DIPLOMA CURRICULUM OF

COMPUTER SCIENCE AND ENGINEERING

(SECOND YEAR)

(3rd Semester)

(To be implemented from 2025-26)

Prepared by;

National Institute of Technical Teachers' Training & Research Kolkata

Block – FC, Sector – III, Salt Lake City, Kolkata – 700 106

Vetted by:

Domain experts from Polytechnics of Odisha

State Council for Technical Education & Vocational Training

Near Raj Bhawan, Unit-VIII, Bhubaneswar, Odisha

Table of Contents

SEMESTER – III

SL. No	Category of Course	Code No	Course Title	Study Scheme				Evaluation Scheme				Total Marks	Credits
				Pre-requisite	Contact Hours/ week			Theory		Practical			
					L	T	P	End Exam	Progressive Assessment	End Exam	Progressive Assessment		
1	Programme Core	CSEPC 201	Programming with C++		3	0	0	70	30	-	-	100	3
2		CSEPC 203	Programming with Python		3	0	0	70	30	-	-	100	3
3		CSEPC 205	Data Structures		3	0	0	70	30	-	-	100	3
4		CSEPC 207	Digital Electronics and Computer Organization		3	0	0	70	30	-	-	100	3
5		CSEPC 209	Algorithms		3	0	0	70	30	-	-	100	3
6		CSEPC 211	Programming with C++ Lab		0	0	4	-	-	15	35	50	2
7		CSEPC 213	Programming with Python Lab		0	0	4	-	-	15	35	50	2
8		CSEPC 215	Data Structures Lab		0	0	4	-	-	15	35	50	2
9		CSEPC 217	Digital Electronics Lab		0	0	4	-	-	15	35	50	2
10	Summer Internship	SI 201	Summer Internship – I*		0	0	0	-	-	15	35	50	2
TOTAL					**15**	**0**	**16**	**350**	**150**	**75**	**175**	**750**	**25**

***4-week internship after 2nd Semester**

SEMESTER - III COURSES

PROGRAMMING WITH C++

L	T	P		
3	0	0	**Total Marks: 100**	**Course Code: CSEPC 201**
Total Contact Hours				**Theory Assessment**
Theory	: 45Hrs			End Term Exam : 70
				Progressive Assessment : 30
Pre-Requisite	**: Nil**			
Credit	**: 3**			Category of Course : PC

RATIONALE:

Programming with C++ is a foundation course for any would-be IT professional. It gives exposure to the basic techniques of computer programming in current technological scenario. This course is most essential for any curriculum of Computer Science and Engineering.

LEARNING OUTCOMES:

After completion of the course, the students will be able to:

- Describe object-oriented programming (OOP) principles.
- Develop proficiency in C++ syntax and programming constructs.
- Implement advanced OOP features for software design.
- Demonstrate polymorphism and operator overloading.
- Handle exceptions and ensure robust program execution.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	Introduction to object oriented programming, user defined types, structures, unions, polymorphism, encapsulation. Getting started with C++ syntax, data-type, variables, strings, functions, default values in functions, recursion, namespaces, operators, flow control, arrays and pointers.	12
II	Abstraction mechanism: Classes, private, public, constructors, destructors, member data, member functions, inline function, friend functions, static members, and references. Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors.	11
III	Polymorphism: Binding, Static binding, Dynamic binding, Static polymorphism: Function Overloading, Ambiguity in function overloading, Dynamic polymorphism: Base class pointer, object slicing, late binding, method overriding with virtual functions, pure virtual functions, abstract	9

	classes.	
IV	Operator Overloading: This pointer, applications of this pointer, Operator function, member and non-member operator function, operator overloading, I/O operators.	9
V	Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration, unexpected exceptions	4

REFERENCES:

1.	Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India)
2.	ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education
3.	C++: The Complete Reference - Schildt, McGraw-Hill Education (India)
4.	C++ and Object Oriented Programming - Jana, PHI Learning.
5.	Object Oriented Programming with C++ - Rajiv Sahay, Oxford
6.	Mastering C++ - Venugopal, McGraw-Hill Education (India)

