

www.sjcit.ac.in

||Jai Sri Gurudev||

Sri Adichunchanagiri Shikshana Trust (R.)

SJC INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU from 2024-25

AICTE Approved, Accredited by NAAC with A+ Grade & NBA (CSE, ISE, ECE, ME & CV), NIRF(151-300) in 2023,Gold Rated by QS I-Gauge

P.B. No.20, B.B Road, Chikkaballapur - 562 101, Karnataka

Estd : 1986

III Semester
Scheme & Syllabus of B.E. Programme
Computer Science & Engineering (AI&ML)
(2025 Scheme)

Core Values

Spiritual, Judicious, Commitment, Integrity, Trustworthy



AICTE | NATIONAL INSTITUTIONAL RANKING FRAMEWORK 151-300 | NBA | NAAC A+ | QS I-GAUGE GOLD | INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative) 4-Star Rating

||jai sri gurudev||
Sri Adichunchanagiri Shikshana Trust (R.)

SJC INSTITUTE OF TECHNOLOGY

(An Autonomus Institute Under VTU Belagavi & Approved by AICTE, New Delhi)
Accredited by NAAC with A+ Grade & NBA (CSE, ISE, ECE, ME & CV), NIRF (151-300), QS-I Gauge GOLD Rated
Estd.:1986
Post Box #20, B.B. Road, Chikkaballapur - 562 101, Karnataka

B.E. in Dept of Computer Science and Engineering (Artificial Intelligence and Machine Learning)

Scheme of Teaching and Examinations

2025

Outcome-Based Education (OBE) and Choice-Based Credit System (CBCS) (Effective from the academic year 2025-26)

III SEMESTER

Sl. No.	Course	Course Code	Course Title	Teaching Department (TD) and Question Paper Setting Board (PSB)	Teaching and Learning Scheme					Examination				Credits
					CI		LI	TW & SL	Total Hours per sem	uration in hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	SL						
1	ASC	1BMATS301	Discrete Mathematical Structures and Graph theory	TD /PSB: Mathematics	42	28	0	50	120	3	50	50	100	4
2	IPCC	1BCS302	Object Oriented Programming with Java	TD/PSB : CS Allied Depts	42	0	28	50	120	3	50	50	100	4
3	IPCC	1BCS303	Digital Design and Computer Organization	TD/PSB : CS Allied Depts	42	0	28	50	120	3	50	50	100	4
4	PCC	1BCS304	Operating Systems	TD/PSB : CS Allied Depts	42	0	0	48	90	3	50	50	100	3
5	PCC	1BCS305	Data Structures and Algorithms	TD/PSB : CS Allied Depts	42	0	0	48	90	3	50	50	100	3
6	PCCL	1BCSL306	Data Structures Laboratory	TD/PSB : CS Allied Depts	0	0	28	02	30	3	50	50	100	1
7	AEC	1BXXL307x	Ability Enhancement Course Laboratory	TD/PSB : CS Allied Depts	0	0	28	02	30	2	50	50	100	1
8	SDC	1BCP308	Community Project	Any Department/ Respective Engineering Dept.	0	0	0	30	30	2	50	50	100	1

9	NCMC	1BNS309	National Service Scheme	Campus	NSS coordinator	0	0	0	28	28	--	100	---	100	PP
		1BPE309	Physical Education		Physical Education Director										
		1BYO309	Yoga		Yoga Teacher										
		1BMU309	Music		Music Teacher										
		1BNC309	National Cadet Corps - I		NCC coordinator										
10	NCMC	1BCE310	Competency Enhancement Course-1	CDCAC		14	0	0	14	28	--	100	---	100	PP
11	NCMC	1BMATDIP311	Additional Mathematics course – I	TD/PSB -Maths Dept.		14	0	0	14	28	--	100	---	100	PP
Total												**700**	**400**	**1100**	**21**

Ability Enhancement Course (Laboratory) 1BxxL307x			
1BCSL307A	Project Management with Git	1BCIL307C	Intelligent Design Controller
1BCSL307B	Web Application Development with HTML and PHP	1BCSL307D	Data Visualization with Python

Semester: III					
DISCRETE MATHEMATICAL STRUCTURE AND GRAPH THEORY **(Common to the Branches: AD, AI, CD, CS & IS)**					
Course Code	**:**	1BMATS301	**Credits**	**:**	4
Hours/Semester (L:T:P:SL)	**:**	30:20:0:0	**CIE**	**:**	50 Marks
Total Hours	**:**	50	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	3

Prerequisites:	
Courses	Calculus and Linear Algebra, Vector calculus and Numerical Methods
Knowledge on Fundamental Concepts	Basic Set Theory, Logic, Proof Techniques, Matrix Operations.
Knowledge on Connected Tools (If any)	Basic computer knowledge

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamentals of propositional and predicate logic for logical reasoning and proof-based argument verification.
CLO2	Gain the knowledge of sets, relations, and functions in computer science applications.
CLO3	Learn the concepts of functions, recurrence relations, and coding theory in computation and data communication.
CLO4	Familiarize with graph properties such as connectivity and planarity for solving computer science problems.
CLO5	Apply the knowledge of trees, spanning trees, and shortest path concepts in network design and optimization.

Module – 1	**10Hrs**
Mathematical Logic and Proof Techniques Propositional Logic: Propositions, logical connectives, truth tables, tautology, contradiction, logical equivalences — De Morgan's laws, absorption, distributive laws. Normal forms Conjunctive Normal Form (CNF) and Disjunctive Normal Form (DNF), Rules of inference. Quantifiers: quantifiers (universal and existential), logical equivalences with quantifiers. Methods of proof and disproof: Direct proof, indirect proof, proof by contradiction.	
Module – 2	**10Hrs**
Set Theory and Relations: Set Theory: Sets, subsets, power sets, Cartesian products, set operations — union, intersection, complement, difference, symmetric difference. Venn diagrams. Inclusion exclusion principle. Permutations and combinations. Relations: Binary relations, properties of relations — reflexive, symmetric, antisymmetric, transitive. Equivalence relations and partitions. Partial orders, Hasse diagrams, lattices.	
Module – 3	**10Hrs**

Functions: Types of functions — injective, surjective, bijective. Composition of functions, inverse functions. Recursive functions. Pigeonhole principle and its generalizations.

Recursive definition, Recurrence Relation

Recursive definition, First order linear recurrence relation-Formulation, problems and examples, Second order linear recurrence relations with constant coefficients-Homogeneous and non-homogeneous, Generating functions.

Coding theory:

Elementary coding theory, the hamming metric, the parity-Check and Generator Matrices

Module – 4	**10Hrs**

Graph Theory – I:

Introduction to Graphs: Definitions — simple graph, multigraph, pseudo graph, directed graph. Degree sequence, handshaking lemma: theorem and its properties. Special graphs complete, bipartite, regular. Subgraphs, graph isomorphism.

Connectivity and Paths: Walks, trails, paths, cycles. Connected and disconnected graphs. Euler paths and circuits — Euler's theorem. Hamiltonian paths and cycles sufficient conditions (Dirac's and Ore's theorems).

Directed Graphs: Digraphs, tournaments, strongly connected components. Adjacency matrix and incidence matrix representation.

Module – 5	**10Hrs**

Graph Theory – II:

Trees: Definition, properties of trees, rooted trees, m-ary trees. Tree traversal algorithms preorder, inorder, postorder. Spanning trees — BFS and DFS algorithms. Weighted graphs Prim's and Kruskal's algorithms for minimum spanning trees.

Shortest path algorithm — Dijkstra's algorithm. Problems.

Topics for Self learning		
Module-1	**Applications:**	Boolean search engines, if-else conditions in coding, SQL WHERE clauses, everyday decision making
	Self-Study:	Logical puzzles, argument validation
Module-2	**Applications:**	Database table joins (union/intersection), Social media friend lists (relations)
	Self-Study:	Relation Matrices and Graph Representation of Relations, Wars hall's Algorithm for Transitive Closure
Module-3	**Applications:**	Password validation (bijective functions), Fibonacci in nature (recurrence), Error detection in QR codes & barcodes (Hamming/parity check)
	Self-Study:	Hamming distance concept.
Module-4	**Applications:**	Google Maps shortest path (connectivity), Social networks friend Connections (graphs).
	Self-Study:	Graph Coloring, Network Flow and Connectivity Analysis
Module-5	**Applications:**	Google Maps navigation (Dijkstra's), Network cable layout with Minimum cost (Prim's/Kruskal's).
	Self-Study:	Adjacency Matrix, Incidence Matrix.

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply propositional and predicate logic to formulate and verify logical arguments using appropriate proof techniques
CO2	Solve problems in computer science using set theory, relations and functions.
CO3	Apply functions, recurrence relations and coding theory concepts to solve problems in computer science and data communication.
CO4	Analyze properties of graphs including connectivity, planarity to solve problems in computer science.
CO5	Apply tree structures, spanning tree algorithms, shortest path algorithms to design efficient solutions for network and optimization problems.

Text Books	
1	Discrete Mathematical Structures – G. Shankar Rao, 3rd Edition, New Age International Publishers, 2021. ISBN: 978-93-86418-20-3.Chapter 1: 1.1–1.26 Chapter 2: 2.1–2.27 Chapter 3: 3.1-3.17 Chapter 4: 4.1-4.15 Chapter 5: 5.1–5.20 Chapter 7: 7.1-7.12 Chapter 16: 16.1-16.3
2	Discrete Mathematics with Graph Theory – Edgar G. Goodaire and Michael M. Parmenter, Pearson, 2023. ISBN: 978-93-5606-514-7 Chapter 1: 1.1–1.4 Chapter 3: 3.1–3.3 Chapter 5: 5.1–5.4 Chapter 4: 4.1–4.5 Chapter 7: 7.1–7.3
3	Discrete and Combinatorial Mathematics – Ralph P. Grimaldi, 5th Edition, Pearson Education, 2018. ISBN: 978-0-321-38502-4. Chapter 16

Reference Books	
1	Elements of Discrete Mathematics – C. L. Liu and D. P. Mohapatra, 4th Edition, McGraw-Hill Education, 2017. ISBN: 978-93-5260-515-6
2	Introduction to Graph Theory – Douglas B. West, 2nd Edition, Pearson Education, 2017 (Reprint). ISBN: 978-81-317-4489-9.
3	Discrete Mathematics for Computer Scientists and Mathematicians – Joe L. Mott, Abraham Kandel, and Theodore P. Baker, 2nd Edition, Pearson Education, 2015. ISBN: 978-93-3255-049-0.

