designs and implements the required solution with minor issues.	Implements the core algorithm with partial correctness.	Partially implements the solution with significant errors.	Unable to design or implement the solution.
Results and Analysis (2) [CO1, CO2] [PO4]	Successfully executes the program using multiple test cases,	Executes the program correctly and provides	Executes the program with limited test cases and basic analysis.	Executes the program but cannot properly	Unable to execute or analyze the program.



	validates all operations, and provides accurate analysis.	good analysis.		analyze the results.	
Communication (Viva Voce) (2) [CO3] [PO8, PO9]	Clearly explains the working of the selected data structure, algorithm, implementation, and application using correct technical terminology.	Explains the concepts with good communication and correct terminology.	Explains basic concepts with average communication skills.	Provides incomplete explanation with incorrect terminology.	Unable to explain the implementation or answer questions satisfactorily.
- To qualify and become eligible to appear for SEE, in the CIE component, a student must secure a minimum of 40% of 50 marks, i.e., 20 marks. - To pass the SEE component, a student must secure a minimum of 35% of 50 marks, i.e., 18 marks. - A student is deemed to have successfully completed the course if the combined total of CIE and SEE is at least 40 out of 100 marks.					



Course title: Project Management with Git			Credits	01
Course Code	25CSAE370	Course type	AEC	
Teaching Hours/Week(L:T:P:S)	0:0:2:0	Total Hours of Pedagogy in a semester	0:0:28:2:30	
CIE type	Lab CIE + Lab test	CIE Marks	50	
SEE Examination type	Practical	SEE Marks	50	
SEE Exam Hours	3	Total Marks	100	
Course Outcomes: At the end of the course the student will be able to: CO1. Demonstrate the use of basic Git repository commands for version control operations. CO2. Demonstrate branch management strategies for collaborative software development. CO3. Make use of collaboration techniques using remote repositories and distributed version control systems. CO4. Demonstrate the use of Git tags, releases, and advanced Git operations for effective project management. CO5. Apply Git history modification techniques to maintain repository integrity and support project requirements.				
	Set of Experiments		Number of hours: 0:0:28:0	
1.	Setting Up and Basic Commands: Initialize a new Git repository in a directory. Create a new file and add it to the staging area, and commit the changes with an appropriate commit message.			
2.	Creating and Managing Branches: Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."			
3.	Creating and Managing Branches: Write the commands to stash your changes, switch branches, and then apply the stashed changes.			
4.	Collaboration and Remote Repositories: Clone a remote Git repository to your local machine.			
5.	Collaboration and Remote Repositories: Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.			
6.	Collaboration and Remote Repositories: Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.			
7.	Git Tags and Releases: Write the command to create a lightweight Git tag named “v1.0” for a commit in your local repository.			
8.	Advanced Git Operations: Write the command to cherry-pick a range of commits from "source-branch" to the current branch.			
9.	Analyzing and Changing Git History:			



	Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?
10.	Analyzing and Changing Git History: Write the Git command to display all commits made by the author "JohnDoe" between "January 1, 2025" and "December 31, 2025."
11.	Analyzing and Changing Git History: Write the command to display the last five commits in the repository's history.
12.	Analyzing and Changing Git History: Write the command to undo the changes introduced by the commit with the ID "abc123".
Suggested Learning Resources	
Text Books	
1. Version Control with Git, 3rd Edition, by Prem Kumar Ponuthurai, Jon Loeliger Released October 2022, Publisher(s): O'Reilly Media, Inc.	
Learning Resources	
2. Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, https://git-scm.com/book/en/v2	
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013094443347369_9842782_shared/overview	
4. Git & GitHub Crash Course for Beginners [2026] https://www.youtube.com/watch?v=mAFoROnOfHs	
5. Learn GitHub within GitHub https://learn.github.com/skills	
6. Complete GitHub Actions Course - From BEGINNER to PRO https://www.youtube.com/watch?v=Xwpi0ITkL3U	
7. GIT CHEAT SHEET https://education.github.com/git-cheat-sheet-education.pdf	
8. Learn Git and GitHub https://roadmap.sh/git-github	
9. Learn Git Branching learngitbranching.js.org	
Teaching-Learning Process (Innovative Delivery Methods):	
1. Demonstration-based learning.	
2. Hands-on programming and debugging sessions	
3. Problem-based learning through project activities	
4. Self-learning using online tutorials and digital resources	

