Probability, Distributions and Statistics			
Course Code	**1BMATCS301**	Scheme	2025
Type of Course	ASC	Semester	3
Teaching Hours/Week (L:T:P)	3:2:0	CIE Marks	50
Total Hours of Pedagogy per semester L :T:P:SL&TW:TH	42:28:0:50:120	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 the concepts of modular arithmetic, congruences, linear Diophantine equations, Euler's theorem, Fermat's theorem, Wilson's theorem, and the RSA cryptographic algorithm to solve engineering and computational problems.
2. Apply least squares curve fitting, correlation, regression analysis, and rank correlation techniques to analyse engineering data and establish relationships between variables.
3. Solve engineering problems involving discrete and continuous random variables using probability distributions, mathematical expectation, variance, and standard probability models such as Binomial, Poisson, and Normal distributions.
4. Analyse joint probability distributions and Markov chains by determining expectations, covariance, correlation, transition probabilities, stationary distributions, and absorbing states for stochastic systems.
5. Apply statistical inference techniques including hypothesis testing, confidence intervals, large and small sample tests, Student's t-test, Chi-square test, and the Central Limit Theorem to draw valid conclusions from engineering data.

Module-1: Modular Arithmetic

Introduction of modular arithmetic and its applications in Computer Science and Engineering. Introduction to Congruences, Linear Congruences, The Remainder theorem, Solving Polynomials, Linear Diophantine Equation, System of Linear Congruences, Euler's Theorem, Wilson Theorem and Fermat's little theorem. Applications of Congruences-RSA algorithm.

[Textbook-3: Chapters 4, 5, 6, 7 and 9]

Number of Hours: 8

Module-2: Statistics

Principles of least squares, Curve fitting by the method of least squares in the form: $y = a + bx$, $y = a + bx + cx^2$, and $y = ax^b$. Correlation, Coefficient of correlation, Lines of regression, Angle between regression lines, rank correlation.

[Textbook-1: Chapters 11 and 12, Textbook-2: Chapters 13 and 14]

Number of Hours: 8

Module-3: Probability Distributions

Review of basic probability theory. Random variables (discrete and continuous), probability mass and density functions. Mathematical expectation: mean and variance. Binomial, Poisson and normal distributions-problems (derivations for mean and standard deviation for Binomial and Poisson distributions only).

[Textbook-1: Chapter -3, 4, 5, 6, Textbook-2: Chapters 6 and 7]

Number of Hours: 9

Module-4: Joint probability distribution

Joint Probability distribution for two discrete random variables, expectation, covariance and correlation.

Markov Chain: Introduction to Stochastic Process, Probability Vectors, Stochastic matrices, Regular stochastic matrices, Markov chains, Higher transition probabilities, Stationary distribution of Regular Markov chains.

[Textbook-1: Chapter -3, 4, 5, 6, Textbook-4: Chapter-8] Number of Hours: 8

Module-5: Statistical Inference

Sampling distribution, standard error. Levels of significance. Testing of hypothesis. Test of significances. Confidence limits, simple sampling of attributes. Sampling variables, central limit theorem and confidences limit for unknown mean.

Test of significance for large samples (mean, proportion). Comparison of large samples (mean, proportion).

Test of Significance for means of two small samples, students-'t' distribution, and Chi-square distribution as a test of goodness of fit.

[Textbook-1: Chapter 8 and 10, Textbook-2: Chapters-8, 9, 10 and 12]

Number of Hours: 9

TUTORIAL COMPONENT OF PCC

Tutorial Topic/Problem solving/Case ... etc. (for 2 hours each)**:** Number of Hours: 28

1. Practice problems on linear congruences
2. Practice Problems on Linear Diophantine Equation, System of Linear Congruences
3. Practice problems on Euler's Theorem, Wilson Theorem and Fermat's little theorem.
4. Practice problems on fitting linear form and quadratic form
5. Practice problems on $y = ax^b$ and correlation
6. Practice problems on lines of regression and rank correlation
7. Practice problems on mathematical expectations
8. Practice problems on Binomial and Poisson distributions
9. Practice Problems on Normal distributions
10. Practice problems on covariance, correlation of two discrete random variables
11. Practice problems on Markov chain, stochastic matrices, probability vectors
12. Practice Problems on higher transition probabilities and stationary distributions
13. Practice Problems on test of significance for large samples
14. Practice problems on test of significance for small samples

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

Textbooks:

1. Ronald E. Walpole, Raymond H Myers, Sharon L Myers & Keying Ye "Probability and Statistics for Engineers and Scientists", Pearson Education, 9th edition, 2012.
2. Murray R. Spiegel, Larry J. Stephens, Statistics, Schaum's outline series, 6th edition,
3. Koshy, Thomas. *Elementary number theory with applications*. Academic press, 2nd Edition, 2009.
4. Seymour Lipschutz and Marc Lars Lipson, Probability, Schaum's Outline Series, Second edition, 2011.

Reference books / Manuals:

1. Erwin Kreyszig, "Advanced Engineering Mathematics", John Wiley & Sons, 11 th Edition, 2006.
2. B. S. Grewal "Higher Engineering Mathematics", Khanna publishers, 45 th Ed., 2021.
3. Irwin Miller & Marylees Miller, John E. Freund's "Mathematical Statistics with Applications" Pearson. Dorling Kindersley Pvt. Ltd. India, 8th edition, 2014.
4. S C Gupta and V K Kapoor, "Fundamentals of Mathematical Statistics", S Chand and Company, 12th edition.
5. P. G. Hoel, S. C. Port and C. J. Stone, "Introduction to Probability Theory", Universal Book Stall, (Reprint), 2003.
6. S. Ross, "A First Course in Probability", Pearson Education India, 6th Ed., 2002.

7. David M Burton: "Elementary Number Theory" Mc Graw Hill, 7th Ed.,2017

Web links and e-Resources:

1. Probability and statistics: https://nptel.ac.in/courses/111105090
2. Introduction to Probability and Statistics for Machine Learning: https://www.classcentral.com/course/codesignal-introduction-to-probability-and-statistics-for-machine-learning-358081
3. Probability and Statistics: https://www.classcentral.com/course/probability-statistics-11755 Introduction to probability, statistics, and random processes: https://probabilitycourse.com/
4. Introduction to Probability and Statistics: https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2022/
5. VTU e-Shikshana Program
6. Probability, Distributions and Statistics: https://www.geeksforgeeks.org/maths/maths/

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.

- **Flipped Classroom Approach**
  - Students review video lectures and reading materials before class.
  - Classroom sessions focus on problem-solving, discussions, and applications of transform techniques and optimization models.
- **Problem-Based Learning (PBL)**
  - Real-world engineering problems involving Fourier Transforms, Z-Transforms, and Linear Programming are assigned for collaborative solution development.
- **ICT-Enabled Teaching** (may be conducted in Tutorial hours)
  - Use of MATLAB, Python, and online simulation tools to visualize transforms, optimization problems, and signal-processing applications.
- **Collaborative Learning (** in Tutorial hours).

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	Problem-solving assignment at RBTL-3/4	15
2	Solving GATE questions related to the syllabus content	10

Rubrics for Learning Activity–1: Problem-solving assignment at RBTL-3/4

Criteria/Scale	Superior (5)	Good (4)	Fair (2)	Needs Improvement (1)
Understanding of Concepts	Demonstrates excellent conceptual clarity and application	Good understanding with minor errors	Basic understanding	Poor understanding
Problem-Solving Ability	Solves all problems correctly with proper methodology	Solves most problems correctly	Partial solutions with errors	Unable to solve problems effectively
Mathematical Accuracy	Calculations and derivations are error-free	Minor computational errors	Several errors	Significant errors

Rubrics for Learning Activity–2: Solving GATE questions related to the syllabus content

Criteria/Scale	Superior (5)	Good (4)	Fair (2)	Needs Improvement (1)
Analytical Thinking	Excellent logical reasoning and interpretation	Good reasoning ability	Moderate reasoning	Weak reasoning
Accuracy & Precision	Very high accuracy	Good accuracy	Moderate accuracy	Low accuracy

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 activities etc.)