Web Links and Video Lectures (e-Resources)	
1	NPTEL – Discrete Mathematical Structures (IIT Madras**),** YouTube Playlist**:** https://www.youtube.com/playlist?list=PL0862D1A947252D20
2	Coursera **–** Discrete Mathematics **(**University of California San Diego**),** Online Course**:** https://www.coursera.org/specializations/discrete-mathematics
3	MIT Open Courseware – Mathematics for Computer Science (6.1200J), Spring 2024: https://ocw.mit.edu/courses/6-1200j-mathematics-for-computer-science-spring-2024/

CO – PO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO1	3	3	2	2	-	-		-	1	1	1
CO2	3	3	3	2	-	-	-	-	1	1	1
CO3	3	3	3	2	1	-	-	-	1	2	2
CO4	3	3	2	3	1	-	-	-	1	1	1
CO5	3	3	3	3	2	-	-	-	1	2	2

	CSE		**ISE**		**CSD**		**AI&ML**		**AI&DS**	
	PSO1	**PSO2**	**PSO1**	**PSO2**	**PSO1**	**PSO2**	**PSO1**	**PSO2**	**PSO1**	**PSO2**
CO1	1	1	1	1	1		1	1	1	
CO2		1				1		1		1
CO3		1	1	1		1		1		1
CO4	1	1	1	1	1	1			1	
CO5		1		1	1	1	1	1	1	1

Semester: III					
Object Oriented Programming with Java					
Course Code	**:**	1BCS302	**Credits**	**:**	4
Hours/Semester (L:T:P:SL)	**:**	42:0:28:50	**CIE**	**:**	50 Marks
Total Hours	**:**	120	**SEE**	**:**	50 Marks
Course Type	**:**	IPCC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Problem Solving through Programming, Introduction to Python Programming.
Knowledge on Fundamental Concepts	Basic knowledge of programming concepts including variables, datatypes, operators, conditional statements, loops and arrays.
Knowledge on Connected Tools (If any)	Text Editor/IDE

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the basic primitive constructs and fundamentals of classes and objects in the Java programming language.
CLO2	Apply object-oriented programming features of Java to develop solutions for real-world problems.
CLO3	Gain the knowledge of packages, multithreaded programming and exception handling in Java.
CLO4	Examine thread behavior and synchronization mechanisms for developing concurrent Java applications.
CLO5	Assess string processing and wrapper class utilities to optimize Java program design and functionality.

Module – 1	**08 Hrs**
Introduction to Java: Object-Oriented Programming (Two Paradigms, Abstraction, and The Three OOP Principles), The Bytecode, The Java Buzzwords, Structure of Java Program, Data Types, Variables, Arrays, Operators and Control Statements. **Introducing Classes:** Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, this keyword, Garbage Collection. **Methods and Classes:** Overloading Methods, Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes.	
Module – 2	**08 Hrs**
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. **Interfaces:** Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods.	

Module – 3	**08 Hrs**
Packages: Packages, Packages and Member Access, Importing Packages. **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.	
Module – 4	**08 Hrs**
Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming and Stopping Threads, Obtaining a Thread's State. **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).	
Module – 5	**08 Hrs**
String Handling: The String Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion Using valueof(), Changing the Case of Characters Within a String, joining strings, Additional String Methods, String Buffer, StringBuilder.	

Practical component	
Module – 1	**1.** Create a class namely Account with the data members (Accno : integer, Name : String, Phone No: integer, Balance_amt: float) and following methods: a. CreateAccount () method to create an account. b. Deposit () method to deposit amount to an account. c. Withdraw () method which gets the amount to be withdrawn from an account. d. PrintAccount () method to display account details.
	2. Develop a program to demonstrate method overloading to create a class called Calculator with three methods named Compute with different type of parameters and print the computed results.
	3. Implement a program to illustrate constructor overloading to calculate the area of a rectangle and a circle using appropriate constructors.
Module – 2	**4.** Develop a Program that does the following: a. Create an abstract class called Vehicle which contains the year of manufacture data member and two abstract methods getdata() and putdata() with a constructor. b. Create two derived classes Two-wheeler and Four-wheeler and implement the abstract methods. Make Four-wheeler as final class. c. Create class MyTwoWheeler which is a subclass of Two-wheeler and demonstrate the use of super keyword to initialize data members of MyTwoWheeler.
	5. Implement a 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 resize methods.
	6. Create a program that models electronic devices (e.g., smartphones, laptops, and tablets) using a common interface for power management. The program

	should allow users to interact with the devices and control their power state.
Module – 3	**7.** Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.
	8. Develop a program to read two integers a and b. Compute a/b and print, when b is not zero. Raise an exception when b is equal to zero also demonstrate working of ArrayIndexOutOfBounds Exception.
Module – 4	**9.** Implement a program to demonstrate multi-threaded application that has two threads. one thread displaying college name once every ten seconds and another displaying department name once every two seconds.
Module – 5	**10.** Design a program to demonstrate string handling including creating strings using constructors and literals, concatenating strings, extracting characters at a specified index and comparing strings for equality.

Topics for Self learning	
Module – 1	JavaDoc Documentation and Coding Standards, Command-Line Arguments in Java, Object Cloning and Copy Constructors in Java, Constructor Chaining using this() and super().
Module – 2	Functional Interfaces, Multiple Interface Implementation, Real-Time Application of Dynamic Method Dispatch.
Module – 3	User-Defined Exceptions in Banking Applications, Package Creation and Access Control, Creating and Using Executable JAR Files.
Module – 4	Thread Synchronization using Bank Account Transactions, Thread Life Cycle and State Transitions, Producer-Consumer Problem.
Module – 5	Regular Expressions for String Validation, String-Based Password Strength Analyzer , StringBuffer vs StringBuilder Performance Comparison.

Course Outcomes: After completing the course, the students will be able to	
CO1	Interpret the fundamental concepts of Java programming, including data types, control structures, classes, objects, methods, constructors, and arrays.
CO2	Apply object-oriented programming principles such as inheritance, interfaces, abstraction, and polymorphism to design Java-based solutions for real-world problems.
CO3	Demonstrate the use of packages, exception handling mechanisms, and access control features in developing robust Java applications.
CO4	Analyze and implement multithreaded Java applications using thread synchronization and inter-thread communication mechanisms.
CO5	Develop Java applications utilizing string handling techniques and object-oriented programming concepts to solve practical computing problems.

Textbook	
1	"Java: The Complete Reference", Herbert Schildt, 13th Edition, 2024, McGraw-Hill, ISBN: 9781265058432. Ch. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 18

Reference Books	
1	"Introduction to Java Programming", Y. Daniel Liang, 2017, Pearson Education, ISBN-13 978-0134671710
2	"Thinking in Java", Bruce Eckel, Fourth Edition, 2006, Prentice Hall, ISBN-13: 978-0131872486

Web links and Video Lectures (e – Resources)	
1	Java Tutorial: https://www.geeksforgeeks.org/java/
2	Introduction To Programming In Java (by Evan Jones, Adam Marcus and Eugene Wu): https://ocw.mit.edu/courses/6-092-introduction-to-programming-in-java-january-iap-2010/
3	Java Tutorial: https://www.w3schools.com/java/

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	1	1		1						1	1	1
CO2	3	2	3		2							1	1
CO3	2	3	2	1	3							1	1
CO4	2	3	2	2	3						1	1	1
CO5	2	2	3	1	3				2	2	1	1	1

Semester: III					
Digital Design and Computer Organization					
Course Code	**:**	1BCS303	**Credits**	**:**	4
Hours/Semester (L:T:P:SL)	**:**	42:0:28:50	**CIE**	**:**	50 Marks
Total Hours	**:**	120	**SEE**	**:**	50 Marks
Course Type	**:**	IPCC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Introduction to Electronics & Communication.
Knowledge on Fundamental Concepts	Understanding of number systems, Boolean algebra, Logic gates, combinational and sequential circuits and data representation.
Knowledge on Connected Tools (If any)	

Course Learning Objectives: This course will enable the students to	
CLO1	Familiarize the functionalities of digital logic system.
CLO2	Understand the functionalities of combinational and sequential logic circuits.
CLO3	Interpret the basic architecture of computer system.
CLO4	Study the working of I/O operations and processing unit.
CLO5	Apply arithmetic concepts to solve computational problems.

Module – 1	**08 Hrs**
Introduction to Digital Design: Binary Logic, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Digital Logic Gates, Introduction, The Map Method, Four Variable- Map, Don't-Care Conditions, NAND and NOR Implementation, Other Hardware Description Language – Verilog Model of a simple circuit.	
Module – 2	**08 Hrs**
Combinational Logic: Introduction, Combinational Circuits, Design Procedure, Binary Adder- Subtractor, Decoders, Encoders, Multiplexers. HDL Models of Combinational Circuits – Adder, Multiplexer, Encoder. **Sequential Logic:** Introduction, Sequential Circuits, Storage Elements: Latches, Flip-Flops.	
Module – 3	**08 Hrs**
Basic Structure of Computers: Functional Units, Basic Operational Concepts, Bus structure, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. **Machine Instructions and Programs**: Memory Location and Addresses, Memory Operations, Instruction and Instruction sequencing, Addressing Modes.	

Module – 4	**08 Hrs**
Input/output Organization: Accessing I/O Devices, Interrupts– Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Direct Memory Access: Bus Arbitration, Speed, size and Cost of memory systems. Cache Memories–Mapping Functions.	
Module – 5	**08 Hrs**
Arithmetic: Addition and subtraction of signed numbers, Design of fast adders, Multiplication of positive numbers, signed–operand multiplication, fast multiplication: Bit-Pair Recoding of Multipliers, Carry-Save Addition of Summands.	

Practical component	
Module – 1	**1.** Given a 4-variable logic expression, simplify it using Map Entered Variable (MEV) technique and implement the same using basic gates.
	2. Implement Binary- Half and Full Adder, Half and Full Subtractor using basic Gates.
Module – 2	**3.** Design Verilog HDL to implement (a+b)*(c+d) using Structural, Dataflow and Behavioral model.
	4. Design Verilog HDL to implement Binary Adder, Subtractor- Half and Full Adder, Half and Full Subtractor.
	5. Design Verilog HDL to implement Decimal Adder.
	6. Design Verilog program to implement different types of Multiplexer like 2:1, 4:1 and 8:1.
	7. Design Verilog program to implement types of De-Multiplexer.
	8. Design Verilog program for implementing various types of Flip-Flops such as SR, JK and D.

Topics for Self learning	
Module – 1	Steps in Quine Mc-clusky technique
	Problems on Quine Mc-clusky Method.
Module – 2	Design MOD-3 Ripple counter.
	Design MOD-5 Synchronous Up Counter.
Module – 3	Explain timing and data transfer in Synchronous Bus.
	Explain timing and data transfer in Asynchronous Bus.
Module – 4	Basic Cache optimization technique.
	Advanced Cache Optimization Technique.
Module – 5	Restoring Integer Division.
	Non-Restoring Integer Division.

Course Outcomes: After completing the course, the students will be able to	
CO1	Interpret the principles of Boolean algebra and digital logic circuit design.
CO2	Design combinational and sequential logic circuits using digital components.
CO3	Analyze the structure, functionality, and programming aspects of computer systems.

CO4	Examine I/O communication, interrupts, DMA, and memory performance in computer systems.
CO5	Assess arithmetic processing techniques used in computer systems.

Textbooks	
1	"Digital Design with an Introduction to Verilog Design", M.Morris Mano, Michael D. Ciletti, 5th Edition, 2013, Pearson Education, ISBN-10:8131794741. Chapters: 1,2,3,4,5
2	"Computer Organization", Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 5th Edition, 2019, Tata McGraw Hill, ISBN-13: 9781259005275. Chapters: 1,2,4,5,6

Reference Books	
1	"Digital Design and Computer Architecture", David Money Harris and Sarah L. Harris, 1st Edition, 2007, Elsevier, ISBN 9780080922812.
2	"Computer Organization and Architecture", William Stallings, 11th Edition, 2022, Pearson Education, ISBN-10: 9356061599.