**Assessment Details**

Assessment Component	Sub Component	Maximum Marks	Weightage / Scaling	Evaluation Criteria / Remarks
Overall Assessment	Continuous Internal Evaluation (CIE)	50	50% of Total	Minimum passing marks: 20/50 (40%)
	Semester End Examination (SEE)	50	50% of Total	Minimum passing marks: 18/50 (35%)
	Overall Pass Requirement	100	100%	Students must secure 40/100 (40%) in CIE + SEE combined to earn credits.
CIE	Record / Journal	100	Scaled to 30 Marks (60%)	The record must contain all prescribed experiments. Each experiment is evaluated for 10 marks based on observation sheet, write-up, correctness, neatness, and timely submission using faculty-defined rubrics.
	Laboratory Test	100	Scaled to 20 Marks (40%)	Conducted after completion of experiments. Evaluation includes test write-up, experiment conduction, procedure, and result (60%) and viva-voce (40%).
	Total CIE	-	50 Marks	Sum of scaled journal (30) and laboratory test (20).



SEE	Practical Examination	100	Scaled to 50 Marks	Conducted jointly by two internal examiners appointed by the Head of the Institution. Minimum duration: 2 hours. The student selects one experiment from a jointly prepared question lot covering all syllabus experiments.
	SEE Evaluation Rubrics	100	Scaled to 50 Marks	Write-up – 20%, Conduction Procedure & Result – 60%, Viva-Voce – 20% (or course-specific rubrics jointly decided by examiners).
	Change of Experiment	-	Penalty Applicable	Change of experiment is permitted only once. If changed, 15% of the marks allotted to the procedure component shall be awarded zero.
Final Result	Credit Award	100	-	A student earns the course credits only after satisfying the minimum passing requirements in CIE (20/50), SEE (18/50), and Overall (40/100).

Self-learning (SL) components

Self-Learning (includes MOOCs, spoken tutorials, online educational resources etc.)

Module No.	Topic	Self-Learning Activity	Resource / Link	No. of Hours	Assessment Method
1	Quiz on GitHub	Self-study	https://www.youtube.com/watch?v=mAFoROnOfHs	2	Quiz



Community Project (Project-Based Learning) / Societal Project		Semester: 3	
Course Code	25HUSE380	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	0:0:0:1	SEE Marks	50
Total Hours of Pedagogy	0:0:0:30:30	Total Marks	100
Credits	1	Exam hours	3
Examination type (SEE)		Practical	

Course Objectives:

The Community Project / Societal Project is a practical, project-based learning course that builds students' awareness on Educational Social Responsibility (ESR). It focuses on real issues such as the adaptation of technology for rural and urban development, the application of tools for enhanced governance in civic societies, and sustainable technological solutions for better living. Students may work individually or in groups of up to four, including interdisciplinary topics, and assessment is based on the prescribed rubrics.

Course outcome (Course Skill Set):**At the end of the course, the student will be able to:**

CO1 (L2 – Understand): Identify and analyze societal problems through field interaction and technical assessment.

CO2 (L3 – Apply): Develop skills in project planning, field survey, data collection, implementation, documentation, and reporting.

CO3 (L3 – Apply): Apply principles of social responsibility in engineering practice through community-oriented interventions

CO4 (L4 – Analyze): Exhibit multidisciplinary learning, problem-solving ability, and communication skills based on collected data (requirements), stakeholder feedback, and baseline conditions.

CO5 (L5 – Evaluate): Formulate feasible societal problem statements and solutions considering practical constraints and sustainability aspects.

CO6 (L6 – Create): Collaborate with technical and non-technical stakeholders and within the team to deliver a product/project and its impact on society.

Restrictions and professional conduct: Societal projects should be academic, socially responsible, technically focused, and politically neutral. Students and faculty must maintain professionalism, ethical conduct, and respect while working with communities and stakeholders and avoid any political, ideological, commercial, or disruptive activities. Projects should also consider practical constraints such as cost, resources, safety, social impact, and stakeholder cooperation.

SET OF PROJECT ACTIVITIES (TW/SL = 30 HOURS)**Methods for project identification:**

Students may identify the project topic using one or more of the following methods (not limited to):

Interaction with local rural or urban stakeholders such as local leaders, corporation/municipality/Panchayath officers, NGOs, etc.; review of issues highlighted in print and electronic media and social networks.

After selecting a broad topic, collect data and requirements, conduct a feasibility survey, estimate timelines and resources, approximate expenditure (if applicable), and define the problem statement with objectives clearly.