Sl. No.	Term Work (TW) Activity	Number of Hours/ Semester
1	Learning Activity-1 (Problem-solving assignment)	10
2	Learning Activity-2 (Solving GATE questions)	10
3	MATLAB/Python computations for the problems related to RSA algorithms, Probability, Statistics and Sampling Distribution concepts.	10

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

Sl. No.	Self-Learning (SL) Activity	Number of Hours / Semester
1	RSA Algorithm in Cryptography (https://www.geeksforgeeks.org/computer-networks/rsa-algorithm-cryptography/)	3

2	Real-Life Applications of Continuous Probability Distribution (https://www.geeksforgeeks.org/maths/real-life-applications-of-continuous-probability-distribution/)	2
3	Application of RSA algorithms, Probability, Statistics and Sampling Distribution concepts.	10
4	Industrial applications of statistics, probability, sampling distributions and congruences.	5

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).

OBJECT ORIENTED PROGRAMMING WITH JAVA			
Course Code	**1BCS302**	Scheme	2025
Type of course	IPCC	Semester	3
Teaching Hours/Week (L:T:P)	3:0:2	CIE Marks	50
Total Hours of Pedagogy L:T:P:TW&SL:TH	42:0:28:50:120	SEE Marks	50
Credits	4	Total Marks	100
Type of Examination (IPCC):	Theory- CIE +SEE, and Lab- CIE only.	Exam Hours	03

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 autoboxing, enumerations, and generics in program development.
6. Choose a suitable IDE tool to design, develop, and debug Java programs.

Module-1

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: The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, 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).

Textbook: Chapters 2, 3, 4, 5 Number of Hours: 9

Module-2

Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, the 'this' Keyword, and Garbage Collection.

Methods and Classes: Overloading Methods, Using Objects as Parameters, Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes.

Textbook: Chapters 6, 7 Number of Hours: 8

Module-3

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, Local Variable Type Inference and Inheritance, The Object Class.

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

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

Textbook: Chapters 8, 9 Number of Hours: 9

Module-4

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, Chained Exceptions.

Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities,Obtaining a Thread's State.

Textbook: Chapters 10, 11 Number of Hours: 8

Module-5

Enumerations, Type Wrappers and Autoboxing: Enumerations (Enumeration Fundamentals, The values() and valueOf() Methods), Type Wrappers (Character, Boolean, The Numeric Type Wrappers), Autoboxing (Autoboxing and Methods, Autoboxing/Unboxing Occurs in Expressions, Autoboxing/Unboxing Boolean and Character Values).

Generics: A Simple Generics Example, A Generic Class with Two Type Parameters, The General Form of a Generic Class.

Textbook: Chapters 12, 14 Number of Hours: 8

PRACTICAL COMPONENT OF IPCC

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 read the marks of three subjects as strings, convert them into wrapper objects, calculate the total and average using autoboxing/unboxing and display the result.
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: (Text Book/ Reference Book/ Manuals):

Textbooks:

1. 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 Video Lectures (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 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).

Rubrics for CIE of Practical Component:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (1)
Fundamental Knowledge: Understanding the problem statement [CO1] [PO1, PO2]	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 [CO2-6] [PO2, PO3]	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 properly.
Implementation using Java [CO2-6] [PO3, PO8, PO5]	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 completely.
Program debugging and testing with suitable tools [CO2-6] [PO5, PO8]	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 [CO1-6] [PO4, PO5]	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 properly.
Demonstration and documentation [CO1-6] [PO8, PO9, PO11]	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)	Marginal (1)
Fundamental Knowledge (2) [CO1] [PO1]	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) [CO2-6] [PO2]	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 clearly.
Design and Implementation (3) [CO2-6] [PO3, PO5]	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 completely.
Result & Analysis (2) [CO2-6] [PO4, PO5]	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 com.
Communication (Viva voce) (2) [CO1-6] [PO8, PO9]	Good Verbal & nonverbal communicatio n skills with precise and correct terminologies/ answers.	Good verbal Communication skills with precise and correct terminologies/ans wers.	Average Communication but with precise and correct terminologies/an swers.	Average Communication but with imprecise and incorrect terminologies/answ ers.	Poor Communication (Minimal interaction/ans wers)

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 activities, etc.)		
Sl. No.	**Term Work (TW) Activity**	**Number of Hours / Semester**

1.	Installation and use of IDE for Java. Writing simple Java programs to understand Java program development and execution model.	04
SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.		
Sl. No.	**Self-Learning (SL) Activity**	**Number 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 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	**1BCS303**	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 and logic minimization techniques to design and simplify digital logic circuits using logic gates, K-maps, and NAND/NOR implementations.
2. Design combinational and sequential logic circuits using adders, decoders, encoders, multiplexers, latches, and flip-flops.
3. Describe the organization and operation of computer systems, including functional units, bus structures, instruction sets, and addressing modes.
4. Illustrate memory and input/output organization techniques, including interrupts, DMA, cache memory, and semiconductor memory systems.
5. Contrast the operation of processing units, hardwired control systems, and pipelining techniques to assess computer performance and efficiency.
6. Identify computer architecture components, memory organization, and system performance through experimentation and case studies

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, 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):

Textbooks:

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.

Reference books / Manuals:

1. Charles H Roth and Larry L Kinney, Analog and Digital Electronics, Cengage Learning
2. William Stallings: Computer Organization & Architecture, 11th 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

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 statements	25

List of Problem Statements:

NOTE: Make use of simulation packages like Xilinx Simulator Interface , Multisim, Modelsim, 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. Evaluation is for 25 marks.

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.

Rubrics for Learning Activity-1:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (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 properly.
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 suitably.
Simulation, Testing and Verification(CO 2, 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 properly.
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 drawn.
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 on time.

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-Termworks) 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			
Course Code	**1BCS304**	Scheme	2025
Type of Course	PCC	Semester	3
Teaching Hours/Week (L:T:P)	3:0:0	CIE Marks	50
Total Hours of Pedagogy per semester L:T:P:SL:TW&SL:TH	42:0:0:48:90	SEE Marks	50
Credits	3	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. Recall the structure, functions, and services of an operating system.
2. Explain process management concepts, CPU scheduling algorithms, and process synchronisation mechanisms.
3. Apply appropriate CPU scheduling algorithms and process synchronisation techniques to solve operating system problems.
4. Apply memory management techniques and deadlock-handling strategies to ensure efficient resource utilization.
5. Explain file system organisation and secondary storage management techniques.
6. Describe information protection mechanisms, security principles, and access control methods in operating systems.

Module-1

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) Number of Hours: 09

Module-2

Process Management: Process concept; Process scheduling; Operations on processes; Inter-process communication
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) Number of Hours: 08

Module-3

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) Number of Hours: 09

Module-4

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) Number of Hours: 08

Module-5

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) Number of Hours: 08

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

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles, 9th edition, Wiley-India, 2015.

Reference books / Manuals:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 7th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. William Stallings Operating Systems: Internals and Design Principles, 7th Edition, Pearson.