Web links and Video Lectures (e – Resources)	
1	https://www.youtube.com/watch?v=g3oSkhK9ghg&list=PLDVWzw0WSIX6sIZo1WiVFooEFOV3d0LNx
2	https://www.youtube.com/watch?v=CDO28Esqmcg&list=PLhwVAYxlh5dvB1MkZrcRZy6x_a2yORNAu

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2	1		2			1	1		1	2	1
CO2	3	3	3	1	2	1		1	1		1	2	2
CO3	3	3	3	1	2			1	1		1	3	2
CO4	2	3	1	2	2			1	1		1	2	3
CO5	2	3	2	1	1			1	1		1	2	2

Semester: III

Operating Systems					
Course Code	:	1BCS304	**Credits**	:	3
Hours/Semester (L:T:P:SL)	:	42:0:0:48	**CIE**	:	50 Marks
Total Hours	:	90Hrs	**SEE**	:	50 Marks
Course Type	:	PCC	**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Problem Solving through Programming.
Knowledge on Fundamental Concepts	Variables, loops, functions, and arrays.
Knowledge on Connected Tools (If any)	Text Editors.

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the importance of an operating system and identify its different types.
CLO2	Explore the mechanisms used by operating systems to optimize the utilization of CPU, memory, and storage resources.
CLO3	Apply system-level APIs for managing processor, memory, and storage resources.
CLO4	Analyze file system and storage management operations using system-level commands.
CLO5	Recognize the importance of protecting information and how operating systems enforce security mechanisms.

Module – 1	**08 Hrs.**
Introduction to operating systems, System structures: What operating systems do, Computer System organization, Computer System architecture, Operating System structure, Operating System operations, Distributed system, Special-purpose systems, and Computing environments. **Operating System Services:** User - Operating System interface, System calls, Types of system calls, System programs, Operating system design and implementation, Operating System structure, Virtual machines, Operating System debugging, Operating System generation, System boot.	
Module – 2	**08 Hrs.**
Process Management: Process concept, Process scheduling, Operations on processes, Inter process communication. **Multi-threaded Programming:** Overview, Multithreading models, Thread Libraries, Threading issues. **Process Scheduling**: Basic concepts, Scheduling Criteria, Scheduling Algorithms, Thread scheduling, Multiple-processor scheduling.	
Module – 3	**08 Hrs.**
Process Synchronization: Synchronization: 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.
Module – 4 **08 Hrs.**
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.
Module – 5 **08 Hrs.**
File Systems: File concepts, Directory and Disk Structures. **Implementing File system:** File system structure, File system implementation, Directory implementation, Allocation methods, Free space management. **Secondary Storage Structure, Protection:** Overview of Mass storage structures, Disk structure, Disk scheduling. **Protection**: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of Access Matrix.

Topics for Self learning/Applications:	
Module – 1	**Self-Study:** RTOS, Mobile Operating Systems, Virtualization, Container-based OS.
	Application: Autonomous Vehicle Computing Platforms, Smart Home Automation Systems, Wearable Health Monitoring Devices.
Module – 2	**Self-Study:** Cloud Process Scheduling, Multithreading Applications, Linux Process Management, Asynchronous Programming.
	Application: Airport Flight Control Systems, Cloud Gaming Platforms, Automated Stock Trading Systems.
Module – 3	**Self-Study:** Database Synchronization, Distributed Locking, Race Conditions, Deadlock in Cloud Systems.
	Application: Online Banking Transaction Processing, Hospital Patient Record Management, Warehouse Robot Coordination.
Module – 4	**Self-Study:** Garbage Collection, Hypervisors, Memory Leak Detection, Cache Optimization.
	Application: Artificial Intelligence Model Execution, Augmented Reality (AR) Applications, Video Streaming Platforms.
Module – 5	**Self-Study:** Cloud Storage, Distributed File Systems, RAID Technology, File Encryption.
	Application: Blockchain Data Storage, Digital Evidence Management Systems, Smart City Surveillance Storage.

Course Outcomes: After completing the course, the students will be able to	
CO1	Understand the structure and functionality of operating system.
CO2	Apply appropriate CPU scheduling algorithms for the given problem.
CO3	Analyze the various techniques for process Synchronization and Deadlock handling.
CO4	Develop solutions for memory management problems using suitable operating system techniques.
CO5	Implement Storage Management and data protection strategies.

Textbook

1	"Operating System Concepts", Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 8th Edition, 2012, Wiley, ISBN: 978-81-265-2051-0. Text Book 1: Ch 1,2,3,4,5,6,7,8,9,10,11,12,14.

Reference Books	
1	"Operating Systems: A Modern Perspective", Gary J. Nutt, 20 Aug 2009, Addison Wesley, the University of Virginia, ISBN 0201741962.
2	"Modern operating systems", Tanenbaum, Andrew, 4th Edition, 2009, Pearson Education, Inc. ISBN 013359162X, 978-0133591620.
3	"Understanding Operating System", Ann McHoes Ida M Fylnn, 6th Edition, 2013, Cengage Learning, ISBN: 9788131521564.
4	"Operating Systems: A Concept Based Approach", D.M Dhamdhere, 3rd Edition, 2013, McGraw- Hill, 2013, ISBN: 978-1259005589.
5	"An Introduction to Operating Systems: Concepts and Practice", P.C.P. Bhatt, 4th Edition, 2014, PHI(EEE), ISBN: 9789387472877.
6	"William Stallings Operating Systems: Internals and Design Principles", 6th Edition, 2011, Pearson, ISBN: 978-0273751502.

Web links and Video Lectures (e – Resources)	
1	https://nptel.ac.in/courses/106106144
2	https://youtu.be/mXw9ruZaxzQ
3	https://archive.nptel.ac.in/courses/106/105/106105214/
4	https://www.youtube.com/watch?v=3-ITLMMeeXY&list=PL3pGy4HtqwD0n7bQfHjPnsWzkeRn6mkO
5	https://github.com/nthnluu/weenix - writing parts of a Unix kernel.
6	https://github.com/cfenollosa/os-tutorial - How to create an OS from scratch

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2	2	1		1					1	2	2
CO2	3	2	3	2			1				1	2	2
CO3	2	2	3	2	2		1			1	1	2	2
CO4	2	2	3	2	2		1			1	1	2	2
CO5	2	2	2	2	2	1	1	1	1	1	1	2	2

Semester: III					
Data Structures and Algorithms					
Course Code	**:**	1BCS305	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Problem Solving through Programming.
Knowledge on Fundamental Concepts	Control Structures, Functions, Arrays, Pointers, Structures, Algorithms and Computational Complexity.
Knowledge on Connected Tools (If any)	Edit plus, GCC, Code: Blocks, Visual Studio.

Course Learning Objectives: This course will enable the students to	
CLO1	Gain the knowledge of the fundamentals of data structures and their applications.
CLO2	Describe the representation of different data structures.
CLO3	Develop efficient solutions for real-world computational problems using linear data structures.
CLO4	Examine the applications of trees and graphs in designing efficient problem-solving solutions.
CLO5	Evaluate and select appropriate hashing techniques for efficient key–value pair mapping and data retrieval in real-world applications.

Module – 1	**08 Hrs**
Introduction to Data Structure and Algorithms: Classification of Data Structures, Operations on Data Structures, Algorithms, Different Approaches to Designing an Algorithm, Time and Space Complexity, Big O Notation, Omega Notation, Theta Notation.	
Module – 2	**08 Hrs**
Linear Data Structures: Introduction to Stacks, Array representations of stacks, Operations on a Stack, Applications of Stacks, Introduction to Queues, Array Representation of Queues, Types of Queues, Applications of Queues.	
Module – 3	**08 Hrs**
Linked Lists: Introduction, Singly Linked Lists, Circular Linked Lists, Doubly Linked Lists, Circular Doubly Linked Lists, Header Linked Lists, and Applications of Linked lists – Polynomial Representation.	
Module – 4	**08 Hrs**
Trees: Introduction- Basic Terminology, Types of Trees, Creating a Binary Tree from a general Tree, Traversing a Binary Tree: Pre-order Traversal, In-order Traversal, Post-order Traversal, Constructing a Binary Tree from, Traversal Results, Binary Search Trees, Operations on Binary Search Trees: Search, insert and delete.	
Module – 5	**08 Hrs**
Graphs: Representation of Graphs, Graph Traversal Algorithms, Topological Sorting, Warshall's Algorithm, Modified Warshall's Algorithm. **Hashing and Collision:** Introduction, Hash Tables, Different Hash Functions and Collisions.	

Topics for Self learning	
Module –1	Abstract Data Type & Memory Allocation Techniques.
	Algorithm, Control Structures Used In Algorithms.
	Practice finding time and space complexity of simple programs using step-count method.
Module – 2	Multiple Stacks.
	Study FIFO (First In First Out) concept and compare queues with stacks.
Module – 3	Study the need for linked lists over arrays and compare their advantages and limitations.
	Multi-linked Lists.
	Understand the concept of header nodes and their benefits in linked list operations.
Module – 4	Huffman's Tree , Applications of Trees.
	Practice implementing Pre-order, In-order, and Post-order traversals using recursive and non-recursive approaches.
	Determining the Height of a Binary Search Tree.
	Determining the Number of Nodes.
	Finding the Smallest Node & Largest Node in a Binary Search Tree.
Module – 5	Graph Terminology & Directed Graphs.
	Applications of Graphs.

Course Outcomes: After completing the course, the students will be able to	
CO1	Comprehend fundamental concepts of Data Structures, Algorithm paradigms and their Efficiency.
CO2	Illustrate and apply linear data structures to solve real-world computational problems.
CO3	Analyze tree structures, traversals, and Binary Search Tree operations for efficient data organization and searching.
CO4	Apply graph representations, traversal algorithms, topological sorting, and Warshall's variants to solve connectivity, scheduling, and shortest-path problems.
CO5	Analyze the occurrence of collisions in hash tables and employ appropriate collision resolution techniques to optimize search time.

Textbook	
1	"Data Structures using C", Reema Thareja, 2nd Edition,2014, Oxford University Press, ISBN: ISBN: 9780198099307 **Chapters:** 2,6, 7, 8, 9, 10, 13

Reference Books	
1	"Fundamentals of Data Structures in C", Ellis Horowitz, S. Sahni, Freed, 2nd Edition, 2015, ISBN: 9788173716058.

2	"Simplified approach to Data Structures", Vishal Goyal, Lalit Goyal and Pawan Kumar, Shroff publications and Distributors, ISBN: 978-93-5110-531-2.

Web links and Video Lectures (e – Resources)	
1	https://www.youtube.com/watch?v=zWg7U0OEAoE&list=PLBF3763AF2E1C572F
2	https://www.youtube.com/watch?v=PGWZUgzDMYI&list=PLBF3763AF2E1C572F&index=3
3	https://nptel.ac.in/courses/106104697
4	https://www.youtube.com/watch?v=KW0UvOW0XIo&list=PLBF3763AF2E1C572F&index=5
5	https://www.youtube.com/watch?v=tORLeHHtazM&list=PLBF3763AF2E1C572F&index=6

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2	1									2	1
CO2	3	3	2	1	1	1					1	3	2
CO3	2	3	1	1	1	1					1	3	2
CO4	2	2	3	3	3	1	1	2	1	1	2	2	3
CO5	1	1	1	1	1	3	3	2	3	3	2	3	3

Semester: III					
Data Structures Laboratory					
Course Code	**:**	1BCSL306	**Credit**	**:**	1
Hours/Semester (L:T:P:SL)	**:**	0:0:28:02	**CIE**	**:**	50 Marks
Total Hours	**:**	30 Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	PCCL	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Problem Solving through Programming, Data Structures and Algorithms.
Knowledge on Fundamental Concepts	Fundamentals of C Programing, Data Structures and algorithms Concepts.
Knowledge on Connected Tools (If any)	Edit plus, GCC, Code::Blocks, Visual Studio.