Sl. No.	Broad Areas / Project Titles	Measurable Engineering Outcome	Specific POs	SDGs
1	Computer-Based Learning Tool for Mentally Challenged Students	Design and develop a web/browser-based learning tool as per the requirements of the special school.	5, 6	4
2	Medication Management System for an Old-Age Home	Design and develop a web-based application to monitor the administration of medicine to senior citizens residing in an old-age home.	6	3
3	Virtual Labs for Learning Science	Design and develop a simulation environment to conduct experiments in Physics, Chemistry, and Biology for high-school students in rural areas.	5, 6	4
4	Local Skill Development and Employment Information Portal	Design and develop a web-based portal that provides information on local skill-development programs, vocational training, internships, and employment opportunities for youth in the community.	6	4, 8
5	Digital Complaint Portal for Gram Panchayat	Design and develop a web-based complaint portal for registering, tracking, and managing civic complaints within a Gram Panchayat.	6	16
6	Plant Care Reminder Application	Implement a reminder mechanism that generates timely notifications for watering, fertilizing, pruning, and other plant-care activities.	6	13, 15
7	Basic Crop Recommendation System	Develop a basic crop recommendation system that recommends suitable crops based on user-provided parameters such as soil type, season, rainfall, and water availability.	6	2, 13, 15
8	Lost and Found Application	Develop a Lost and Found application that enables users to report, search, and manage lost and found items.	6	11, 16
9	Campus Navigation System	Develop a campus navigation system that enables users to locate important campus facilities and obtain directions between selected locations through a simple digital map and location information.	6	15
10	Public Waste Collection Monitoring System	Design and develop a web/mobile application that enables citizens to report uncollected garbage with geotagged photos, track complaint status, and assist local municipal authorities in monitoring waste collection and sanitation services.	6	3, 11, 16
11	Volunteer Management System for NGOs	Develop a web-based system to manage volunteer registration, event scheduling, attendance, and activity reporting for NGOs.	6	16
12	Local Farmer Market Information System	Design and develop a digital platform that connects farmers directly with consumers by providing crop availability, prices, and contact information.	6	2, 8, 12
13	Government Scheme Awareness for Betterment of Livelihood	Design and develop a user-friendly interface in the native language to help rural communities access and effectively utilize relevant government schemes.	6	11



14	Community Renewable Energy Awareness and Management Portal	Design and develop a web/mobile portal to provide information on renewable energy resources and estimate potential energy savings within the community.	6	7, 11, 13
15	Digital Learning Platform for Rural Government Schools	Design and develop a digital learning platform that provides interactive learning content and activity-based learning materials in regional languages for rural and village government schools.	5, 6	4
16	Digital Heritage Preservation and Cultural Information Portal	Develop a digital platform to document, preserve, and promote local heritage sites, cultural events, traditions, and historical information.	6	11, 16
17	Community Disaster Preparedness and Emergency Information System	Design and develop a web/mobile application for disaster preparedness, emergency information dissemination, and alerts during natural disasters.	6	11, 13
18	Pesticide Usage Advisory and Organic Farming Awareness System	Design and develop an application that provides guidance on appropriate pesticide usage, promotes organic farming practices, and suggests suitable organic alternatives based on soil-test reports and crop information.	6	2, 3, 15
19	Cyber Security Awareness Platform for Phishing, Digital Arrest Scams, and Malicious Applications	Develop a web application that enables users to submit suspected text messages or links and generates a risk score based on phishing patterns, deceptive URLs, and high-risk keywords.	5, 6	16
20	Local Public Transportation Information and Route Assistance System	Design and develop a web/mobile application that provides location-based information on public transportation services, route maps, estimated travel options, and nearby transport facilities to improve accessibility and encourage public transportation use.	6	11, 15
21	Digital Inventory and Sales Management System	Design and develop a web-based inventory, billing, and sales analytics system for small businesses with dashboard reports and low-stock alerts.	6	8, 9
22	Smart Financial Literacy and Budget Planner for the Public	Develop a mobile/web application that helps users manage household budgets, savings, government schemes, and basic financial planning.	6	1, 8
23	Local Business Promotion and Market Analytics Portal	Design a digital platform that enables local vendors and small businesses to advertise products, analyze customer trends, and improve their digital visibility.	6	8, 9
24	Blood Donor and Emergency Blood Availability Portal	Develop a web/mobile platform to connect blood donors, hospitals, and patients with real-time blood availability information.	6	3
25	Women Safety and Emergency Alert Application	Develop a mobile application with SOS alerts, live location sharing, emergency contacts, and information on nearby police stations.	6	5, 11, 16
26	Public Grievance and Civic Issue Reporting System	Develop a web/mobile application that allows citizens to report civic issues such as potholes, streetlight failures, and drainage problems and monitor complaint resolution status.	6	11, 15, 16
27	Digital Public Library and Book Exchange System	Design and develop a web-based platform that enables community members or schools to manage book donations, borrowing, returns, and book-exchange activities.	6	4