Web links and Video Lectures (e-Resources):

1. Introduction to Operating System: https://youtu.be/mXw9ruZaxzQ
2. Different Types of Operating Systems: https://youtu.be/vBURTt97EkA 3
3. Process Synchronisation in Operating System: https://www.youtube.com/watch?v=783KAB-tuE4&list=PLIemF3uozcAKTgsCIj82voMK3TMR0YE_f
4. Introduction to Unix: https://infyspringboard.onwingspan.com/web/en/login

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. Case-Based Teaching
2. Simulation and Virtual Labs:
   Operating System Lab: https://erpjietuniverse.in/virtual_lab/Operating_system/labs/index.html
3. ICT-Enabled Teaching
4. Peer teaching/learning

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.	Implementation of Operating System Concepts using C Programming	25

1. Implementation of Operating System Concepts using C Programming

List of Problems:

1. Develop a C program to implement the Process system calls (fork (), exec(), wait(), create process, terminate process)
2. Simulate the following CPU scheduling algorithms to find the turnaround time and waiting time

   a) FCFS b) SJF c) Round Robin d) Priority.

3. Develop a C program to simulate producer-consumer problem using semaphores.
4. Develop a C program that demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write, and close APIs in your program.
5. Develop a C program to simulate Banker's Algorithm for Deadlock Avoidance.
6. Develop a C program to simulate the following contiguous memory allocation techniques:

   a) Worst fit b) Best fit c) First fit.

7. Develop a C program to simulate page replacement algorithms:

   a) FIFO b) LRU

8. Simulate following file organization techniques:

   a) Single level directory b) Two level directory

9. Develop a C program to simulate the linked file allocation strategies.
10. Develop a C program to simulate SCAN disk scheduling algorithm.

Note: 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. Evaluation is for 25 marks.

Rubrics for Learning Activity-1:

Criteria/ Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (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 basic OS concepts .
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 suitably.
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 documentation issues.	Adequate documentation and readability.	Poor documentation and inconsistent coding style.	No suitable documentation and difficult to understand code.

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: Implementation of Operating System Concepts Using C Programming	30
SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.		
Sl. No.	**Self-Learning (SL) Activity**	**Number of Hours / Semester**
1	Springboard Certification on: Introduction to Unix	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 STRUCTURES AND APPLICATIONS			
Course Code	**1BCS305**	Scheme	2025
Type of Course	PCC	Semester	3
Teaching Hours/Week (L:T:P)	3:0:0	CIE Marks	50
Total Hours of Pedagogy per semester L:T:P:SL:TW&SL:TH	42:0:0:48:90	SEE Marks	50
Credits	3	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. 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 organise the data effectively.

Module-1

Introduction: Data Structures, Classifications (Primitive & Non-Primitive).

Pointers and Dynamic Memory Allocation, Arrays, Dynamically Allocated Arrays, Structures and Unions, Polynomials.

Sparse Matrices: Sparse Matrix Representation and Transposing a Matrix.

Textbook: Chapter-1 (1.2), Chapter-2 (2.1-2.4, 2.5.2, 2.5.3), Reference Book-1 (1.3) Number of Hours: 08

Module-2

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) Number of Hours: 08

Module-3

Circular Queues, Circular queues using Dynamic arrays, Multiple Stacks and Queues.

LINKED LISTS: Singly Linked Lists and Chains, Representing Chains in C, Linked Stacks and Queues, Additional List Operations, Sparse Matrices: Sparse Matrix Representation, Doubly Linked List.

Textbook: Chapter-3 (3.4, 3.7), Chapter-4 (4.1 to 4.3, 4.5, 4.7.1, 4.8) Number of Hours: 09

Module-4

Trees: Introduction-Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals – preorder, Inorder, postorder, Threaded Binary Trees.

Binary Search Trees – Definition, Searching, Insertion, Deletion, Counting Binary Trees.

Textbook: Chapter-5 (5.1.1, 5.2, 5.3.1-5.3.3, 5.5, 5.7.1- 5.7.4, 5.11.1) Number of Hours: 09

Module-5

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) Number of Hours: 08

Suggested Learning Resources: (Text Book/ 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 / Manuals:

1. Seymour Lipschutz, Data Structures Schaum's Outlines, 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 Video Lectures (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=KW0UvOW0XIo&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 Liftist 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)	Marginal (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 properly.
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 20 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 Improvement (2)	Marginal (1)
Understanding of Data Structure Concepts (CO1, CO2 /PO1,PO2)	Demonstrates comprehensive understanding of fundamental and advanced data structure concepts, terminology, and applications.	Demonstrates good understanding of concepts with minor gaps.	Understands basic concepts and terminology with limited depth.	Shows partial understanding and requires additional support.	Unable to demonstrate understanding of fundamental concepts properly.
Application of Data Structures and Algorithms (CO3, CO5 /PO1,PO2)	Effectively applies suitable data structures and algorithms to solve complex computational problems and assessments.	Applies appropriate concepts to solve problems with minor errors.	Applies basic data structure concepts to solve standard problems.	Applies concepts with significant guidance and limited accuracy.	Unable to apply concepts for problem-solving suitably.
Problem Solving Skills and Practical Application (CO4/PO4)	Demonstrates strong Problem Solving skills by implementing data structures through hands-on exercises and coding activities.	Provide solutions correctly with minor errors.	Provide basic solutions with limited optimization.	Requires significant assistance for Problem Solving tasks.	Unable to provide 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 or 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 STRUCTURES LABORATORY			
Course Code	**1BCSL306**	Scheme	2025
Type of Course	PCCL	Semester	3
Teaching Hours/Week (L:T:P)	0:0:2	CIE Marks	50
Total Hours of Pedagogy L:T:P:SL:TW	0:0:28:2:30	SEE Marks	50
Credits	1	Total Marks	100
Examination type (SEE)	Practical	Exam Hours	03

Course outcome (Course Skill Set)

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

1. Develop programs to implement fundamental and linear data structures and their operations for solving computational problems.
2. Develop programs using non-linear data structures and hashing techniques to solve computational problems.
3. Select appropriate data structure to organize, process, and retrieve data while solving real-world problems.
4. Build experimental documents involving data structures, algorithms, program design, implementation and results with 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 record (structure) titled 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. Build 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 C program for processing sparse matrices using an appropriate data structure. The program shall accept suitable sparse matrix data, use an appropriate representation, and perform operations such as addition and transpose. Demonstrate the correctness of the solution using suitable test cases.

3. Develop a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX >15).
   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 at any given point of time.
   e. Exit.

   Support structured program with appropriate functions for each of the above operations.

4. 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 a print job from the Printer Queue (delete).
   c. Display the number of print jobs currently waiting in the Queue.
   d. Demonstrate Queue overflow and underflow conditions.
   e. Exit.

5. Develop a Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes.
   a. Read the polynomial: degree of the polynomial, terms (coefficient and power).
   b. Represent the Polynomial in one variable based on degree.
   c. Find the sum of two polynomials POLY1 and POLY2 and store the result in POLYSUM.

   Support the program with appropriate functions for each of the above operations.

6. Develop a Program in C for the following:
   a. Create a Binary Tree (createTree) by inserting integer elements as keys entered by the user in level order.
   b. Implement the following functions:
      i. createTree() – Create a binary tree through key Insertions.
      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. Develop 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.

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. A compiler design team needs a tool to convert arithmetic expressions from infix notation to postfix notation for efficient code generation. The expression contains alphanumeric operands, operators (+, -, *, /, %, ^), and brackets: (,), {, }, [,]. Develop a C program that supports the following::

   a. Read an infix expression.

   b. Validate the expression for correct bracket nesting and syntax.

   c. Convert the valid infix expression to postfix notation.

   Note: An input expression is valid if:

   • Open brackets must be closed by the same type of brackets.

   • Open brackets must be closed in the correct order.

   • Every close bracket has a corresponding open bracket of the same type.