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the concepts of arrays, strings, stacks, queues, linked lists, trees, graphs, and hashing used in data structure applications.
CLO2	Interpret the working principles of various data structures and their operations through algorithmic representations.
CLO3	Describe the role of dynamic memory allocation, recursion, and traversal techniques in data structure implementation.
CLO4	Apply data structure concepts and C programming techniques to develop solutions for computational problems.
CLO5	Demonstrate the use of appropriate data structures to efficiently manage and process data in practical scenarios.

Experiments	
Sl. No.	**List of problems for which student should Develop C program and execute in the Laboratory.**
1.	To sort a given set of n integer elements using Insertion Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2.	A software company is developing a text processing and validation module for a messaging application. As part of this system, a Stack data structure (implemented using arrays) is required to manage and validate user inputs efficiently. Perform the following operations on a stack of integers: a. Insert elements into the stack to simulate incoming data. b. Remove elements from the stack to simulate processing or undo operations. c. Utilize stack operations to check whether a given sequence (e.g., number or string) is a palindrome, which is useful in pattern

	validation and data verification. d. Handle exceptional conditions such as: • Overflow (when attempting to insert into a full stack) • Underflow (when attempting to remove from an empty stack) • Display the current contents and status of the stack for monitoring purposes.
3.	Develop a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.
4.	Perform the following Stack Applications a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ b. Solving Tower of Hanoi problem with n disks
5.	A food delivery app handles customer orders as they arrive. The orders must be processed in the order they were received, making queue data structure ideal. Design and implement a C program to simulate this system using a Circular queue. Program should have options to add and remove the food orders in appropriate order for their service. (Array Implementation of Queue with maximum size MAX). Include C functions to perform the following operations. a. Add a new food order. b. Delete an older order. c. Display all current orders in queue. d. Demonstrate Overflow and Underflow conditions.
6.	A student To-Do Task Manager app allows the students to manage academic Task with member‘s id, title, due_date, Category, priority. The facilities to be provided for the users are as follows. a. Create Tasks. b. Add Task to start/end. c. Delete Task from start/end. d. View first/last task. e. Handle empty list conditions Develop a Program for the above operations using Singly Linked List.
7.	Perform the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit
8.	Perform the following operations on Singly Circular Linked List (SCLL) with header nodes a. Represent and Evaluate a Polynomial P(x,y,z) = 6x2y2z-4yz5+3x3yz+2xy5z-2xyz3

	b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) Support the program with appropriate functions for each of the above operations
9.	Perform the following operations on Binary Search Tree (BST) of Integers. a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in In-order, Pre-order and Post-order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
10.	Write a Program for the following operations on Graph(G) of Cities a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method.
11.	Write a Program to a. Find the transitive closure using Warshall's algorithm. b. Design and implement C Program to solve All-Pairs Shortest Paths problem using Floyd's algorithm.
12.	Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Develop a Program in C that uses Hash function H: K →L as H(K)=K mod m (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.

Open Ended Experiments	
1	Selecting the best Algorithm for student marks.
2	Hospital patient token management systems.
3	Ride -sharing app for active driver list.
4	College admission rank list using Binary Search Trees.
5	City bus route optimization using Graphs.

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply linear data structures to solve computational problems.
CO2	Implement dynamic linked data structures for efficient data management.
CO3	Analyze algorithms for sorting, searching, recursion, and expression processing.
CO4	Evaluate tree and graph structures for solving complex computational problems.
CO5	Design optimized solutions using graph algorithms and hashing techniques.

Textbooks	
1	"Data Structures Schaum's Outlines", Seymour Lipschutz, 1st Ed, McGraw Hill, 2014, ISBN 978-1259029967.
2	"Fundamentals of Data Structures in C", Ellis Horowitz, Sartaj Sahni and Susan AndersonFreed, 2nd Ed, Universities Press, 2014, ISBN 9780929306407.
3	"Introduction to the Design and Analysis of Algorithms", Anany Levitin, 3rd Edition, Pearson, 2021, ISBN13: 9780137541133.

Reference Books	
1	"Data Structures using C", Reema Thareja, 3rd Edition, 2023, Oxford press, ISBN 978-9354977190.
2	"A Pseudo-code approach with C", Gilberg & Forouzan, 2nd Edition, Cengage Learning,2014, ISBN 978-8131503140.
3	"An Introduction to Data Structures with Applications", Jean-Paul Tremblay & Paul G. Sorenson, 2nd Edition, McGraw Hill, 2017 ISBN 978-0074624715.

Web links and Video Lectures (e – Resources)	
1	Virtual Labs : http://cse01-iiith.vlabs.ac.in/
2	https://nptel.ac.in/courses/106102064
3	http://www.digimat.in/nptel/courses/video/106106145/L01.html
4	https://www.youtube.com/watch?v=SFcKedpnqwg&list=PLFW6lRTa1g83G4WDJOXocqhy_XdMraBJW

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2	1		3						2	3	2
CO2	3	2	2		3						2	3	2
CO3	3	3	2	2	3						2	2	3
CO4	3	3	3	2	3						2	2	3
CO5	3	3	3	3	3					1	3	3	3

Semester: III					
Project Management with Git Laboratory					
Course Code	:	1BCSL307A	**Credit**	:	1
Hours/Week (L:T:P:SL)	:	0:0:28:2	**CIE**	:	50 Marks
Total Hours	:	30Hrs	**SEE**	:	50 Marks
Course Type	:	AEC	**SEE Duration**	:	2 Hours

Prerequisites:	
Courses	Problem Solving through Programming, Python Programming.
Knowledge on Fundamental Concepts	File Handling, shell/terminal usage and Basic CLI Navigation.
Knowledge on Connected Tools (If any)	IDLEs.

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamental concepts and workflow of version control system.
CLO2	Familiarize with Git commands to manage code repositories.
CLO3	Learn various commands for handling the repositories.
CLO4	Utilize Git tools for advanced version control system operations.
CLO5	Perform Git history for undo operations.

Sl. No	**Lab Experiments**
1.	**Git Setup & Repository** Demonstrate the installation and configuration of Git, and create and initialize a repository by adding and committing files.
2.	**Basic File Tracking** Examine and track file changes using Git commands and interpret repository status and commit history.
3.	**Branch Creation & Switching** Create and manage branches and switch between them to support parallel development.
4.	**Branch Merging** Integrate changes from different branches by performing merge operations and evaluate the results.
5.	**Stashing Changes** Apply Git stash operations to temporarily save and restore working directory changes during context switching.
6.	**Working with Remote Repositories** Demonstrate cloning and manage working directories for remote repositories.
7.	**Fetching & Rebasing** Analyze and apply fetch and rebase operations to synchronize and update local branches with remote repositories.

8.	**Merge with Custom Message** Execute branch merging operations with custom commit messages to maintain clear project history.
9.	**Tags & Advanced Operations** Apply and manage advanced Git operations such as tagging and cherry-picking for version control.
10.	**Git History Analysis & Undo** Analyze, interpret, and modify Git history by filtering commits and performing undo operations.

Course Outcomes: After completing the course, the students will be able to	
CO1	Implement Git repository initialization, file tracking, and commit operations to maintain version-controlled projects.
CO2	Examine repository status, commit history, and file modifications to identify and manage development changes.
CO3	Utilize branching, stashing, and merging techniques to facilitate parallel development and code integration.
CO4	Integrate local and remote repository updates using cloning, fetching, rebasing, and customized merge operations.
CO5	Assess repository versions and development history to ensure project consistency.

Textbooks	
1	"Pro Git", Scott Chacon & Ben Straub, Apress®, 2nd edition, 2009 ISBN:978-1-4842-0076-6.
2	"Version Control with Git", Prem Kumar Ponuthorai & Jon Loeliger, O'Reilly Media, Inc., 3rd Edition, 2022, ISBN: 978-1-4920-9119-6.
3	"Learn Git in a Month of Lunches", Rick Umali,1st Edition, Manning Publications (distributed by Simon & Schuster, 2015, ISBN-10: 1617292419.

Web links and Video Lectures (e – Resources)	
1	https://research.computing.yale.edu/sites/default/files/files/Version%20Control%20wit h%20Git%281%29.pdf?utm_source=chatgpt.com
2	https://onlinecourses.swayam2.ac.in/aic20_sp10/preview
3	https://www.youtube.com/watch?v=SNryJ5UDI1w
4	https://www.cccwi.de/wp-content/uploads/2014/03/git-intro.pdf?utm_source=chatgpt.com

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	1			3			2	2		1	2	2
CO2	2	2	2		3			2	2	1	1	1	1
CO3	2	2	2		3			2	2	1	1	1	1
CO4	1	2			3			2	2	1	1	2	2
CO5	2	2	2	1	3			2	2	1	2	1	1

Semester: III					
Web Application Development with HTML and PHP					
Course Code	**:**	1BCSL307B	**Credit**	**:**	1
Hours/Semester (L:T:P:SL)	**:**	0:0:28:02	**CIE**	**:**	50 Marks
Total Hours	**:**	30Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	AEC	**SEE Duration**	**:**	2 Hours

Prerequisites:	
Courses	Problem Solving using Programming
Knowledge on Fundamental Concepts	Computer Basics, Basic Logic Building, Text Editing Tools, Web and Internet Basics
Knowledge on Connected Tools (If any)	Notepad++

Course Learning Objectives: This course will enable the students to	
CLO1	Understand core web programming concepts.
CLO2	Learn the syntax, structure, and elements of HTML/XHTML.
CLO3	Explore the use of PHP in web application development.
CLO4	Develop knowledge of JavaScript and DOM for interactive web pages.
CLO5	Explore AI integration in modern web applications.

Experiments	
Sl. No.	**List of problems for which student should Develop program and execute in the Laboratory**
1.	Develop the HTML page named as “Myfirstwebpage.html”. Add the following tags with relevant content. 1. Set the title of the page as “My First Web Page” 2. Within the body use the following tags: a) Moving text = “Basic HTML Tags” b) Different heading tags (h1 to h6) c) Paragraph d) Horizontal line e) Line Break f) Block Quote g) Pre tag h) Different Logical Style (<b>,<u>,<sub>,<sup>etc).
2.	Develop the HTML page named as “Table.html” to display your class time table. a) Provide the title as Time Table with table header and table footer, row-span and col-span etc. b) Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours.) c) Provide colour options for rows.
3.	Develop an external style sheet named as “style.css” and provide different styles for h2, h3, hr, p, div, span, time, img & a tags. Apply different CSS selectors for tags and demonstrate the significance of each.
4.	Develop HTML page named as “registration.html” having variety of HTML input elements with background colors, table for alignment & provide font colors & size using CSS styles.

5.	Develop HTML page named as "newpaper.html" having variety of HTML semantic elements with background colors, text-colors & size for figure, table, aside, section, article, header, footer… etc.
6.	Apply HTML, CSS and JavaScript to design a simple calculator to perform the following operations: sum, product, difference, remainder, quotient, power, square-root and square.
7.	Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
8.	Develop a web page to demonstrate DOM manipulation by dynamically changing content, style, and behavior based on user interaction.
9.	Write a PHP program to display a digital clock which displays the current time of the server
10.	Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
11.	Develop a PHP program (with HTML/CSS) to sort the student records which are stored in the database using selection sort.
12.	Develop a web-based AI spam detection application to identify spam messages using predefined keywords.