28	Community Water Quality Reporting and Awareness Portal	Design and develop a web-based portal for citizens to report drinking-water issues, record basic water-quality observations, and access educational resources on safe water practices.	6	3, 6
29	Community Blood Donor Registry and Emergency Request System	Design and develop a web/mobile application that maintains a local blood-donor database, enables emergency blood requests, and notifies eligible donors based on blood group and location.	6	3, 11
30	Smart Community Health Camp Management System	Design and develop a web/mobile application to manage community health camps through online patient registration, doctor scheduling, digital health-record maintenance, and medicine-inventory tracking.	6	3
31	Food Waste Reduction and Food Donation Platform	Develop a platform that connects restaurants, hotels, households, and NGOs to facilitate the timely donation of surplus food and reduce food wastage.	6	2, 12
32	Community Tree Plantation and Green Cover Monitoring Portal	Develop a web/mobile application to record, monitor, and track tree-plantation activities and community green-cover initiatives.	6	13, 15
33	Career Guidance and Information Portal	Develop an application that provides information on aptitude tests, courses, scholarships, internships, and placement opportunities to support students in career planning.	6	4, 8
34	NGO Donation and Resource Management System	Develop a web application to manage donations, inventory, volunteers, and financial transparency for NGOs.	6	16
35	Smart Local Tourism Business Portal	Develop a web platform promoting local tourism, homestays, guides, and cultural businesses with booking facilities.	6	8, 11, 15
36	Local Artisan and Handicraft Marketplace	Design and develop a digital platform to promote local artisans and traditional handicraft products.	6	8, 11
37	Waste Recycling Marketplace	Develop a digital platform connecting households, recyclers, and scrap collectors with waste-pickup scheduling facilities.	6	11, 12
38	Digital Accessibility Evaluation and Improvement for Public Websites	Evaluate the accessibility of at least 10 government or public-service websites using WCAG guidelines, develop accessibility recommendations, and implement a prototype demonstrating measurable compliance improvements.	6	16
39	Digital Library and Knowledge Sharing Platform for Government Schools	Develop a multilingual digital library providing e-books, educational videos, and quizzes to improve access to learning resources.	6	4
40	Community Renewable Energy Awareness and Management Portal	Design and develop a web/mobile portal to provide information on renewable energy resources and promote sustainable energy practices within the community.	6	7, 11, 15
41	Local Business Digital Marketplace	Develop a web portal and onboard at least 25 local vendors with digital product catalogs to improve their online presence and market reach.	6	8, 12



42	Community Blood Donor and Emergency Volunteer Network	Develop a web/mobile application connecting blood donors and emergency volunteers to facilitate timely assistance during emergencies.	6	3
43	Mental Health Awareness and Helpline Directory App	Develop an application providing verified helpline contacts, self-help resources, and awareness content to promote mental well-being and reduce stigma.	6	3
44	Diet and Nutrition Awareness Tool for Anganwadi Children	Develop an application to track and suggest balanced meal plans for children under community nutrition programs.	5, 6	2, 3
45	Digital Literacy Portal for Elderly Citizens	Develop a simplified and accessible web application that provides step-by-step learning resources for basic smartphone and internet skills for senior citizens.	6	4, 11
46	Scholarship and Government Scheme Eligibility Checker	Develop a web-based tool that matches students with eligible scholarships and welfare schemes based on user-provided eligibility criteria.	5, 6	4
47	Digital Record-Keeping System for Self-Help Groups	Develop a digital ledger and record-keeping application to help women's Self-Help Groups manage financial transactions, member records, savings, and activities.	6	5, 8
48	Elderly Companion & Wellness Check-In App	Develop an application enabling family members and volunteers to schedule and log wellness check-ins for elderly people living alone.	6	3
49	Old-Age Home Volunteer Coordination Portal	Develop a platform that matches volunteers with old-age homes for scheduled visits, events, and support activities.	6	3, 11
50	Local Job & Skill Matching Portal for Daily-Wage Workers	Develop a web application connecting unskilled and semi-skilled workers with nearby job opportunities based on skills and availability.	6	8
51	Street Vendor Digital Registry and Support Portal	Develop an application to maintain a digital registry of street vendors, provide municipal registration information, and improve customer visibility.	6	8
52	Microloan and Repayment Tracker for Small Vendors	Develop an application helping small vendors track microloans, repayments, and daily sales.	6	8
53	Rainwater Harvesting Feasibility & Awareness Tool	Develop a tool to estimate rooftop rainwater-harvesting potential based on roof area and local rainfall data.	5, 6	6, 13
54	Water Quality Reporting System for a Village / Ward	Develop a web/mobile application that enables citizens to report water-contamination issues and track the status of corrective actions.	6	6
55	Community Skill-Building Workshop Registration and Tracker App	Develop an application to register participants and track attendance for community skill-building workshops such as tailoring, computer usage, and first aid.	6	4, 11
56	Local Event & Festival Information Aggregator App	Develop an application that aggregates and displays community event and festival schedules for a town or ward.	6	11
57	Community Sports Ground / Facility Booking App	Develop an application that enables local users to check availability and book shared sports grounds or courts.	6	3, 11