   • Operators must have valid operands on both sides.

2. A University wants to manage student records effectively. Each student record contains USN, Name, Semester (1-8), and Department. Choose suitable data structure (Singly Linked List/ Doubly Linked List/Circular Linked Lists) and develop a menu-driven C program to manage these records with the following operations:

   a. Create a list of student records (choose any insertion method: beginning /end).

   b. View all student records.

   c. Display student records for a given semester.

   d. Insert the new student in the list after the given student's name.

   e. Delete a student record (based on USN or Name)

   Note: For CIE Test and SEE, any ONE option from (c) to (e) may be asked along with options (a) & (b).

3. A manufacturing unit needs a system (repository) to manage machine parts used in assembly. Each part has:

   • Part Number (alphanumeric, e.g., "P1001", "BOLT23")

   • Part Name (e.g., "Bolt", "Wheel")

• Model Number (e.g., "M2024", "X100")

• Description (e.g., "Stainless steel ")

Develop a C program to implement the following three operations.

a. Add a new part to the inventory/repository

b. Remove an obsolete part (by part number or model number)

c. Search and display part details (by part number or model number)

All three operations happen frequently without any order. Choose suitable data structure that results in the best performance.

4. Develop a C program by selecting suitable data structure to manage hierarchical data/information having keys (numeric). The program shall provide appropriate operations such as insertion of information, traversal (display of all information), quick search for information based on the key. Demonstrate the solution using relevant test cases.

5. Develop a C program to store employee records having fields: emp-no, name, designation, department) using suitable Indexing mechanism. The program shall support record management operations such adding new employee, search for employee details based on emp-no, remove employee record based on emp-no, and display employee records based on department. Demonstrate the correctness of the implementation using appropriate test cases.

6. A transportation department wants to analyse connectivity between cities. Each city has a City Name and cities are interconnected (not necessarily all) with roads. Develop a menu-driven C program with the following operations:

a. Create a graph representing N cities with interconnecting roads.

b. Display the graph representation.

c. Print all cities reachable by road from a given city..

d. Check whether any two given cities are connected via road.

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

Text books:

1. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2023
2. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2023.

Reference books / Manuals:

1. A M Tenenbaum, Data Structures using C, PHI, 1989.
2. Seymour Lipschutz, Data Structures Schaum's Outlines, McGraw Hill, 2014.

Web links and Video Lectures (e-Resources):

1. GNU-GCC Documentation: https://gcc.gnu.org/
2. Virtual Labs on Data Structures-1: https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html
3. DSA Tutorial: https://www.geeksforgeeks.org/dsa/dsa-tutorial-learn-data-structures-and-algorithms/
4. NPTEL Course on DSA: https://nptel.ac.in/courses/106106130
5. 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.

1. Engineering tool usage for the conduction of experiment
2. Demonstration through ICT tools
3. 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)	Marginal (1)
Fundamental Knowledge: Understanding the problem statement and selection of appropriate data structure [CO1, CO2, CO3] [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 completely or identify an appropriate data structure.
Design of Algorithm and Data Structure [CO1, CO2, CO3] [PO2, PO3]	Develops multiple algorithmic solutions, compares them, and justifies the best design with appropriate data	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 perform data structure operations suitably.

	structure selection.				
Implementation (Program Coding) [CO1, CO2, CO3] [PO5, PO5]	Implements the selected data structure efficiently using appropriate C constructs with correct operations and modular programming practices.	Implements the selected data structure correctly and explains the implementation clearly.	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 partially.
Program Debugging and Testing [CO1, CO2, CO3] [PO4, PO5]	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, CO3] [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 [CO4] [PO8, PO9]	Demonstration is excellent. Lab record is well-organized with problem statement, algorithm, data structure representation, program, output, observations, and conclusions presented using proper formatting.	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.	The demonstration is poor. Lab record is incomplete, improperly formatted, or not submitted on time.

Rubrics for SEE / CIE Test:

Criteria /Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (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, CO3] [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, CO3] [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, CO3] [PO4]	Successfully executes the program using multiple test cases, validates all operations, and provides accurate analysis.	Executes the program correctly and provides good analysis.	Executes the program with limited test cases and basic analysis.	Executes the program but cannot properly analyze the results.	Unable to execute or analyze the program.
Communication (Viva Voce) (2) [CO1, CO2, CO3, CO4] [PO8, PO9]	Clearly explains the working of the selected data structure, algorithm, implementation, and application using correct	Explains the concepts with good communication and correct terminology.	Explains basic concepts with average communication skills.	Provides incomplete explanations with incorrect terminology.	Unable to explain the implementation or answer questions satisfactorily.

	technical terminology.				

- 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**.

Project Management (with Git)			
Course Code	**1BCSL307A**	Scheme	2025
Type of Course	AEC	Semester	3
Teaching Hours/Week (L:T: P)	0:0:2	CIE Marks	50
Total Hours of Pedagogy L :T:P:SL:TW	0:0:28:2:30	SEE Marks	50
Credits	1	Total Marks	100
Type of Examination	Practical	Exam Hours	03

Course outcome (Course Skill Set)

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

1. Demonstrate the use of basic Git repository commands for version control operations.
2. Implement branch management strategies for collaborative software development
3. Apply collaboration techniques using remote repositories and distributed version control systems.
4. Configure Git tags, releases, and advanced Git operations for effective project management.
5. Modify Git history to maintain repository integrity and support project requirements.

SET OF EXPERIMENTS

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. Analysing 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. Analysing 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. Analysing and Changing Git History:

Write the command to display the last five commits in the repository's history.

12. Analysing and Changing Git History:

Write the command to undo the changes introduced by the commit with the ID "abc123".

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

Textbooks:

1. Version Control with Git, 3rd Edition, by Prem Kumar Ponuthorai, Jon Loeliger Released October 2022, Publisher(s): O'Reilly Media, Inc.
2. Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, https://gitscm.com/book/en/v2

Reference books / Manuals:

1. Git Pocket Guide, by Richard E. Silverman, https://www.oreilly.com/library/view/git-pocket-guide/9781449327507/

Web links and Video Lectures (e-Resources):

1. Git and GitHub Fundamentals (Infosys Springboard): https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01309444334736998478 2_shared/overview
2. Version Control Using Git (Infosys Springboard): https://infyspringboard.onwingspan.com/web/en/login?ref=%2Fapp%2Ftoc%2Flex_auth_01330134712177459211926_share%2Foverview
3. Pro Git (Official Git Documentation): https://git-scm.com/book/en/v2
4. Introduction to Git and GitHub (Coursera Google): https://www.coursera.org/learn/introduction-git-github

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. Case Studies and Problem-Solving Activities
2. Tool Demonstration and Programming
3. GitHub/GitLab Integration

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.