Open Ended Experiments	
1	Design and develop a responsive personal portfolio website using HTML, CSS, and JavaScript. The website should contain personal information, education details, skills, achievements, projects, and a contact form with proper validation. Apply suitable styling and interactive features to improve the user experience.
2	Develop a Student Record Management System using PHP and database connectivity to perform operations such as adding, updating, deleting, searching, and sorting student records. Design an attractive user interface using HTML/CSS and display records in tabular format.
3	Develop an AI-based web application to analyze user-entered text using predefined rules or keywords. The application may perform tasks such as spam detection, sentiment analysis, email categorization, FAQ chatbot, or fake news keyword identification. Implement the solution using HTML, CSS, JavaScript, and PHP.
4	Develop a web application that analyzes student performance, interests, or selected topics and recommends suitable learning resources, courses, or practice questions using machine learning-based recommendation techniques. The system should provide dynamic suggestions and track user interaction for improvement.
5	Develop a web application that accepts an image upload and automatically generates a caption or description using a pre-trained deep learning model. Integrate the model with PHP/Python backend and display the generated caption on the web page. This helps students understand image processing, CNN, and AI integration in web applications.

Course Outcomes: After completing the course, the students will be able to	
CO1	Interpret the fundamentals of web programming concepts to build structured web pages.
CO2	Design interactive and responsive web pages using HTML and CSS.
CO3	Apply JavaScript for dynamic and interactive content.
CO4	Analyze server-side processes to design and optimize PHP applications.
CO5	Develop web-based AI applications.

Textbooks	
1	"Programming the World Wide Web", Robert W Sebesta, 6th Edition, 2008, Pearson Education, ISBN 978-0-13-213081-3.
2	"Fundamentals of Web Development", Randy Connolly, Ricardo Hoar, 1st Edition, 2016, Pearson Education, ISBN: 978-9332575271.

Reference Books	
1	"Internet & World Wide Web How to program", M. Deitel, P. J. Deitel, A. B. Goldberg, 4th Edition, 2008, Pearson Education / PHI, ISBN 0-13-175242-1.
2	"Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5", Robin Nixon, 4th Edition, 2015, O"Reilly Publications, ISBN: 978-9352130153.
3	"PHP and MySQL Web Development", Luke Welling, Laura Thomson, 5th Edition, 2016, Pearson Education, ISBN: 978-9332582736.
4	"Professional JavaScript for Web Developers", Nicholas C Zakas, 3rd Edition, 2012, Wrox/Wiley India, ISBN: 978-8126535088.

Web links and Video Lectures (e – Resources)	
1	Fundamentals of WEB Programming: https://www.youtube.com/watch?v=DR9dr6gxhDM
2	HTML and XHTML: https://www.youtube.com/watch?v=A1XlIDDXgwg
3	CSS:https://www.youtube.com/watch?v=J35jug1uHzE
4	Java Script and HTML Documents: https://www.youtube.com/watch?v=Gd0RBdFRvF0
5	https://www.w3schools.com/

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2	2	1								2	1
CO2	2	2	2	2	2				1	2		3	2
CO3	2	3	2	2	2				1	2		3	2
CO4	3	2	1	1	3				1	2	1	3	2
CO5	3	2	2	2	3	2	2		1	2	1	3	3

Semester: III					
Intelligent Design Controller Laboratory					
Course Code	**:**	1BCIL307C	**Credit**	**:**	1
Hours/Semester (L:T:P:SL)	**:**	0:0:28:2	**CIE**	**:**	50 Marks
Total Hours	**:**	30	**SEE**	**:**	50 Marks
Course Type	**:**	AEC	**SEE Duration**	**:**	2 Hours

Prerequisites:	
Courses	Computer Organization, Basic programing.
Knowledge on Fundamental Concepts	Basic Mathematics.
Knowledge on Connected Tools (If any)	IDE, Compiler.

Course Learning Objectives: This course will enable the students to	
CLO1	Demonstrate the foundational of embedded system platforms such as Arduino Uno, ESP32, and vision-enabled devices for intelligent system design.
CLO2	Acquire practical skills in interfacing sensors, actuators, and peripheral devices for real-time embedded applications.
CLO3	Explore IoT communication concepts using wireless technologies for data acquisition and control systems.
CLO4	Apply Edge AI and TinyML concepts for developing intelligent, low-power embedded solutions.
CLO5	Develop real-world AIoT-based applications integrating embedded systems, IoT, and AI techniques.

Experiments	
Sl. No.	**List of problems for which student should Develop code and execute in the Laboratory using respected boards and external peripherals.**
1.	Study of Architecture Arduino Uno, ESP32, and Arduino Nicla Vision. (**Study experiment**)
2.	Design and Implementation of LED Interfacing and Digital Output Control using Arduino Uno.
3.	Develop an Environmental Monitoring System using DHT11 Sensor with LCD/OLED Display Interface.
4.	Implement a Smart Distance-Based Actuation System Using an Ultrasonic Sensor and Servo Motor
5.	Develop an IoT-Based Wireless LED Control System Using the ESP32 Wi-Fi Module
6.	Design and Implementation of a Wi-Fi-Enabled DC Motor Control System Using ESP32 for Remote Automation
7.	Develop a Bluetooth-Based Wireless Device Control System Using ESP32 and a Mobile Application Interface
8.	Implement of IoT Communication Using the MQTT Publish–Subscribe Protocol with ESP32
9.	Design and Development of an Image Acquisition System Using Arduino Nicla Vision for Edge AI Applications
10.	Implement a Real-Time Motion Detection System Using the Frame Differencing Technique on Arduino Nicla Vision

11.	Design and Implementation of an Edge AI-Based Object Detection System Using TensorFlow Lite and Edge Impulse on Arduino Nicla Vision

Open Ended Experiments (Mini Project)	
1	**Develop a project on to illustrate the application on** i. Smart Agriculture ii. Smart Healthcare iii. Smart Surveillance iv. Smart Transportation v. Smart Home vi. Industrial Automation vii. Other

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the architecture and working of embedded platforms (Arduino, ESP32, Nicla Vision).
CO2	Implement interfacing of sensors and actuators for real-time applications.
CO3	Develop IoT-based systems using Wi-Fi, Bluetooth, and communication protocols.
CO4	Analyze Edge AI applications using TinyML tools (TensorFlow Lite, Edge Impulse).
CO5	Build an AIoT-based solution for real-world problems.

Text Books	
1	Exploring Arduino: Tools and Techniques for Engineering Wizardry, Jeremy Blum, Wiley India / John Wiley & Sons, 2nd Edition, 2019. ISBN: 978-1-119-40530-6
2	Internet of Things – A Hands-on Approach, Arshdeep Bahga and Vijay Madisett , Universities Press, 1st Edition, 2015. ISBN: 978-8173719547.
3	Kolban's Book on ESP32 (or Kolban's Book on the ESP32 & ESP8266), Neil Kolban, Self-Published via Leanpub, 1st Edition, 2016. https://esp32.com/viewtopic.php?t=287#p1240
4	TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, Pete Warden and Daniel Situnayake, O'Reilly Media, 1st Edition, 2020. ISBN: 978-9352139606.

Reference Books	
1	Programming Arduino: Getting Started with Sketches, Simon Monk, McGraw-Hill Education, 3rd Edition, 2022. ISBN: 978-1264676989
2	AI at the Edge: Solving Real-World Problems with Embedded Machine Learning, Daniel Situnayake & Jenny Plunkett, O'Reilly Media, 1st Edition, 2023. ISBN: 978-1098120207

Web links and Video Lectures (e – Resources)	
1	https://docs.arduino.cc/hardware/nicla-vision/
2	https://www.youtube.com/watch?v=ILh38jd0GNU&list=PL5efXgSvwk9UCtJ6JKTyWAccSVfTXSlA3
3	https://www.youtube.com/playlist?list=PLGs0VKk2DiYw-L-RibttcvK-WBZm8WLEP
4	https://www.youtube.com/watch?v=xPlN_Tk3VLQ
5	https://docs.edgeimpulse.com/
6	https://docs.arduino.cc/tutorials/nicla-vision/image-classification/

CO – PO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2			2							3	2
CO2	3	3	2		3				2			3	3
CO3	2	3	3	2	3	1	1		2	1		2	3
CO4	2	3	2	3	3				2			2	3
CO5	3	3	3	3	3	2	2	1	3	2	2	3	3

Semester: III					
Data Visualization with Python					
Course Code	**:**	1BCSL307D	**Credit**	**:**	1
Hours/Semester (L:T:P:SL)	**:**	0:0:28:02	**CIE**	**:**	50 Marks
Total Hours	**:**	30Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	AEC	**SEE Duration**	**:**	2 Hours

Prerequisites:	
Courses	Python.
Knowledge on Fundamental Concepts	Programming Skills.
Knowledge on Connected Tools (If any)	Jupyter notebook, Anaconda.

Course Learning Objectives: This course will enable the students to	
CLO1	Demonstrate the use of IDLE or PyCharm IDE to create Python Applications.
CLO2	Demonstrate the ability to import, clean, and preprocess datasets using Python libraries such as NumPy and Pandas.
CLO3	Create and visualize data using Matplotlib, Seaborn, and Bokeh.
CLO4	Working with Plotly for 3D, time series and maps
CLO5	Develop a data analytics mini-project using Python for data preprocessing, multi-dataset handling, and visualization to generate insights.

Experiments	
Sl. No.	**List of problems for which student should Develop python program and execute in the Laboratory**
1.	Design a Python program using the NumPy library to create arrays representing *months* and their corresponding *revenue*. Using an appropriate plotting technique, generate a line chart to visualize the revenue trend across the months. Ensure the chart is properly labelled with a title, axis labels, and grid for better readability.
2.	Design a python program to preprocess raw dataset by handling missing values, duplicates and inconsistent data before visualization.
3.	Given dataset containing Date, Sales, Expenses, Customer Age, and Category. Using only the Pandas. plot() built-in interface, perform the following high-priority visualization tasks to extract meaningful business insights. a) Generate a line plot comparing Sales and Expenses over the Date index. Set the figure size to (10, 5), enable the grid, and add a title. b) Visualize the spread of Customer Age using a histogram with exactly 20 bins. Calculate the total sales per Category and represent the data using a Horizontal Bar Chart (kind='barh').
4.	Design a Python program which explains uses of customizing Seaborn plots with Aesthetic functions.
5.	a) Design a python program to explain working with Bokeh line graph using Annotations and legends. b) Design a python program for plotting different types of plots using Bokeh.
6.	Develop a python program to draw 3D Plots using Plotly Libraries.
7.	a) Develop a python program to draw Time Series using Plotly Libraries.

	b) Design a python program for creating maps using Plotly Libraries.
8.	For the given dataset that contains immigration details to Canada from 1980 to 2013, • Create an area plot for top 6 immigrant countries from 1990 to 2013. • Create and year-wise immigrant bar chart from India to Canada during the period of 1980 to 2013. • Create a boxplot for Indian, Phillipin and China immigrants. • Show the total no. of immigrants from India and France countries using Area Chart and Pie chart. Create a scatter Histogram for the immigrants from Fiji and Singapore in the year 2013.
9.	For the given data set that contains the data of flights that were on time in January for the years 2019 and 2020. Using the two data sets visualize the data using matplotlib and plotly libraries to depict the following: • Show the difference in statistics for distance for both the years using the appropriate plotting technique. • Visualize the no. of flights whose destination airport id is 11778 and 11267 using a bar plot or bar chart. Create a Sunburst Plot for both the years depicting the difference among them.
10.	Visualize the given Placement Data Full Class dataset that contains details about Campus Recruitment using the below techniques for appropriate dimensions and differentiate between the two techniques: • Histogram and Bar Chart [For histogram let no. of bins = 10]. • Facet Plot and Pair Plot. Area Chart and Pie Chart [For yes or no data].