58	Talent Scouting Platform for Rural Sports Players	Develop a platform where local sports talent can create profiles and gain visibility among coaches and scouts.	6	3
59	Youth Skill Development and Internship Finder App	Develop an application listing local internships, apprenticeships, and skill-training programs to help youth identify relevant opportunities.	6	4, 8
60	Financial Literacy Awareness App for Rural Households	Develop an application that teaches basic banking, savings, and fraud-avoidance concepts with local-language support.	6	4, 8
61	Legal Aid & Rights Awareness Portal	Develop a portal listing free legal-aid contacts and providing simplified explanations of common citizen legal rights.	6	16
62	Digital Insurance / Pension Scheme Awareness Tool	Develop a tool that helps unorganized-sector workers check eligibility and access information for government insurance and pension schemes.	5, 6	8
63	Livestock Health Record and Vaccination Tracker	Develop an application to record livestock health history and manage vaccination schedules for farmers.	6	15
64	Soil Testing Appointment & Report Access Portal	Develop a portal that enables farmers to book soil-testing appointments and access their digital soil-test reports.	6	15
65	Power Outage Reporting and Tracking App for a Locality	Develop an application that enables residents to report power outages and track the status of reported issues with the local utility service.	6	7, 11
66	Community Borewell / Water Tanker Scheduling App	Develop an application to schedule, monitor, and track water-tanker deliveries for water-scarce localities.	6	6, 11
67	Cyber Safety and Digital Fraud Awareness App	Develop an application that educates students and senior citizens about common online scams, digital fraud, and safe digital practices.	6	16
68	Adult Literacy Program Registration and Progress Tracker	Develop an application to register learners and track attendance, participation, and learning progress in community adult-literacy programs.	6	4
69	Inclusive Learning Content Tool for Differently Abled Students	Develop a tool delivering accessible audio/visual learning content adapted for students with visual or hearing impairments.	5, 6	4
70	School Dropout Tracking and Follow-up System	Develop an application for teachers to record attendance trends, identify students at risk of dropping out, and track follow-up activities.	6	4
71	Public Park / Garden Maintenance Complaint Tracker	Develop an application that enables citizens to report maintenance issues in public parks and gardens and track their resolution status.	6	11, 15
72	Digital Suggestion Box for Local Governance	Develop an application that enables citizens to submit suggestions and feedback to local governing bodies on civic and community matters.	5, 6	16



73	Community Health Screening Camp Scheduler	Develop an application to schedule community health-screening camps and manage participant registration and attendance for basic health screenings.	6	3
74	Digital Queue Management System for Health Camps	Develop a token-based appointment and queue-management system to organize patient registration, appointments, and waiting queues for medical camps or Primary Health Centres (PHCs).	6	3
75	Streetlight Fault Reporting System	Develop an application that enables citizens to report faulty streetlights using photographs and GPS-based location information and track the status of reported faults.	6	11
76	Village Asset Mapping System	Develop a GIS-enabled or map-based application to document and visualize community assets such as schools, hospitals, water tanks, roads, and other public facilities.	6	11
77	Food Waste Reduction & Expiry Management System	Design and develop a web/mobile application to record food items, monitor expiry dates, generate expiry reminders, and maintain food inventory to help reduce food wastage.	6	2, 12
78	Community Hall Booking & Management System	Design and develop a web-based platform to manage community hall bookings, track availability, process booking requests, schedule maintenance, and maintain booking records.	6	11
79	Public Infrastructure Asset Monitoring System	Design and develop a web-based platform to maintain records of public infrastructure assets, report maintenance issues, assign maintenance requests, and track their resolution status.	6	9, 11

Teaching Learning Process:

The suggested TLP for Community / Societal Project.

The following structured sequence of activities is to be carried out during the semester.