Rubrics for CIE – Continuous assessment:

Criteria/Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (1)
Git Fundamentals and Repository Operations [CO1] [PO1, PO5]	Demonstrates comprehensive understanding of Git concepts, repositories, commits, and version control.	Demonstrates good understanding of Git concepts with minor gaps.	Demonstrates basic understanding of Git concepts.	Limited understanding of Git concepts.	Unable to properly explain fundamental Git concepts.
Branching and Workflow Management [CO2] [PO2, PO3]	Independently creates, manages, and merges branches correctly.	Performs branch operations with minor guidance.	Performs basic branch operations with some errors.	Requires significant assistance for branch management.	Unable to perform branch operations properly.
Collaboration and Remote Repository Management [CO3] [PO2, PO5, PO9]	Efficiently performs clone, fetch, pull, push, and rebase operations.	Performs most remote repository operations correctly.	Performs basic collaboration tasks with assistance.	Difficulty in executing remote repository commands.	Unable to use remote repositories properly.
Tags, Releases and Advanced Git Operations [CO4] [PO4, PO11]	Correctly applies tags, stash, cherry-pick, and advanced Git operations.	Performs advanced operations with minor mistakes.	Performs some advanced operations with guidance.	Limited ability to perform advanced operations.	Unable to perform advanced Git operations properly
Git History Analysis and Repository Maintenance [CO5] [PO2, PO4,PO5, PO11]	Accurately analyses commit history and apply history modification commands.	Performs history analysis with minor errors.	Performs basic history viewing commands.	Requires assistance to analyse history.	Unable to analyse Git history properly.
Demonstration and documentation [CO1-CO5] [PO10, PO11]	Demonstrates all experiments confidently and maintains a complete, well-organized, and accurate lab record.	Demonstrates most experiments correctly and well-organized lab record with minor omissions.	Demonstrates basic experiments and maintains an adequate lab record with some missing details.	Demonstrates limited understanding of experiments and maintains an incomplete or poorly organized lab record.	Unable to demonstrate experiments effectively and fails to maintain a proper lab record.

Rubrics for SEE / CIE Test:

Criteria/Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (1)
Git Fundamentals and Repository Operations (2) [CO1] [PO1, PO5]	Demonstrates excellent understanding of Git concepts and workflows.	Demonstrates good understanding of Git concepts with minor gaps.	Demonstrates basic understanding of Git concepts.	Demonstrates limited understanding of Git concepts.	Unable to explain basic Git concepts properly.
Understanding of Problem Definition (1) [CO1] [PO2]	Clearly understands and accurately interprets the given Git task and requirements.	Understands the task with minor clarification.	Demonstrates basic understanding of the task.	Shows limited understanding of the task requirements.	Unable to understand the given task or requirements properly.
Design and Implementation (3) [CO2, CO3, CO4] [PO3, PO5]	Designs and implements Git workflows and operations accurately and efficiently.	Implements Git workflows with minor errors.	Demonstrates basic implementation of Git operations.	Requires significant assistance for implementation	Unable to implement the required Git operations minimally.
Result & Analysis (2) [CO5] [PO4. PO5]	Accurately analyses results and draws appropriate conclusions.	Analyse results with minor errors.	Demonstrates basic result interpretation.	Shows limited ability to analyse results.	Unable to interpret or analyse results properly.
Communication (Viva voce) (2) [CO3] [PO8, PO9]	Communicates ideas, concepts, and solutions clearly and confidently during viva.	Communicates concepts effectively with minor errors.	Demonstrates basic communication and understanding.	Communicates with difficulty and requires frequent prompting.	Unable to communicate concepts or answer viva questions properly.

- 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**.

EXPLORATORY DATA ANALYSIS			
Course Code	**1BAIL307A**	Scheme	2025
Type of Course	AEC	Semester	3
Teaching Hours/Week (L :T : P)	0:0:2	CIE Marks	50
Total Hours of Pedagogy L:T:P:SL:TW&SL:TH	0:0:28:2:30	SEE Marks	50
Credits	1	Total Marks	100
Type of Examination	Practical	Exam Hours	03

Course outcome (Course Skill Set)

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

1. Compare real-world datasets having different dataset structures and characteristics using Jupyter Notebook, NumPy, and Pandas.
2. Apply descriptive statistical techniques to summarize, analyze, and interpret data for exploratory analysis.
3. Make use of appropriate preprocessing techniques to handle data quality issues such as missing values, duplicate records, and inconsistencies.
4. Perform data filtering, sorting, grouping, and aggregation operations to extract meaningful information from datasets.
5. Interpret visualisations using Matplotlib and Seaborn to identify trends, patterns, and relationships in data.
6. Communicate insights through reports and visual presentations of end-to-end Exploratory Data Analysis (EDA).

SET OF EXPERIMENTS

1. Load a real-world dataset such as Student Performance, Employee Information, or Sales Data using Pandas in Jupyter Notebook. Explore the dataset by displaying the first few records, identifying the number of rows and columns, examining data types, and generating a summary of the dataset to understand its structure and contents.
    a. Expected Input: CSV dataset containing records and attributes.
    b. Expected Output: Display of dataset records, dimensions, column names, data types, and summary information.
    c. Learning Outcome: Ability to load and explore datasets using Pandas and understand the structure of real-world data.

2. Extract numerical attributes from a dataset and perform basic statistical analysis using NumPy and Pandas. Compute measures such as mean, median, mode, standard deviation, minimum, and maximum values, and interpret the results to understand the distribution and characteristics of the data.
    a. Expected Input: Dataset containing numerical attributes.
    b. Expected Output: Statistical summary showing central tendency and dispersion measures.
    c. Learning Outcome: Ability to perform basic statistical analysis and interpret numerical data characteristics.

3. Analyze the quality of a dataset by identifying missing values, counting null entries in each attribute, and calculating the percentage of missing data. Generate a simple data quality report highlighting attributes that require cleaning before further analysis.
    a. Expected Input: Dataset containing missing or incomplete records.
    b. Expected Output: Missing value count, percentage of missing data, and data quality report.
    c. Learning Outcome: Ability to assess data quality and identify issues affecting analysis.

4. Perform data cleaning by handling missing values using suitable techniques such as replacement with mean, median, or mode values and removal of incomplete records. Compare the dataset before and after cleaning and analyze the impact of the cleaning process.
    a. Expected Input: Dataset containing missing values.
    b. Expected Output: Cleaned dataset and comparison report before and after preprocessing.

   c. Learning Outcome: Ability to apply data cleaning techniques and prepare datasets for analysis.

5. Detect and remove duplicate records from a dataset using Pandas functions. Compare the number of records before and after duplicate removal and discuss the importance of maintaining data consistency in analytical applications.
   a. Expected Input: Dataset containing duplicate records.
   b. Expected Output: Duplicate record count and cleaned dataset without duplicate entries.
   c. Learning Outcome: Ability to identify and eliminate duplicate records to improve data quality.

6. Perform data filtering, sorting, and selection operations on a dataset to answer business-oriented questions such as identifying high-performing students, top-selling products, or employees with the highest salaries. Present the extracted information in a meaningful format.
   a. Expected Input: Dataset with categorical and numerical attributes.
   b. Expected Output: Filtered and sorted data satisfying specified conditions.
   c. Learning Outcome: Ability to retrieve and analyze relevant information from large datasets using query operations.

7. Apply grouping and aggregation techniques using Pandas to generate summaries from a dataset. Compute category-wise counts, averages, totals, and other statistical measures to derive meaningful insights from grouped data.
   a. Expected Input: Dataset containing categorical and numerical attributes.
   b. Expected Output: Group-wise summary tables and aggregated results.
   c. Learning Outcome: Ability to summarize and analyze data using grouping and aggregation techniques.

8. Conduct exploratory statistical analysis on a dataset by generating descriptive statistics and identifying patterns, variations, and unusual observations. Interpret the results to understand the overall behavior of the dataset and support data-driven decision-making.
   a. Expected Input: Real-world dataset.
   b. Expected Output: Statistical report highlighting key observations and trends.
   c. Learning Outcome: Ability to perform exploratory statistical analysis and derive meaningful insights.

9. Create bar charts and pie charts using Matplotlib or Seaborn to visualize categorical data from a real-world dataset. Analyze the visualizations and draw conclusions regarding trends, distributions, and comparisons among different categories.
   a. Expected Input: Dataset containing categorical attributes.
   b. Expected Output: Bar chart and pie chart visualizations.
   c. Learning Outcome: Ability to visualize categorical data and communicate insights effectively.