Open Ended Experiments	
1	To design and develop a **real-time data analytics application** using Python libraries such as NumPy, Pandas, Matplotlib, Seaborn, Bokeh, or Plotly. To apply **data preprocessing, visualization, and analytical techniques** for solving real-world problems. **Project Requirements** • Students can select **any real-world problem domain**. • The application must include: o **Minimum Two or more datasets/tables** o Proper **data preprocessing** (handling missing values, duplicates, inconsistencies) o Use of **visualization techniques** for insights o Generation of **meaningful analytical reports**

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply data analysis and visualization techniques using Python libraries to extract insights from datasets.
CO2	Implement Matplotlib to create different types of 2D plots for effective data visualization and interpretation.
CO3	Design advanced visualizations using Seaborn and Bokeh.
CO4	Develop interactive and advanced 3D/time-series visualizations using Plotly.

CO5	Apply Python tools and libraries to preprocess data, combine multiple datasets, and create visualizations for data analysis.

Textbook	
1	"Python Data Analytics - With Pandas", NumPy, and Matplotlib, Fabio Nelli, 3rd Edition, 2023, A Press Publishing, ISBN-13 (pbk): 978-1-4842-9531-1

Reference Book	
1	"Intro to Python for Computer Science and Data science", Paul Deitel and Harvey deitel, 1st edition, 2020, Pearson Education Limited, ISBN 10: 1-292-36490-4 (print) , ISBN 13:978-1-292-36490-2 (print), ISBN 13: 978-1- 292-36493-3

Web links and Video Lectures (e – Resources)	
1	Python https://www.youtube.com/watch?v=_uQrJ0TkZlc
2	https://www.geeksforgeeks.org/data-visualization-using-matplotlib/

CO – PO – PSO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2		2	3					1	1	3	2
CO2	2	2		2	3					1		2	2
CO3	2	2	2	2	3					1		2	3
CO4	2	2	3	2	3				1	1		2	3
CO5	3	3	2	3	3				1	1	2	3	3

Semester: III					
Community Project					
Course Code	:	1BCP308	**Credit**	:	1
Hours/Semester (L:T:P:SL)	:	0:0:0:30	**CIE**	:	50 Marks
Total Hours	:	30	**SEE**	:	50 Marks
Course Type	:	SDC	**SEE Duration**	:	2 Hours

Course Learning Objectives: This course will enable the students to	
CLO1	Foster civic engagement and social responsibility by immersing students in communities to identify societal, welfare, and sustainability challenges.
CLO2	Apply engineering principles and interdisciplinary thinking to analyze community problems and design practical solutions.
CLO3	Develop professional skills in teamwork, project management, and effective communication with diverse community stakeholders.
CLO4	Instill an understanding of sustainable development, engineering ethics, and the long-term socio-economic impacts of technical interventions.
CLO5	To cultivate ethical engineering practices and an appreciation for sustainable development by analyzing the long-term impacts of technology on society.

Restrictions and professional conduct:

Community / Societal Projects shall remain academic, socially responsible, technically focused, and politically neutral. Students and faculty shall maintain professionalism, ethical conduct, public respect and institutional integrity while interacting with communities, public groups, organizations and stakeholders.

Students and faculty shall avoid the following during project execution:

- Association with political party activities, campaigns or propaganda.
- Participation in activities that may disturb communal harmony, public order or institutional reputation.
- Use of the project platform for personal, political, ideological or commercial promotion.
- Disrespectful, discriminatory or unethical interaction with community members or stakeholders.

Projects must also recognize constraints related to cost, resources, environment and safety, and stakeholder behavior/cooperation/acceptance when planning and implementing activities.

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:

- Interaction with local rural or urban stakeholders such as local leaders, corporation/municipality personnel, NGOs, panchayat representatives, etc.
- Review of issues highlighted in print and electronic media, and social networks (WhatsApp, Facebook, X, Instagram, etc.).
- After selecting a broad topic, students shall collect data, conduct a feasibility survey, estimate timelines and resources, approximate expenditure, and then define the problem statement and objectives clearly.

List of Project Topics and Engineering Outcomes:

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

Sl. No.	Project title	Measurable engineering outcome	Mapped POs	SDGs
1	Rural and urban water conservation and management systems	Prepare a site-specific water-saving plan, estimate water demand–supply gap, and propose a feasible conservation intervention with quantified savings.	PO1, PO2, PO3, PO4, PO10, PO11	SDG 6, SDG 11, SDG 12
2	Waste management and recycling	Conduct waste audit, classify waste streams, and design a segregation/recycling workflow with measurable reduction in mixed waste.	PO2, PO3, PO4, PO6, PO8, PO10	SDG 11, SDG 12, SDG 13
3	Afforestation	Identify a degraded area, prepare plantation layout, and demonstrate survival-rate tracking and maintenance schedule for saplings.	PO2, PO3, PO4, PO6, PO7, PO11	SDG 11, SDG 13, SDG 15
4	Biogas generation and power generation	Estimate organic waste availability, size a small biogas system, and document expected gas output or energy yield.	PO1, PO2, PO3, PO4, PO5, PO10	SDG 7, SDG 11, SDG 12
5	DPR for solar power rural plant	Prepare a DPR with load assessment, panel sizing, cost estimate, and projected energy generation and payback.	PO1, PO2, PO3, PO4, PO5, PO10	SDG 7, SDG 9, SDG 11
6	DPR for electrification of village/tribal community	Map unserved/underserved households, propose electrification layout, and estimate coverage, losses, and implementation cost.	PO1, PO2, PO3, PO4, PO10, PO11	SDG 7, SDG 9, SDG 10
7	DPR for rural drainage system	Survey drainage bottlenecks, develop a drainage layout, and estimate runoff handling capacity and flood-risk reduction.	PO1, PO2, PO3, PO4, PO6, PO10	SDG 6, SDG 9, SDG 11
8	Sustainability and environmental health projects	Assess local environmental health issues and propose interventions with measurable improvement in hygiene, safety, or resource use.	PO2, PO3, PO4, PO6, PO7, PO8	SDG 3, SDG 6, SDG 11, SDG 12
9	Plastic free campaigns	Measure plastic usage before and after awareness intervention and report percentage reduction or behavioural change.	PO2, PO3, PO6, PO7, PO8, PO10	SDG 11, SDG 12, SDG 13

10	Health and nutrition awareness	Conduct baseline and post-program awareness survey and document improvement in awareness scoreor	PO2, PO6, PO7, PO8, PO9, PO10	SDG 2, SDG 3, SDG 11
11	Societal projects involving health wellness	Identify a community health need and develop an awareness/service module with measurable participation and feedback outcomes.	PO2, PO3, PO6, PO8, PO9, PO10	SDG 3, SDG 11, SDG 17
12	Air quality monitoring system	Collect air-quality data, compare with reference limits, and present trends along with possible mitigation suggestions.	PO2, PO3, PO4, PO5, PO6, PO10	SDG 3, SDG 11, SDG 13
13	Plastic waste reduction campaign	Track collection, segregation, and replacement actions, and quantify reduction in plastic waste generation or littering.	PO2, PO3, PO6, PO7, PO8, PO10	SDG 11, SDG 12, SDG 13
14	Urban greening projects	Prepare greening plan, identify planting locations, and assess environmental benefits such as shade, aesthetics, or heat mitigation.	PO2, PO3, PO4, PO6, PO7, PO11	SDG 11, SDG 13, SDG 15
15	Smart agriculture solution	Design a low-cost monitoring or irrigation support solution and estimate improvement in water use efficiency or crop support.	PO1, PO2, PO3, PO4, PO5, PO10	SDG 2, SDG 6, SDG 9
16	Smart irrigation system	Estimate crop water requirement, propose control logic, and quantify water savings compared with conventional irrigation.	PO1, PO2, PO3, PO4, PO5, PO10	SDG 2, SDG 6, SDG 12
17	Youth mentoring programmes	Deliver a structured mentoring activity and measure participation, engagement, and skill/awareness improvement.	PO6, PO8, PO9, PO10, PO11	SDG 4, SDG 10, SDG 17
18	Adult education	Conduct learner-needs assessment, prepare instructional content, and measure improvement in basic literacy or numeracy indicators.	PO6, PO8, PO9, PO10, PO11	SDG 4, SDG 10, SDG 17
19	Food security initiatives /	Estimate storage need, design a simple cold-chain concept, and	PO1, PO2,	SDG 2, SDG 9,

	cold storage system	document reduction in spoilage or wastage.	PO3, PO4, PO5, PO10	SDG 12
20	Human rights awareness with legal systems	Prepare and deliver an awareness module and assess increase in legal awareness and access-to-rights understanding.	PO6, PO7, PO8, PO9, PO10	SDG 5, SDG 10, SDG 16
21	House tax collection through digital media	Develop a simple digital workflow or awareness support and measure improvement in compliance, convenience, or response time.	PO2, PO3, PO4, PO5, PO10, PO11	SDG 9, SDG 11, SDG 16
22	Use of application / application program	Build or demonstrate a useful community app and measure usability, adoption, or efficiency improvement.	PO1, PO3, PO4, PO5, PO9, PO10	SDG 9, SDG 11, SDG 17
23	Creation of Panchayat library	Plan the library model, catalogue resources, and measure usage, accessibility, or participation outcomes.	PO6, PO8, PO9, PO10, PO11	SDG 4, SDG 10, SDG 11, SDG 17

Suggested Learning Resources:

Textbooks:

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

Reference books:

1. NPTEL courses on Project-Based Learning and Social Innovation.
2. SWAYAM courses on community engagement and sustainability.
3. UN Sustainable Development Goals portal: https://sdgs.un.org
4. MyGov India and local civic body portals for community initiatives.
5. YouTube resources on design thinking, social innovation, and community service learning.

NGO and government field-action case studies related to sanitation, water, health, education, and environment.

Web links and Video Lectures (e-Resources):

1. NPTEL courses on Engineering Geology (IIT Roorkee, IIT Kharagpur).

2. NPTEL courses on Soil Mechanics and Foundation Engineering (IIT Delhi, IIT Bombay).
3. Virtual Labs – Geotechnical Engineering Virtual Lab (IIT Kanpur): https://vlab.co.in
4. Geological Survey of India (GSI) – Maps, reports, and publications: https://www.gsi.gov.in
5. YouTube channels: Civil Engineering Essentials, The Engineering Classroom (soil mechanics experiments).
6. American Society of Civil Engineers (ASCE) – Geotechnical resources and standards

Teaching Learning Process:

The suggested TLP for Community / Societal Project is as follows:

Week	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 /	Support computations and	Analyze data, perform	Technical

	sizing / mapping	technical review	calculations, prepare maps/charts	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 totaling 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 scaled to 50
- The SEE shall be conducted batch-wise by two examiners, one faculty from the same department and one from a different department appointed by the Principal.