Week / Phase	Teaching–Learning Activity	Faculty Role	Student Activity	Output / Evidence
1	Course introduction, objectives, ethics, conduct, and assessment briefing	Explain course scope, rules, rubrics, and expected deliverables	Understand course requirements and form teams	Team list, course plan
2	Community problem identification and topic selection	Guide topic framing and feasibility	Identify local issues through discussion and observation	Shortlisted project topic
3	Stakeholder interaction and preliminary survey	Demonstrate survey methods and stakeholder engagement	Collect initial data from community/users	Survey notes, interview records
4	Problem statement and objective formulation	Review and refine the problem statement	Prepare problem statement, scope, and objectives	Approved problem statement
5	Literature/case study review and baseline analysis	Suggest relevant references and examples	Study existing solutions and compare approaches	Baseline analysis summary
6	Project planning and role assignment	Help teams organize tasks, timeline, and responsibilities	Plan activities, divide roles, and prepare a schedule	Work plan / Gantt chart
7	Feasibility analysis and solution brainstorming	Check technical feasibility and practicality	Generate possible engineering solutions	Solution alternatives sheet
8	Selection of final solution and design approach	Review final concept and design logic	Finalize concept, methods, and expected outcomes	Concept note / design outline
9	Data processing, calculations, sizing, and mapping	Support computations and conduct technical review	Analyze data, perform calculations, and prepare maps/charts	Technical analysis sheet
10	Prototype / model / audit / DPR / application development	Monitor progress and address technical issues	Develop the selected project output	Working model / draft DPR / map / application
11	Testing, refinement, and implementation planning	Evaluate technical correctness and suggest improvements	Test the prototype or validate the proposal with users	Test results / revised output



1 2	Field implementation / demonstration / awareness activity	Facilitate community interaction and documentation	Conduct implementation, demonstration, or awareness activity	Photographs, attendance, implementation record
1 3	Documentation of outcomes, feedback, and impact	Review report structure and completeness	Compile findings, stakeholder feedback, and observations	Draft report
1 4	Presentation preparation and viva practice	Conduct mock presentation and provide feedback	Prepare slides/poster and rehearse presentation	Presentation deck
1 5	Final presentation, submission, and reflection	Evaluate performance using prescribed rubrics	Present project, submit report, and reflect on learning	Final report, presentation, viva

Assessment Structure

A. Continuous Internal Evaluation (CIE) – 50 Marks

- Students shall work in **batches of 3–4** and undertake at least **one project**, or a combination of **two related activities**, preferably leading to a **logical implemented or implementable outcome**.
- CIE shall be conducted by a **Departmental Committee** comprising:
 - Head of the Department / Nominee
 - Project Guide
 - One Senior Faculty Member
- The evaluation shall be based on prescribed rubrics carrying a total of **50 marks**.
- CIE Pass Requirement:** A student must secure a minimum of **20 marks out of 50** in CIE to become eligible to appear for the SEE.
- Students who do not secure the required CIE marks may be provided **additional opportunities before the closure of the semester**.
- No re-doing of CIE shall be permitted after the end of the semester.**

B. Semester End Examination (SEE) – 100 Marks, Scaled to 50

- Duration:** 2 Hours
- Maximum Marks:** 100
- The SEE marks shall be **entered as 100 in the VTU portal**. The **ITI-SMU module will automatically scale the marks from 100 to 50** for result processing.
- The SEE shall be conducted **batch-wise** by a committee comprising:
 - Project Guide
 - One Faculty Member from the Same Department
 - One Faculty Member from a Different Department
- SEE Pass Requirement:**
 - For a pass in SEE, a student shall secure minimum 35 marks out of 100.
 - For declaration of a pass grade, the sum of CIE marks (out of 50) and scaled SEE marks (out of 50) shall be ≥ 40 ; otherwise, the student is awarded grade F (Fail). A student must secure a minimum of **35 marks out of 100** in the SEE.