10. Develop line plots and scatter plots to visualize trends and relationships among variables in a dataset. Analyze the generated plots to identify growth patterns, variations, correlations, and potential business insights.
    a. Expected Input: Dataset containing numerical attributes.
    b. Expected Output: Line plots and scatter plots showing trends and relationships.
    c. Learning Outcome: Ability to analyze trends and relationships using graphical visualization techniques.

11. Perform correlation analysis on a dataset by computing the correlation matrix and visualizing it using a heat map. Identify strongly and weakly related attributes and discuss the significance of correlation analysis in exploratory data analysis and machine learning applications.
    a. Expected Input: Dataset with multiple numerical attributes.
    b. Expected Output: Correlation matrix and heat map visualization.
    c. Learning Outcome: Ability to analyze relationships among variables and identify influential attributes.

12. Carry out a comprehensive Exploratory Data Analysis (EDA) micro-project on a real-world dataset from domains such as education, healthcare, finance, retail, sports, or social media. The analysis shall include dataset understanding, data quality assessment, data cleaning, statistical analysis, visualization, correlation analysis, and presentation of key findings and recommendations through a Jupyter Notebook report.
    a. Expected Input: Real-world dataset selected by the student.
    b. Expected Output: Complete EDA report including visualizations, statistical summaries, observations, and recommendations.
    c. Learning Outcome: Ability to independently perform end-to-end exploratory data analysis on real-world datasets and communicate actionable insights.

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

Textbooks:

1. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, 2nd Edition, O'Reilly Media, 2022.
2. Elbert L. Hearon, Data Analytics with Applications in Business ,Aurora University, 2026. (https://biz.libretexts.org/Courses/Aurora_University/Data_Analytics_with_Applications_in_Business)

Reference books / Manuals:

1. Wayne L. Winston, Microsoft Excel 2019 Data Analysis and Business Modeling, 6th Edition, Microsoft Press, 2019.
2. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and Jupyter, 3rd Edition, O'Reilly Media, 2022.
3. Documentation of NumPy, Pandas, Matplotlib and Seaborn Libraries: https://numpy.org, https://pandas.pydata.org, https://matplotlib.org, https://seaborn.pydata.org

Web links and Video Lectures (e-Resources):

1. Python Data Science Handbook (Jake VanderPlas): https://jakevdp.github.io/PythonDataScienceHandbook/
2. Pandas Documentation: https://pandas.pydata.org/docs/
3. NumPy Documentation: https://numpy.org/doc/
4. Matplotlib Documentation: https://matplotlib.org/stable/users/index.html
5. Seaborn Documentation: https://seaborn.pydata.org/
6. Kaggle Datasets Repository: https://www.kaggle.com/datasets
7. FreeCodeCamp – Data Analysis with Python (Video Course): https://www.youtube.com/watch?v=r-uOLxNrNk8
8. Corey Schafer Python Pandas Tutorial Series: https://www.youtube.com/playlist?list=PL-osiE80TeTsWmV9i9c58mdDCSskIFdDS

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. Conduct hands-on laboratory sessions using Jupyter Notebook with real-world datasets from domains such as education, healthcare, business, sports, and social media to promote experiential learning and data-driven problem solving.
2. Adopt Problem-Based Learning (PBL) by presenting students with practical data analysis scenarios that require data exploration, cleaning, visualization, and interpretation to derive meaningful insights.

3. Encourage collaborative learning through team-based exploratory data analysis activities, peer discussions, and presentation of analytical findings.
4. Integrate industry-relevant datasets from open repositories such as Kaggle, UCI Machine Learning Repository, and Government Open Data Portals to expose students to real-world data challenges.
5. Demonstrate data analysis workflows using Python libraries such as NumPy, Pandas, Matplotlib, and Seaborn, enabling students to learn concepts through practical implementation.
6. Promote inquiry-based learning by encouraging students to formulate analytical questions, investigate datasets, and interpret results using statistical and visualization techniques.
7. Conduct mini-projects and case studies that require students to perform end-to-end exploratory data analysis and communicate findings through reports and presentations.
8. Utilize online documentation, video tutorials, and interactive notebooks to support self-learning and continuous skill development beyond classroom sessions.

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.

Rubrics for CIE – Continuous assessment:

Criteria/Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (1)
Dataset Understanding [CO1, PO1]	Clearly understands dataset structure, attributes, and context; accurately explains data characteristics.	Understands dataset structure and most attributes.	Understands basic dataset characteristics with minor gaps.	Limited understanding of dataset structure and attributes.	Unable to explain dataset characteristics properly.
Data Analysis and Processing [CO2, CO3, PO2, PO5]	Correctly performs statistical analysis and data preprocessing with appropriate methods and interpretation.	Performs analysis and preprocessing with minor errors.	Performs basic analysis but requires guidance for interpretation.	Incomplete analysis and preprocessing.	Unable to perform required analysis properly.
Visualization and Insight Generation [CO4, PO3, PO4, PO5]	Creates meaningful visualizations and derives accurate insights supported by data.	Creates appropriate visualizations and provides reasonable interpretations.	Generates visualizations but provides limited insights.	Visualizations are incomplete or poorly interpreted.	Unable to create meaningful visualizations properly.

Implementation and Tool Usage [CO1, CO4, PO5]	Efficiently implements solutions using Python, NumPy, Pandas, and visualization libraries with minimal errors.	Correct implementation with minor errors.	Basic implementation with moderate assistance.	Significant errors requiring extensive assistance.	Unable to implement the experiment properly.
Documentation and Presentation [CO5, CO6, PO9]	Well-organized notebook/report with clear observations, conclusions, and presentation of results.	Good documentation with proper observations.	Adequate documentation with limited interpretation.	Incomplete documentation and observations.	Poor or missing documentation.
Code Quality, Debugging and Best Practices [CO3, CO6, PO5, PO11]	Writes clean, modular, well-documented code; efficiently debugs errors and follows coding standards and best practices.	Code is mostly organized with minor debugging issues.	Functional code with moderate guidance and limited documentation.	Code contains several logical/syntax errors requiring extensive assistance.	Unable to develop functional code or resolve errors effectively.

Rubrics for SEE / CIE Test:

Criteria/Scale	Superior (5)	Good (4)	Fair (3)	Needs Improvement (2)	Marginal (1)
Problem Understanding and Dataset Analysis (2) [CO1, CO2, PO1, PO2, PO5]	Clearly understands the problem statement and accurately analyzes the dataset requirements.	Good understanding with minor clarification required.	Basic understanding of the problem and dataset.	Limited understanding of the task.	Unable to understand the problem statement properly.
Program Development and Execution (1) [CO3, CO4, PO3, PO5]	Develops and executes a complete solution independently with correct implementation and outputs.	Executes the solution with minor errors.	Completes major portions of the implementation with guidance.	Partial implementation with significant errors.	Unable to develop or execute the program properly.
Result Interpretation and Insight Generation (2) [CO2, CO4, CO5, PO4, PO5]	Correctly interprets outputs, visualizations, and derives meaningful insights from the data.	Interprets results with minor inaccuracies.	Provides basic interpretation of results.	Limited interpretation of outputs.	Unable to interpret results properly.
Use of Data Analysis Tools	Effectively utilizes Pandas, NumPy,	Uses tools correctly with minor errors.	Demonstrates basic usage of tools.	Limited ability to use tools.	Unable to use required tools effectively.

and Visualization (2) [CO4, CO5, PO5	Matplotlib/Seaborn and appropriate EDA techniques.				
Code Quality and Debugging (1) [CO3, CO6, PO5, PO11]	Develops efficient, well-structured, error-free code and independently debugs issues using appropriate programming practices.	Code is mostly correct with minor debugging required.	Functional code with moderate guidance.	Significant coding and debugging errors requiring assistance.	Unable to write or debug the program properly.
Viva Voce and Communication (2) [CO6, PO9]	Demonstrates excellent conceptual understanding and communicates findings confidently.	Good conceptual understanding and communication.	Satisfactory responses to questions.	Limited conceptual understanding.	Unable to explain the experiment or results properly.