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/industry/ co mmunity	8	Problem is clearly identified with strong evidence of societal relevance and well-defined scope.	Problem is Relevant and scope is clear with adequate supporting evidence.	Problem defined but justificatio n or evidence is weak.	Problem unclear or not relevant; little/no supporting evidence.
Formation of problem statement; Engineering approach, innovation & feasibility	10	Problem statement is precise; approach is sound, innovative and clearly feasible.	Problem statement adequate; approach feasible with minor gaps.	Problem statement partial; approach feasibility is marginal.	Problem statement missing or approach not feasible.
Interaction with stakeholders; Data collection, analysis & interpretation	8	Thorough stakeholder engagement; data collection is comprehensiv e; analysis is well-interpreted	Good engagemen t; data and analysis adequate with minor gaps.	Limited engagement ; data incomplete and analysis superficial.	Little/no stakehold er contact; insufficien t or no data/ analysis.
Project documentation; Clarity, organization & problem-solving demonstrated	10	Documentation is complete, well-organized, analytical, and shows strong problem-solving	Documentation complete and clear with some analytical content.	Documentation on basic with omissions; problem-solving limited.	Documentatio n missing/poorly organized; no clear problem-solving.
Communication& Presentation skills	6	Presentation is confident, structured, and answers queries fully; visuals effective.	Presentatio n clear and satisfactory; responds to most queries.	Presentati on acceptable but lacks clarity or confidence ; weak responses.	Presentati on poor or absent; unable to respond.
Contribution & Benefit to society (impact, sustainability)	8	Clear, demonstrable societal benefit and sustainability; realistic path to implementation.	Benefit plausible; sustainability considered with minor gaps.	Limited or unclear societal benefit; sustainability not well addressed.	No evident benefit or sustainability ; project not community-oriented.

SEE Rubric — Semester End Examination (Will be conducted for 100 marks; later scaled to 50 marks):

Criterion	Weight (marks)	Excellent (Full marks)	Good	Fair	Needs Improvement
Problem identification & Relevance to society/industry/ community	16	Problem and relevance are articulated clearly with strong evidence and context.	Problem relevance clear with reasonable evidence.	Problem stated but evidence/ context weak.	Problem poorly defined or irrelevant.
Formation of problem statement; Engineering approach, innovation & feasibility	20	Problem statement is precise; approach demonstrates engineering depth, novelty and feasibility.	Approach is technically sound and Feasible with minor limitations.	Approach basic; limited technical depth or feasibility concerns.	No viable engineering approach or infeasible plan.
Interaction with stakeholders; Data collection, analysis & interpretation	16	Excellent stakeholder engagement; rigorous data collection and insightful analysis.	Good engagement and acceptable analysis.	Superficial data collection and limited analysis.	Inadequate or no data; poor analysis.
Project documentation; Clarity, technical rigor & problem-solving demonstrated	20	Report is comprehensive , technically rigorous, and demonstrates systematic problem-solving.	Report is complete and technically correct with minor gaps.	Report lacks technical depth or has notable omissions.	Report missing key elements or technically incorrect.
Communication , Presentation & Viva-voce responses	12	Candidate presents expertly, defends work persuasively and answers all queries correctly.	Good presentation and satisfactory answers to most queries.	Presentation adequate but fails to answer some technical questions.	Poor presentation and unable to defend work.
Contribution & Benefit to society; Impact assessment & sustainability	16	Clear, evidence-based assessment of social impact and Sustainability with scaling recommendations	Reasonable impact discussion and sustainability considerations.	Limited evaluation of impact or sustainability.	No coherent assessment of impact or sustainability

Course Outcomes: After completing the course, the students will be able to	
CO1	Investigate a community to identify unmet needs, sustainability gaps, or technical challenges using structured empathy, observations, and interviews.
CO2	Design a practical, engineering-based solution that directly addresses the identified community issue.
CO3	Apply engineering concepts and project management principles to execute and deploy the designed solution within community constraints.

CO4	Assess the social, environmental, ethical, and economic impacts of the project to ensure sustainable development and community welfare.
CO5	Function effectively individually or in a team, and communicate project progress and outcomes clearly to technical evaluators and community members.

CO – PO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO1	2	3				3					
CO2	3		3		2	2					
CO3	3			2	2					3	
CO4						3	3				2
CO5								3	3		2

Semester: III						
Physical Education (PE) (Sports & Athletics) **Common to All Branches**						
Course Code	**:**	1BPE309		**Credits**	**:**	0
Hours/Week(L:T:P:SL)	**:**	24:0:0:16		**CIE**	**:**	**100 Marks**
Total Hours	**:**	24		**SEE**	**:**	**-**
Course Type	**:**	NCMC		**SEE Duration**	**:**	**-**

Course Learning Objectives: This course will enable the students to	
CLO1	Develop awareness about personal health, wellness, and the importance of an active lifestyle through pre-fitness evaluation and orientation.
CLO2	Improve general physical fitness through structured training in strength, speed, and warm-up exercises.
CLO3	Acquire sport-specific skills and techniques in traditional games such as Kabaddi or Kho-Kho through practical sessions.

Module – 1	**4 Hrs**
Orientation A. Lifestyle. B. Health & Wellness. C. Pre-Fitness test.	
Module – 2	**4 Hrs**
General Fitness & Components of Fitness A. Warming up (Free Hand exercises). B. Strength – Push-up / Pull-ups. C. Speed – 30 Mtr Dash.	
Module-3	**16 Hrs**
Specific games (Any one to be selected by the student) A. Kabaddi – Hand touch, Toe Touch, Thigh Hold, Ankle hold and Bonus. B. Kho-Kho – Giving Kho, Single Chain, Pole dive, Pole turning, 3-6 Up.	

Topics for Self learning	
Module – 1	Healthy lifestyle practices, balanced nutrition, sleep and stress management, and awareness of personal health and wellness.
Module – 2	Components of physical fitness, principles of fitness training, benefits of regular exercise, and methods for monitoring personal fitness levels.
Module – 3	Rules and regulations of Kabaddi and Kho-Kho, history and development of indigenous sports, basic game strategies, sportsmanship, teamwork, and prevention of common sports injuries.

Course Outcomes: After completing the course, the students will be able to

CO1	Describe the principles of physical education, health, nutrition, fitness and wellness for maintaining a healthy and active lifestyle.
CO2	Perform exercises, sports, yoga, and recreational activities to improve physical fitness, mental well-being, teamwork and sportsmanship.
CO3	Apply health, fitness, and wellness practices to promote lifelong well-being and develop foundational skills for sports and fitness careers.

Scheme and Assessment for auditing the course and Grades:

Sl. No.	Activity	Marks
1.	Participation of student in all the modules.	20
2.	Quizzes – 2, each of 15 marks.	30
3.	Final presentation / exhibition / Participation in competitions/ practical on specific tasks assigned to the students.	50
	Total	**100**

CO – PO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1						3	2				2
CO2						3	2	3	3		2
CO3						3	3	2	2	2	3

Semester: III					
National Cadet Corps (NCC) – I **(General Awareness – Common Subjects) (Common to all Branches)**					
Course Code	**:**	1BNC309	**Credit**	**:**	0 (NCMC)
Hours/Semester (L:T:P:SL)	**:**	14:0:0:16	**CIE**	**:**	100 Marks
Total Hours	**:**	30	**SEE**	**:**	NA
Course Type	**:**	NCMC	**SEE Duration**	**:**	NA

Prerequisites:	
Courses	Nil
Knowledge on Fundamental Concepts	-
Knowledge on Connected Tools (If any)	-

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the aim, organisation and philosophy of training of the NCC, and appreciate the importance of national integration
CLO2	Acquire general awareness of disaster management and civil affairs, and the cadet's role during emergencies
CLO3	Gain general knowledge of social awareness, community development and the youth's role in nation building

Module – 1	**05 Hrs**
The NCC and National Integration – General Awareness The NCC: Aims and Objectives of NCC; Organisation, Training and NCC Song; Incentives for NCC Cadets. National Integration: Religions, Culture, Traditions and Customs of India; Importance and Necessity of National Integration. Freedom Struggle and Nationalist Movement in India; National Interests, Objectives, Threats and Opportunities. Problems/Challenges of National Integration; Unity in Diversity; National Integration Council; Contribution of Youth in Nation Building.	
Module – 2	**05 Hrs**
Disaster Management and Civil Affairs – General Awareness Civil Defence Organisation and the National Disaster Management Authority (NDMA); Types of Emergencies/Natural Disasters. Fire Services and Fire Fighting (general awareness); Essential Services and their Maintenance during a disaster. Assistance during Natural/Other Calamities (Flood/Cyclone/Earthquake/Accident); Setting up of Relief Camps. Collection and Distribution of Aid Material during Disaster Management.	
Module – 3	**04 Hrs**
Social Awareness and Community Development – General Awareness Basics of Social Service and its Need; Rural Development Programmes (MGNREGA, SGSY, NSAP, etc.); Role and Contribution of NGOs. Contribution of Youth towards Social Welfare; Civic Responsibilities; Family Planning (general awareness). Causes and Prevention of HIV/AIDS and Role of Youth; Counter Terrorism (general awareness); Corruption. Social Evils – Female Foeticide, Dowry, Child Abuse/Trafficking; Right to Information (RTI) and Right to Education (RTE).	

Topics for Self learning	
Module – 1	Reading on the aims/objectives and organisation of NCC from the Cadet's Handbook; a short note on any one national integration theme (e.g., Unity in Diversity)
Module – 2	Case-note on a recent natural disaster and the civil administration's/NDMA's response; awareness reading on relief-camp organisation
Module – 3	Short write-up on one rural development programme (e.g., MGNREGA); awareness reading on RTI/RTE provisions

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the aim, organisation and philosophy of training of the NCC and describe the importance of national integration
CO2	Describe the role of civil defence, NDMA and the cadet during disaster management and relief operations
CO3	Explain the elements of social awareness, community development programmes and the youth's civic responsibilities

Text Books	
1	Cadet's Handbook – Common Subjects (SD/SW), Directorate General NCC, Ministry of Defence, Government of India, Edition dated 06 May 2013 (or latest edition).
2	NCC Handbook for 'A', 'B' and 'C' Certificate Examinations, Upkar Prakashan, Latest Edition.

Reference Books	
1	NCC General Knowledge and Current Affairs, Arihant Publications, Latest Edition.
2	National Disaster Management Authority (NDMA) – guidelines and awareness publications, Latest Edition.
3	NCC Directorate/Group HQ training literature and circulars, as issued from time to time.

Web links and Video Lectures (e – Resources)	
1	Directorate General National Cadet Corps – https://www.nccindia.gov.in/
2	National Disaster Management Authority – https://ndma.gov.in

CO – PO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO1						3		2	2		
CO2						3	2	2			
CO3						3		2	2		

Semester: III						
Competency Enhancement Course-1 **Stream: CSE/ECE/ME/CV**						
Course Code	**:**	1BCE310		**Credits**	**:**	0
Hours/Week (L:T:P:SL)	**:**	36:0:0:24		**CIE**	**:**	**100 Marks**
Total Hours	**:**	36		**SEE**	**:**	-
Course Type	**:**	NCMC		**SEE Duration**	**:**	-

Prerequisites:	
Courses	Fundamentals of Mathematics & English
Knowledge on Fundamental Concepts	Arithmetic operations, fractions, decimals, and elementary algebra. comprehension of basic English
Knowledge on Connected Tools (If any)	**NA**

Course Learning Objectives: This course will enable the students to	
CLO1	Gain a strong foundation in aptitude skills such as number systems, percentages, ratio & proportion, and interest calculations to improve quantitative reasoning and problem-solving efficiency.
CLO2	Apply Vedic Mathematics techniques and shortcuts for rapid mental calculations, leading to increased accuracy and speed in aptitude-based assessments.
CLO3	Master logical reasoning topics like syllogisms, blood relations, puzzles and seating arrangements to develop structured thinking and effective decision-making skills.
CLO4	Understand and solve problems involving coding-decoding, visual sequences, clocks, calendars and analogies to enhance pattern recognition and reasoning speed.
CLO5	Utilize aptitude and reasoning skills to perform well in competitive exams, campus placements, and workplace situations, contributing to overall academic and professional success.