CIE Rubric — Continuous Internal Evaluation (Total 50 marks):

Criterion	Weight (Marks)	Excellent – Full Marks	Good	Fair	Needs Improvement
Problem identification & relevance to society/environment	8	Problem is clearly identified with strong evidence of societal/environmental relevance and a well-defined scope.	Relevant problem with clear scope and adequate supporting evidence.	Problem is defined, but justification of relevance is weak.	Problem is unclear or lacks societal/environmental relevance.
Problem statement, engineering approach, innovation & feasibility	10	Precise problem statement with a sound, innovative, technically feasible engineering approach.	Adequate problem statement with a feasible approach and minor gaps.	Partially defined problem statement with marginal feasibility.	Problem statement is missing, unclear, or the proposed approach is infeasible.
Stakeholder interaction, data collection, analysis & interpretation	8	Thorough stakeholder engagement with comprehensive data collection and well-interpreted analysis.	Good stakeholder engagement with adequate data collection and analysis.	Limited stakeholder engagement with incomplete data and superficial analysis.	Little or no stakeholder interaction; inadequate or absent data/analysis.
Project documentation, clarity, organization & demonstrated problem-solving	10	Complete, well-organized documentation demonstrating strong analytical thinking and effective problem-solving.	Complete and clear documentation with some analytical content and evidence of problem-solving.	Basic documentation with omissions and limited evidence of problem-solving.	Documentation is missing or poorly organized, with no clear evidence of problem-solving.



Communication & Presentation Skills	6	Confident and well-structured presentation; communicates ideas clearly, uses effective visuals, and responds comprehensively to questions.	Clear and satisfactory presentation; responds appropriately to most questions.	Acceptable presentation but lacks clarity or confidence; responses to questions are weak.	Poor or ineffective presentation; unable to respond adequately to questions.
Contribution & Benefit to Society (Impact & Sustainability)	8	Demonstrates clear and measurable societal benefit with a realistic and well-defined sustainability pathway.	Demonstrates plausible societal benefit with sustainability considerations and minor gaps.	Societal benefit is limited or unclear; sustainability aspects are weak.	No evident societal benefit or consideration of sustainability.
Total	50				

Notes for CIE rubric:

- Minimum CIE pass requirement: 20/50 to be eligible for SEE.

SEE Rubric — Semester End Examination (Enter marks out of 100; scaled to 50 later):

Criterion	Weight (Marks)	Excellent – Full Marks	Good	Fair	Needs Improvement
1. Problem identification & relevance to society/environment	16	Problem is clearly articulated with strong evidence, context, and demonstrated societal/environmental relevance.	Problem is clearly stated with reasonable supporting evidence and context.	Problem is stated, but supporting evidence or context is weak.	Problem is poorly defined or lacks relevant societal/environmental significance.
2. Problem statement, engineering approach, innovation & feasibility	20	Precise problem statement demonstrating strong engineering depth, novelty, innovation, and technical feasibility.	Technically sound and feasible approach with minor limitations.	Basic engineering approach with limited technical depth or feasibility concerns.	No viable engineering approach or proposed plan is technically infeasible.



3. Stakeholder interaction, data collection, analysis & interpretation	16	Excellent stakeholder engagement, rigorous data collection, and insightful analysis and interpretation.	Good stakeholder engagement with adequate data collection and acceptable analysis.	Superficial data collection with limited analysis and interpretation.	Inadequate or no stakeholder interaction/data collection; poor or absent analysis.
4. Project documentation, clarity, technical rigor & problem-solving	20	Comprehensive, technically rigorous, systematic documentation demonstrating strong problem-solving.	Complete and technically correct documentation with minor gaps.	Documentation lacks technical depth or contains notable omissions.	Key elements are missing or the documentation is technically incorrect.
5. Communication, presentation & viva-voce responses	12	Presents the work confidently and professionally, defends the approach persuasively, and answers technical queries accurately.	Good presentation with satisfactory responses to most queries.	Adequate presentation but unable to answer some technical questions effectively.	Poor presentation and unable to adequately defend or explain the work.
6. Contribution & benefit to society; impact assessment & sustainability	16	Provides a clear, evidence-based assessment of societal impact, sustainability, and realistic recommendations for scaling the solution.	Provides a reasonable discussion of impact with appropriate sustainability considerations.	Provides limited evaluation of impact or sustainability.	Provides no coherent assessment of societal impact or sustainability.
Total	100				



SEE Pass Rule and Implementation Guidelines

SEE Pass Rule

- A student must secure a **minimum of 35 marks out of 100 in the SEE itself** to be eligible for pass consideration.
- The SEE marks are **scaled from 100 to 50** for final result processing.
- The **final passing requirement** is based on the combined CIE and scaled SEE marks:

Final Marks = CIE (out of 50) + SEE (scaled to 50)

- A student shall be considered **passed only if**:
 - **CIE $\geq 20/50$**
 - **SEE $\geq 35/100$**
 - **CIE + Scaled SEE $\geq 40/100$**
- If any of the above requirements is not satisfied, the student shall be awarded **Grade F (Fail)**.