PROGRAMMING WITH PYTHON

L	T	P		**Course Code: CSEPC 203**	
3	0	0			
Total Contact Hours				**Theory Assessment**	
Theory	: 45Hrs		**Total Marks: 100**	End Term Exam	: 70
				Progressive Assessment	: 30
Pre-Requisite	**: Nil**				
Credit	**: 3**			Category of Course	: PC

RATIONALE:

Python is a versatile and beginner-friendly scripting language, renowned for its simplicity and readability. It enables rapid development of applications in fields like web development, data analysis, automation, and artificial intelligence. Learning Python equips students with a powerful toolset to solve complex problems and adapt to evolving technology demands.

LEARNING OUTCOMES:

After completion of the course, the students will be able to:

- Define Python's core syntax, data types, and key concepts of object-oriented programming.
- Explain how control structures and data structures function in Python.
- Implement Python programs using file handling, modules, and libraries like NumPy, Pandas, and Matplotlib for data analysis and visualization.
- Analyze Python scripts to identify and resolve logical or syntactic errors and optimize code using advanced techniques like recursion and lambda functions.
- Develop a real-world mini-project by integrating Python concepts such as OOP, libraries, and automation tools for practical problem-solving.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Python:** Overview of Python: Features and Applications Setting Up the Python Environment (Python Installation, IDEs), Python Syntax: Variables, Data Types, and Operators Writing, Executing, and Debugging Python Scripts	8
II	**Control Structures and Functions:** Conditional Statements: if, else, elif Loops: for, while, and Nested Loops Functions: Defining, Calling, and Scope of Variables Introduction to Lambda Functions and Recursion	8
III	**Data Structures in Python:** Lists, Tuples, Sets, and Dictionaries: Operations and Applications, List Comprehensions and Dictionary Comprehensions Working with Strings: Methods and Manipulation, Introduction to Python's Collections Module	9
IV	**File Handling and Modules:** File Operations, Reading, Writing, and	8

	Appending Files Working with CSV and JSON Files, Python Modules: Built-In Modules (e.g., math, os, datetime) Creating and Using Custom Modules	
V	**Object-Oriented Programming (OOP) in Python:** Understanding Classes and Objects Concepts of Encapsulation, Inheritance, and Polymorphism, Working with Magic Methods and Operator Overloading, Exception Handling in Python	7
VI	**Advanced Python and Applications:** Introduction to Libraries: NumPy, Pandas, Mini-Project: Developing a Python Script for a Real-World Problem	5

REFERENCES:

1.	John M. Zelle, Python Programming: An Introduction to Computer Science, 2nd Edition, Franklin, Beedle & Associates Inc., Portland, 2010.
2.	Al Sweigart**,** Automate the Boring Stuff with Python, 2nd Edition, No Starch Press, San Francisco, 2019.
3.	Reema Thareja, Python Programming Using Problem Solving Approach, Oxford University Press, New Delhi, 2017.
4.	Sheetal Taneja and Naveen Kumar, Python Programming: A Modular Approach with Graphics, Database, Mobile, and Web Applications, Pearson India, New Delhi, 2018.
5.	R. Nageswara Rao, Core Python Programming, Dreamtech Press, New Delhi, 2018.
6.	A.K. Sharma, Python for Beginners: With Hands-On Examples, BPB Publications, New Delhi, 2020.

DATA STRUCTURES

L	T	P		Course Code: CSEPC 205	
3	0	0			
Total Contact Hours				**Theory Assessment**	
Theory	: 45Hrs		**Total Marks: 100**	End Term Exam	: 70
				Progressive Assessment	: 30
Pre-Requisite	**: Nil**				
Credit	**: 3**			Category of Course	: PC

RATIONALE:

Data structures are at the core of any computational process. Almost all computer programs use data structures. So, the knowledge of data structures is required to build the foundation of an IT professional. This course is intended to equip the student with the fundamental knowledge.