Module – 1	**6 Hrs**

Introduction to Aptitude and Vedic Mathematics

Understanding Aptitude

Overview of aptitude skills and their significance in problem-solving and decision-making. Types of aptitude skills commonly assessed in competitive exams and professional settings.

Vedic Mathematics Techniques

Introduction to Vedic Mathematics for faster and more efficient calculations. Key techniques and shortcuts to improve speed and accuracy in quantitative problems.

Importance of Aptitude Skills

The role of aptitude in career development and academic success. How strong aptitude skills enhance analytical thinking, logical reasoning, and performance in assessments.

Number System

Divisibility & Remainder, Multiples & Factors, Integers, LCM & HCF, Complete a number Series, Find the Missing Term and Wrong Term, Simplification, BODMAS Rule, Approximation, Decimals, Fractions, Surds & Indices

Percentage

Calculation-oriented basic percentage, Profit and Loss, Successive Selling type, Discount & MP, Dishonest Dealings

T-1,C1-11	
Module – 2	**6 Hrs**
Coding Decoding: Letter Coding, Number Coding, symbol coding Clocks and Calendar **Reasoning -** Blood Relation, Sense of Direction, Arithmetic & Alphabet. Visual Sequence, Visual analogy and classification. **Analytical Reasoning -** Single & Multiple comparisons, Linear Sequencing. **T-3-C2,3,4,6,7,10,12**	
Module-3	**6 Hrs**
Logical Aptitude - Syllogism, Venn-diagram method, Three statement syllogism, Deductive and inductive reasoning. Introduction to puzzle and games organizing information, parts of an argument, common flaws, arguments and assumptions. **Linear Seating Arrangement** Single or Double rows facing each other or away from each other in the same direction **Circular Seating Arrangement** Uni- & Bi-directional problems on Circular, Square, Rectangular, Hexagonal tables **T2-C2,5. R-1- C1-3**	
Module-4	**6 Hrs**
Ratio and Proportion Simple Ratios, Compound Ratios, Comprehend and Dividend, Direct & Indirect Proportions, Problems on ages, Mixtures & Allegation **T-1,C12,13&16**	
Module-5	**6 Hrs**
Partnerships Interest Simple Interest, Compound Interest, Mixed Interest, Installments. **T-1,C14,15,17&29**	

Topics for Self learning	
Module – 1	Divisibility rules and digital root techniques, Advanced Vedic Mathematics shortcuts (multiplication, squaring, cube roots), Roman numerals and number patterns
	Applications of percentages in banking and business, Practice using online aptitude platforms (IndiaBIX, Testbook, etc.)
Module – 2	Coding-Decoding and Analytical Reasoning, Mirror and water images, Paper folding and paper cutting, Embedded figures and figure completion
	Ranking and ordering problems, Introduction to non-verbal reasoning puzzles
Module – 3	Family tree puzzles, Floor and flat-based puzzles, Scheduling and arrangement puzzles
	Statement–Conclusion and Cause–Effect reasoning, Critical reasoning questions from competitive examinations
Module – 4	Ratio and Proportion, Variation (Direct, Inverse and Joint Variation). Average and weighted average, Time and Work fundamentals
	Pipes and Cisterns (introductory concepts), Real-life applications of ratio and proportion in engineering and finance
Module – 5	Partnership and Interest, Banking concepts related to loans and EMIs, Compound annual growth rate (CAGR)
	Basics of financial literacy and investment, Recurring Deposits (RD) and Fixed Deposits (FD) calculations, Introduction to annuities and financial planning

Course Outcomes: After completing the course, the students will be able to	
CO1	Solve numerical problems using Vedic Math and number systems to boost speed and accuracy.
CO2	Apply logical reasoning to puzzles, syllogisms, seating and classification problems.
CO3	Develop letter, number and symbol patterns and solve visual reasoning tasks.
CO4	Accurately solve problems on percentages, profit-loss, interest, ratios and partnerships.
CO5	Utilize aptitude and reasoning skills to excel in competitive exams and improve employability.

Text Books	
1	**Quantitative Aptitude for Competitive Examinations** by R.S. Aggarwal Chand Publishing. – 2023.
2	**Vedic Mathematics** by Bharati Krishna Tirthaji (ISBN- 978-8120801639), Sixteenth Reprint, Published by Motilal Banarsidass Publishers, - 1985
3	**Logical and Analytical Reasoning** by A.K. Gupta (ISBN- 978-9350127766), 39th Edition, Ramesh Publishing House - 2024

Reference Books	
1	**Puzzle Test & Seating Arrangements** by K. Kundan (ISBN: 9789394666023),1st Edition, BSC Publishing Company Pvt Ltd. - 2023
2	**Logical Reasoning and Data Interpretation for the CAT** by Nishit K. Sinha, (ISBN: 978-9353946616) 7th Edition, Pearson Education -2020

Web links and Video Lectures (e – Resources)	
1	https://www.youtube.com/watch?v=ZXcULvjpQlM
2	https://www.youtube.com/watch?v=ZXcULvjpQlM
3	https://www.youtube.com/watch?v=OHZ40sOT7mk
4	https://www.youtube.com/watch?v=9UtPPSX_vnk
5	https://www.youtube.com/watch?v=DIbjA3b21ow

CO – PO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO1	**3**	**2**									
CO2	**3**	**3**									
CO3	**2**	**3**									
CO4	**3**	**2**									
CO5	**2**	**2**			**1**				**2**		**2**

Semester: III					
Additional Mathematics-I **(For Lateral Entry Students of all the Branches)**					
Course Code	**:**	1BMATDIP311	**Credits**	**:**	0
Hours/Semester (L:T:P:SL)	**:**	14:0:0:14	**CIE**	**:**	100
Total Hours	**:**	28 Hrs	**SEE**	**:**	0
Course Type	**:**	NCMC	**SEE Duration**	**:**	-

Prerequisites:	
Courses	Basic Mathematics
Knowledge on Fundamental Concepts	Basic differentiation and integration
Knowledge on Connected Tools (If any)	Basic computer knowledge.

Course Learning Objectives: This course will enable the students to	
CLO1	Investigate the concepts of linear algebra to solve systems of linear equations and to determine eigenvalues and eigenvectors.
CLO2	Describe the vector differentiation and integration techniques to analyze scalar and vector Fields.
CLO3	Implement numerical techniques to solve algebraic and transcendental equations, perform interpolation using finite differences and compute numerical integration.

Module – 1	**05 Hrs**
Linear Algebra: Introduction, Rank of a matrix by elementary row operations, Consistency of system of linear equations, Solution by Gauss Elimination method. Eigenvalues and eigenvectors of a square matrix. Problems.	
Module – 2	**05 Hrs**
Vector Differentiation: Differentiation of vector functions. Scalar and vector point functions. Gradient, Divergence, Curl-simple problems. Solenoidal and irrotational vector fields-Problems.	
Module – 3	**05 Hrs**
Numerical Methods-I: Finite differences, Interpolation/extrapolation using Newton's Forward and Backward difference formulae (No derivation), Problems. Solution of polynomial and transcendental equations by Newton–Raphson and Regula–Falsi methods (no derivation), Problems. Numerical Integration: Simpson's $1/3^{rd}$ rule and $3/8^{th}$ rule, problems.	

Topics for Self-learning	
Module – 1	**Applications:** Analyze real-world problems such as traffic flow.
	Self-study: Inverse of a square matrix by Caley- Hamilton theorem.
Module – 2	**Applications**: Velocity and Acceleration in scalar and vector fields.
	Self-Study: Conservation of laws, Analysis of streamlines.
Module – 3	**Applications:** Area, Velocity and population.

	Self-Study: Bisection method, Lagrange's inverse Interpolation**.**

Course Outcomes: After completing the course, the students will be able to	
CO1	Solve the systems of linear equations, determine eigenvalues and eigenvectors and analyze real-world problems such as traffic flow.
CO2	Apply differentiation to solve the vector differentiation and integration techniques to analyze scalar and vector Fields.
CO3	Analyze the numerical techniques to solve algebraic and transcendental equations, perform interpolation using finite differences and compute numerical integration.

Textbook	
1	"Higher Engineering Mathematics", B. S. Grewal,44th Edition, 2021, Khanna Publishers, ISBN:978-81-933284-9-1. Chapter. 2, 28,8,29.

Reference Books	
1	"Advanced Engineering Mathematics", Kreyszig, E., Kreyszig, H., & Norminton, E. J. (2011), 10th Edition. Wiley India Pvt. Ltd. ISBN: 9788126554232
2	"A Textbook of Engineering Mathematics", Bali, N. P., & Goyal, M. Laxmi Publications (P) Ltd., New Delhi. ISBN: 9788131808320.
3	"Higher Engineering Mathematics", Dass, H. K., & Verma, R. Revised Edition. New Delhi: S. Chand Publishing. ISBN: 978812193890
4	"Higher Engineering Mathematics", Ramana, B. V. 25th Edition, McGraw Hill Education (India) Pvt. Ltd., ISBN: 9780070634190

Web links and Video Lectures (e – Resources)	
1	http://nptel.ac.in/courses.php?disciplineID=111
2	http://www.class-central.com/subject/math(MOOCs)
3	http://academicearth.org/

CO – PO MAPPING

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO1	3	3	2								1
CO2	3	3	3								1
CO3	3	3	2								1

	CSE Stream		**ECE Stream**		**CIVIL Stream**		**Mechanical Stream**	
	PSO1	**PSO2**	**PSO1**	**PSO2**	**PSO1**	**PSO2**	**PSO1**	**PSO2**
CO1	1			1		1		1
CO2	1			1		1		1
CO3	1			1		1		1

Institute Vision

Preparing Competent Engineering and Management Professionals to Serve the Society

Institute Mission

- Providing Students with a Sound Knowledge in Fundamentals of their branch of Study.
- Promoting Excellence in Teaching, Training, Research and Consultancy.
- Exposing Students to Emerging Frontiers in various domains enabling Continuous Learning.
- Developing Entrepreneurial acumen to venture into Innovative areas.
- Imparting Value based Professional Education with a sense of Social Responsibility.

Department Vision

Nurturing AI&ML technocrats to become proficient in solving real – world problems.

Department Mission

- Providing fundamentals to enhance niche expertise in AI&M, cognitive computing robotics and neural networks.
- Enhancing the Analytical and problem – solving abilities to meet the challenges of the future.
- Training the students with the latest technology to become sustainable professionals.
- Fostering a culture of collaboration and propagating industry – oriented innovation and research.
- Establishing a stimulating learning environment to create value – based and pioneering.

SJC INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU from 2024-25

P.B. No.20, B.B Road,

Chikkaballapur - 562 101, Karnataka