Suggested Implementation Notes for BoS / Examination Committees

1. **Rubric-based evaluation:** Use numerical scores for each rubric criterion and maintain brief evaluator comments to justify the awarded marks.
2. **Individual contribution in team projects:** For team-based projects, individual contributions shall be documented. Individual marks may be adjusted based on a **peer-evaluation sheet and/or project guide's assessment**.
3. **Maintenance of project evidence:** Relevant supporting evidence such as **field notes, photographs, questionnaires, stakeholder interaction records, DPRs, prototypes, maps, reports, and implementation records** shall be maintained as annexures for SEE evaluation and moderation.
4. **Standardized evaluation sheet:** Use a standardized **spreadsheet-based score sheet** with separate columns for each rubric criterion to facilitate transparent mark entry, automatic aggregation, verification, and archival.
5. **Moderation and record keeping:** The department shall maintain the completed evaluation sheets and supporting evidence for **academic audit, moderation, and accreditation requirements**.



Suggested Learning Resources

Reference Books

1. Phadke, N. D., *Community Engagement and Social Innovation*, Oxford University Press, New Delhi.
2. Garg, N. R., *Project-Based Learning in Engineering Education*, Pearson India Education Services, Noida.
3. Gupta, A. and P. Singh, *Service Learning and Community Projects*, McGraw-Hill Education (India), New Delhi.
4. Wilson, G., *Practical Guide to Community Development Projects*, Routledge, London.

Web Links and Video Lectures (e-Resources)

1. **VTU e-Learning Centre:** <http://elearning.vtu.ac.in>
2. **NPTEL:** Courses and lectures related to project-based learning, design thinking, social innovation, and engineering education.
3. **SWAYAM:** Courses related to community engagement, sustainability, social innovation, and project-based learning.
4. **United Nations Sustainable Development Goals (SDGs):** Resources on SDG targets, indicators, and community-level initiatives.
5. **MyGov India and Local Civic Body Portals:** Information on government initiatives, citizen engagement, and community development programs.
6. **YouTube Educational Resources:** Lectures and demonstrations on design thinking, social innovation, community engagement, and service learning.
7. **NGO and Government Case Studies:** Field-based case studies related to sanitation, water management, healthcare, education, livelihood, and environmental sustainability.

Assessment Structure

Continuous Internal Evaluation (CIE) shall be conducted by the **Departmental Project Review Committee**, comprising:

- Senior Professor / Faculty Member nominated by the Head of the Department
- Project Guide
- One additional faculty member nominated by the Head of the Department

Semester End Examination (SEE) shall be conducted **batch-wise by the prescribed examination committee/university-appointed examiners, as applicable under VTU regulations.**

The evaluation shall be based on **standardized rubrics** designed to assess the attainment of relevant **NBA Graduate Attributes and Program Outcomes**, including:

- Problem analysis
- Investigation and data analysis
- Design/development of engineering solutions
- Innovation and feasibility
- Communication
- Teamwork and individual contribution
- Professionalism and ethical responsibility
- Societal and environmental impact
- Sustainability



The weightage of CIE and SEE shall be 50% each.

Pass Criteria

A candidate shall satisfy **all applicable minimum requirements prescribed by VTU** for Project Work:

1. Secure at least **20/50 in CIE**.
2. Secure at least **35/100 in SEE**.
3. Secure a minimum **aggregate of 40/100**, considering CIE (50) and SEE scaled to 50.
4. If any mandatory minimum requirement is not satisfied, the candidate shall be awarded **Grade F (Fail)**, irrespective of the aggregate score.

This structure provides a clear alignment between **project-based learning, community engagement, engineering problem-solving, NBA graduate attributes, and VTU assessment requirements**.



Final Submission

(Weeks 13–14)

Documents to be Submitted

1. Final Project Report

The report should contain:

- Title Page
- Certificate
- Acknowledgement
- Abstract
- Table of Contents
- List of Figures/Tables
- Introduction
- Community Need Assessment
- Problem Statement
- Objectives
- Methodology
- Design and Development
- Implementation Details
- Results and Outcomes
- Impact Assessment
- Sustainability Analysis
- Challenges and Limitations
- Conclusion and Future Scope
- References
- Annexures

Recommended Length: 25–40 pages

2. Project Poster

Poster should include:

- Project Title
- Team Details
- Problem Statement
- Objectives
- Methodology
- Solution Developed
- Results
- Impact on Community
- Photographs



3. Presentation Slides

Recommended Contents:

- Introduction
- Community Need
- Problem Statement
- Objectives
- Methodology
- Solution
- Implementation
- Impact Assessment
- Conclusion

Recommended Slides: 10–15