LEARNING OUTCOMES:

After completion of the course, the students will be able to:

- Explain fundamental data structure concepts, classifications, and algorithm analysis.
- Apply linear data structures such as stacks, queues, and their variations.
- Develop linked list structures, including singly, doubly, and circular linked lists.
- Implement non-linear data structures like trees and perform operations such as insertion, deletion, and traversal.
- Describe graph representations and traversal techniques for efficient data organization.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Data Structures:** Basic Terminology, Classification of Data Structure, Operations on Data Structure, Asymptotic and worst-case analysis of algorithms.	8
II	**Linear Data Structures:** Stacks-Introduction to Stacks, Array Representation of Stacks, Operations on a Stack, Applications of Stacks-Infix-to-Postfix Transformation, evaluating Postfix Expressions. Queues: Introduction to Queues, Array Representation of Queues, Operations on a Queue, Types of Queues-DeQueue, Circular Queue, Applications of Queues-Round Robin Algorithm.	10
III	**Linked Lists:** Singly Linked List, Representation in Memory, Operations on a Single Linked List, Circular Linked Lists, Doubly Linked Lists, Linked List Representation and Operations of Stack, Linked List Representation and Operations of Queue.	14

IV	**Non-Linear Data Structures:** Trees-Basic Terminologies, Definition and Concepts of Binary Trees, Representations of a Binary Tree using Arrays and Linked Lists, Operations on a Binary Tree-Insertion, Deletion, Traversals, Types of Binary Trees. GRAPHS: Graph Terminologies, Representation of Graphs- Set, Linked, Matrix, Graph Traversals	13

REFERENCES:

1.	Seymour Lipschutz , Data Structures, Schaum's Outline Series, Tata-MacGraw-Hill
2.	Jay Wengrow, A Common-Sense Guide to Data Structures and Algorithms, Second Edition: Level Up Your Core Programming Skills, Pragmatic Bookshelf (2020)
3.	Seymour Lipschutz , Schaum's Outlines - Data Structures With C, Tata-MacGraw-Hill
4.	R. Venkatesan (Author), S. Lovelyn Rose (Author), Data Structures, Wiley
5.	Y. Langsam, M. J. Augenstein, A. M. Tanenbaum, Data Structures Using C and C++, Prentice Hall of India.

DIGITAL ELECTRONICS AND COMPUTER ORGANISATION

L	T	P		
3	0	0		Course Code: CSEPC 207
Total Contact Hours				**Theory Assessment**
Theory		: 45Hrs	**Total Marks: 100**	End Term Exam : 70
				Progressive Assessment : 30
Pre Requisite		**: Nil**		
Credit		**: 3**		Category of Course : PC

RATIONALE:

Digital Electronics and Computer Organization form the foundation of modern computing, enabling the design and functioning of digital systems. They cover essential concepts like logic circuits, memory, and system architecture, bridging hardware and software integration. Learning these topics equips students with the skills to understand and develop efficient, reliable computing systems.

LEARNING OUTCOMES:

After completion of the course, the students will be able to:

- Define key concepts of digital electronics, including number systems, Boolean algebra, and logic gates.
- Explain the principles behind combinational and sequential circuits, such as multiplexers, flip-flops, and counters, and their applications.
- Implement simplified logic circuits using Karnaugh Maps and Boolean algebra to solve real-world digital design problems.
- Analyze the instruction cycle, memory organization, and processor architecture to evaluate system performance and identify bottlenecks.
- Design to simulate a basic CPU operation or create a functional digital circuit using the concepts of digital electronics and computer organization.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Digital Electronics:** Difference Between Analog and Digital Signals Number Systems: Binary, Octal, Decimal, and Hexadecimal Conversion Between Number Systems, Binary Arithmetic Boolean Algebra: Basic Operations, Laws, and Simplification	8
II	**Logic Gates and Circuits: Logic Gates:** AND, OR, NOT, NAND, NOR, XOR, XNOR, Design and Simplification of Logic Circuits Using Boolean Algebra, Karnaugh Maps (K-Maps) for Simplification Practical Applications of Logic Gates in Real-World Circuits	7