- 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**.

Community Project / Societal Project (Project-Based Learning)			
Course Code	**1BCP308**	Scheme	**2025**
Type of Course	AEC/SDC	Semester	3
Teaching Hours/Week (L:T:P)	0:0:0	CIE Marks	50
Total Hours of Pedagogy per Semester CI (L /T): LI(P):TW&SL:TH	0:0:0:30:30	SEE Marks	50
Credits	1	Total Marks	100
Examination type (SEE)	Practical	Exam Hours	03

Course Objective:

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): Choose feasible societal problem statements and solutions considering practical constraints and sustainability aspects.

CO6 (L6 – Create): Create a product by collaborating with stakeholders and within the team to deliver a product/project to the intended community/organization.

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.

List of Project Topics/Titles (Not Limited):

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 at an old-age home.	6	3

3	Virtual Labs for Learning Science	Design and Development simulation environment to do the experimentation list in Physics/Chemistry/Biology for high school in a rural area	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 (Specific nature) for 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, which provides a simple digital map and location information for important facilities on campus.	6	15
10	Public Waste Collection Monitoring System	Design and develop a web/mobile application that enables citizens to report uncollected garbage with geo-tagged 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 a UI in the native language for people in rural areas to make best utilization of the government scheme.	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 energy savings within the community.	6	7, 11, 13
15	Digital Learning Platform for Rural Government Schools	Design and develop a digital learning platform (Tool) that provides interactive learning content and activity-based learning materials in regional languages for rural and small village govt schools (primary govt. schools often lack interactive learning tools)	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, and historical information	6	11, 16
17	Community Disaster Preparedness and Emergency Information System	Design and develop a web/mobile application for disaster preparedness and emergency alerts during natural disasters.	6	11, 13
18	Pesticide Usage Advisory and Organic Farming Awareness System	Design and develop an application that provides recommendations 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	Create a simple web app where users paste suspected text/links to get a risk score based on known 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 or mobile application that provides location-based information on available public transportation services, route maps, estimated travel options, and nearby transport facilities to improve accessibility and promote the use of public transportation.	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 public	Develop a mobile/web application that helps users manage household budgets, savings, government schemes, and financial planning through support.	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 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 nearby police station information.	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 simple water quality observations, and provide awareness on safe water practices through educational resources.	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 by facilitating online patient registration & doctor scheduling/digital health record maintenance/medicine inventory tracking.	6	3
31	Food Waste Reduction and Food Donation Platform	Develop a platform connecting restaurants, hotels, and households with NGOs to donate surplus food before it is wasted.	6	2, 12
32	Community Tree Plantation and Green Cover Monitoring Portal	Develop a web/mobile application to record tree plantation activities.	6	13, 15
33	Career Guidance and Information Portal	Develop an app to get information on aptitude tests, Course/scholarship alerts/placement information	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 platform promoting local artisans and traditional handicraft products.	6	8, 11
37	Waste Recycling Marketplace	Develop a digital platform connecting households, recyclers, and scrap collectors with pickup scheduling.	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 improvement recommendations, and implement a prototype demonstrating compliance improvements.	6	16
39	Digital Library and Knowledge Sharing Platform for Government Schools	Develop a multilingual digital library with 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 the adoption of sustainable energy practices within the community.	6	7, 11, 15
41	Local Business Digital Marketplace	Web portal onboarding of at least 25 local vendors and product catalog	6	8, 12
42	Community Blood Donor and Emergency Volunteer Network	Web/mobile application connecting donors and volunteers	6	3
43	Mental Health Awareness and Helpline Directory App	Develop an application providing verified helpline contacts, self-help resources, and stigma-reducing content.	6	3
44	Diet & Nutrition Awareness Tool for Anganwadi Children	Develop an application to track and suggest balanced meal plans for children under a nutrition scheme.	5, 6	2, 3
45	Digital Literacy Portal for Elderly Citizens	Develop a simplified, accessible web application teaching basic smartphone and internet skills to senior citizens.	6	4, 11
46	Scholarship & Government Scheme Eligibility Checker	Develop a Web tool matching students to eligible scholarships and welfare schemes based on entered criteria.	5, 6	4
47	Digital Record-Keeping System for Self-Help Groups	Develop a Ledger and record-keeping application to help women's SHGs	6	5, 8

		manage savings, loans, and meeting records.		
48	Elderly Companion & Wellness Check-In App	Develop an application enabling family/volunteers to schedule and log wellness check-ins for the elderly living alone.	6	3
49	Old-Age Home Volunteer Coordination Portal	Develop a Platform matching 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.	6	8
51	Street Vendor Digital Registry and Support Portal	Develop an application for the registry helping street vendors to access municipal registration information and gain customer visibility.	6	8
52	Microloan and Repayment Tracker for Small Vendors	Develop an application helping small vendors track micro-loans, repayments, and daily sales.	6	8
53	Rainwater Harvesting Feasibility & Awareness Tool	Develop a tool estimating 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 for citizens to report water contamination issues and track resolution status.	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 aggregating community event and festival schedules for a town or ward.	6	11
57	Community Sports Ground / Facility Booking App	Develop an application for local youth 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 register profiles and gain visibility to coaches and scouts.	6	3
59	Youth Skill Development and Internship Finder App	Develop an application listing local internships, apprenticeships, and skill-training programs for youth.	6	4, 8
60	Financial Literacy Awareness App for Rural Households	Develop an application teaching 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 with simplified explanations of common citizen legal rights.	6	16
62	Digital Insurance / Pension Scheme Awareness Tool	Develop a tool helping unorganized-sector workers check eligibility and enroll in government insurance/pension schemes.	5, 6	8
63	Livestock Health Record and Vaccination Tracker	Develop an application to record cattle and livestock health history and vaccination schedules for farmers.	6	15
64	Soil Testing Appointment & Report Access Portal	Develop a portal letting farmers book soil-testing slots and access digital test reports.	6	15
65	Power Outage Reporting and Tracking App for a Locality	Develop an application allowing residents to report and track power outages to the local utility board.	6	7, 11
66	Community Borewell / Water Tanker Scheduling App	Develop an application to schedule and track water tanker deliveries for water-scarce localities.	6	6, 11
67	Cyber Safety and Digital Fraud Awareness App	Develop an application educating students and senior citizens on common online scams and safe digital practices.	6	16
68	Adult Literacy Program Registration and Progress Tracker	Develop an application to register learners and track attendance and progress in a community adult-literacy drive.	6	4
69	Inclusive Learning Content tool for Differently Abled Students	Develop a tool delivering 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 and flag/follow up on students at risk of dropping out.	6	4
71	Public Park / Garden Maintenance Complaint Tracker	Develop an application for citizens to report maintenance issues in public parks and gardens and track resolution.	6	11,15
72	Digital Suggestion Box for Local Governance	Develop an application allowing citizens to submit suggestions and feedback to local governing bodies on civic matters.	5, 6	16
73	Community Health Screening Camp Scheduler	Develop an application to schedule and manage attendance for community blood-pressure/diabetes screening camps.	6	3
74	Digital Queue Management System for Health Camps	Develop a token-based appointment and queue management system/app	6	3

		for medical camps or PHCs.		
75	Streetlight Fault Reporting System	Develop an application for citizens to report faulty streetlights using photographs and GPS location.	6	11
76	Village Asset Mapping System	Develop a GIS-enabled or map-based application to document community assets such as schools, hospitals, water tanks, and roads.	6	11
77	Food Waste Reduction & Expiry Management System	Design and develop a web/mobile application to record food items, monitor expiry dates, provide 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 hall 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 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 record
4	Problem statement and objective formulation	Review and refine 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 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 outcome	Concept note / design outline
9	Data processing / calculations / sizing / mapping	Support Computations and technical review	Analyze data, perform calculations, prepare maps/charts	Technical analysis sheet
10	Prototype / model / audit / DPR / app development	Monitor progress and correct technical issues	Develop the selected output	Working model/draft DPR / map/app
11	Testing, refinement, and implementation planning	Evaluate technical correctness and improvements	Test prototype or validate proposal with users	Test results / revised output
12	Field implementation / demonstration / awareness activity	Facilitate community interaction and documentation	Conduct implementation or demonstration	Photos, attendance, implementation record
13	Documentation of outcomes, feedback, and impact	Review report structure and completeness	Compile findings, stakeholder feedback, and observations	Draft report
14	Presentation preparation and viva practice	Conduct mock presentation and feedback	Prepare slides/poster and rehearse	Presentation deck
15	Final presentation, submission, and reflection	Evaluate performance using rubrics	Present project, submit report, reflect on learning	Final report, presentation, viva

Assessment structure:

A. Continuous Internal Evaluation (CIE) – 50 marks

- Students, in batches of 3–4, shall undertake at least one project (or a combination of two related activities) aiming, as far as possible, at a logical implemented or implementable outcome.
- CIE will be conducted by a departmental committee (HoD/nominee, guide and one senior faculty) using the following rubrics totalling 50 marks:

CIE pass requirement:

A student shall secure a minimum of 20 marks out of 50 in CIE to become eligible to appear for SEE.

- Students who fail to earn the required CIE marks may be given additional opportunities before closure of the semester; no re-doing of CIE is permitted after the end of the semester.

B. Semester End Examination (SEE) – 100 marks (scaled to 50)

- Examination duration: 2 hours.
- Maximum examination marks: 100 (entered as 100 in the VTU portal; the ITI-SMU module automatically scales 100 → 50 for result processing).
- The SEE shall be conducted batch-wise by a committee of three faculty: project guide, one faculty from the same department and one from a different department.

SEE pass requirement and overall passing rule:

- 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).

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 clearly identified with strong evidence of relevance and well-defined scope. (8)	Relevant problem with clear scope and adequate evidence. (6)	Problem defined but justification is weak. (4)	Problem unclear or not relevant. (2)
Formation of problem statement; engineering approach, innovation & feasibility	10	Precise statement; sound, innovative, feasible approach. (10)	Adequate statement; feasible with minor gaps. (8)	Partial statement; marginal feasibility. (5)	Missing or infeasible. (2)
Interaction with stakeholders; data collection, analysis & interpretation	8	Thorough engagement; comprehensive data and well-interpreted analysis. (8)	Good engagement; adequate data and analysis. (6)	Limited engagement; incomplete data and superficial analysis. (4)	Little or no stakeholder contact; no data/analysis. (2)
Project documentation; clarity, organization & problem-solving demonstrated	10	Complete, well-organized, analytical, and strong problem-solving. (10)	Complete and clear with some analytical content. (8)	Basic with omissions; problem-solving limited. (5)	Missing/poorly organized; no clear problem-solving. (2)

Communication & presentation skills	6	Confident, structured, and answers queries fully; visuals effective. (6)	Clear and satisfactory; responds to most queries. (5)	Acceptable but lacks clarity or confidence; weak responses. (3)	Poor or absent; unable to respond. (2)
Contribution & benefit to society (impact, sustainability)	8	Clear, demonstrable societal benefit and realistic sustainability path. (8)	Benefit plausible; sustainability considered with minor gaps. (6)	Limited or unclear benefit; sustainability weak. (4)	No evident benefit or sustainability. (2)

Notes for CIE rubric:

- Minimum CIE pass requirement: 20/50 to be eligible for SEE. [Use departmental committee (HoD/guide + senior faculty) to apply the rubric and record marks.]

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

Criterion	Weight (marks)	Excellent (Full marks)	Good	Fair	Needs Improvement
Problem identification & relevance to society/environment	16	Clearly articulated with strong evidence and context. (16)	Clear with reasonable evidence. (12)	Problem stated but evidence/context weak. (8)	Poorly defined or irrelevant. (4)
Formation of problem statement; engineering approach, innovation & feasibility	20	Precise; demonstrates engineering depth, novelty and feasibility. (20)	Technically sound and feasible with minor limitations. (15)	Basic; limited technical depth or feasibility concerns. (10)	No viable engineering approach or infeasible plan. (5)
Interaction with stakeholders; data collection, analysis & interpretation	16	Excellent engagement; rigorous data collection and insightful analysis. (16)	Good engagement and acceptable analysis. (12)	Superficial data collection and limited analysis. (8)	Inadequate or no data; poor analysis. (4)
Project documentation; clarity, technical rigor & problem-solving demonstrated	20	Comprehensive, technically rigorous, and systematic. (20)	Complete and technically correct with minor gaps. (15)	Lacks technical depth or has notable omissions. (10)	Missing key elements or technically incorrect. (5)
Communication, presentation & viva-voce responses	12	Presents expertly, defends work persuasively and answers all queries correctly. (12)	Good presentation and satisfactory answers to most queries. (9)	Adequate but fails to answer some technical questions. (6)	Poor presentation and unable to defend work. (3)

Contribution & benefit to society; impact assessment & sustainability	16	Clear, evidence-based impact assessment with scaling recommendations. (16)	Reasonable impact discussion and sustainability considerations. (12)	Limited evaluation of impact or sustainability. (8)	No coherent assessment of impact or sustainability. (4)

SEE pass rule:

- Student must secure at least 35/100 in SEE itself to be eligible for pass consideration.
- Final passing is based on CIE (out of 50) + SEE (scaled to 50) >= 40. Otherwise, grade F (Fail) is awarded.

Suggested implementation notes (for BoS / exam committees):

- Use numerical scores from the rubric and keep brief evaluator comments for each criterion.
- For team projects: record individual contribution within the documentation and adjust individual marks using a peer-evaluation sheet or guide's assessment.
- Maintain evidence (field notes, photos, questionnaires, DPRs, reports) as annexures for SEE moderation.
- Use standardized score sheet (spreadsheet) mapping rubric items to columns for easy aggregation and archival.

- Awareness Campaign on Single-Use Plastic and E-Waste Management

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 on Project-Based Learning and Social Innovation.
3. SWAYAM courses on community engagement and sustainability.
4. UN Sustainable Development Goals portal.
5. MyGov India and local civic body portals for community initiatives.
6. YouTube resources on design thinking, social innovation, and community service learning.
7. NGO and government field-action case studies related to sanitation, water, health, education, and environment.

Assessment Structure:

Continuous Internal Evaluation (CIE) shall be conducted by the Departmental Project Review Committee consisting of a Senior Professor, the Project Guide, and one additional faculty member nominated by the Head of the Department. Semester End Evaluation (SEE) shall be conducted by university-appointed examiners. The evaluation shall be based on rubrics designed to assess the attainment of NBA-defined graduate attributes, including problem analysis, investigation, design of solutions, communication, teamwork, and professionalism**.**

The weightage of marks for CIE and SEE is 50% each.

Pass Criteria: A candidate shall secure a minimum of 40% of the total marks prescribed for Project Work, subject to the required minimum in Continuous Internal Evaluation and Semester End Evaluation as per VTU regulations.