III	**Combinational and Sequential Circuits:** Combinational Circuits: Multiplexers, Demultiplexers, Encoders, and Decoders, Sequential Circuits: Flip-Flops (SR, JK, D, T) and Their Applications Counters: Synchronous and Asynchronous Counters Registers and Shift Registers: Types and Uses	8
IV	**Fundamentals of Computer Organization:** Basic Structure of a Computer: CPU, Memory, Input/Output Devices Instruction Cycle: Fetch, Decode, Execute Memory Organization: Types of Memory (RAM, ROM, Cache, Virtual Memory), Introduction to Buses: Address Bus, Data Bus, and Control Bus	8
V	**Processor Architecture and Control:** Unit Introduction to Microprocessors and Microcontrollers, Basics of Arithmetic Logic Unit (ALU) and Control Unit, Instruction Set Architecture (ISA): RISC vs CISC Pipelining and Performance Optimization in Processors	7
VI	**Input/Output Systems and Advanced Topics:** I/O Devices and Interfaces: Keyboard, Mouse, Printers, and Storage Devices Interrupts and DMA (Direct Memory Access), Overview of Modern Trends: Multicore Processors, GPUs, and Embedded Systems, Mini- Project: Design a Simple Digital Circuit or Simulate a Basic CPU Operation	7

REFERENCES:

1.	M. Morris Mano, Digital Design, 5th Edition, Pearson Education, India, 2013.
2.	David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th Edition, Morgan Kaufmann, San Francisco, 2013.
3.	V. Rajaraman and T. Radhakrishnan**,** Digital Logic and Computer Organization, 1st Edition, PHI Learning, New Delhi, 2006.
4.	R. P. Jain**,** Modern Digital Electronics, 4th Edition, Tata McGraw-Hill Education, New Delhi, 2009.
5.	Dhananjay M. Dhamdhere**,** Computer Organization and Assembly Language Programming, 1st Edition, Tata McGraw-Hill Education, New Delhi, 1987.

ALGORITHMS

L	T	P		
3	0	0	Total Marks: 100	Course Code: CSEPC 209
Total Contact Hours				**Theory Assessment**
Theory	: 45Hrs			End Term Exam : 70
				Progressive Assessment : 30
Pre-Requisite	**: Nil**			
Credit	**: 3**			Category of Course : PC

RATIONALE:

Algorithms form the backbone of computer science and software engineering. They enable efficient data storage, retrieval, and manipulation, while algorithms provide systematic methods to solve complex computational problems. Mastering the subject of Algorithms equips students with problem-solving skills essential for developing optimized, scalable, and robust software systems.

LEARNING OUTCOMES:

After completion of the course, the students will be able to:

- Define Algorithm with its characteristics.
- Write algorithms with pseudocode.
- Implement algorithms for sorting and searching using appropriate data structures.
- Analyze the time and space complexity of algorithms
- Design solutions using advanced data structures for real-world applications, such as shortest path problems or flow-based algorithms.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Algorithms:** Definition of algorithm, Criteria of algorithms – Input/output, finiteness, definiteness, effectiveness, writing an algorithm with pseudocode, algorithms and programs	8
II	**Algorithmic Complexity:** Concept of algorithmic complexity, space complexity, time complexity, worst case, average case and best case analysis, Big-O notation, Finding the complexity of an algorithm	8
III	**Recursive algorithms:** Concept of iteration and recursion, examples of recursive algorithms – Fibonacci series, factorial, Tower-of-Hanoi problem, Complexities of recursive algorithms, conversion of recursive algorithm to iterative algorithm.	6
IV	**Algorithm Paradigms:** Greedy, Divide and Conquer, Branch and Bound, Dynamic Programming and Backtracking.	8
V	**Sorting:** The sorting problem. Bubble sort, Selection sort, Insertion sort,	8

	Mergesort, Quicksort, Heap sort, Radix sort.Searching: Symbol Tables, Binary Search Trees, Balanced Search Trees. Hashing,Hash Tables.	
VI	**Graphs:** Definition of a directed and undirected graph. Paths, Cycles, spanning trees, Directed Acyclic Graphs. Topological Sorting. Minimum Spanning Tree algorithms, Shortest Path algorithms: Dijkstra's algorithm. Flow-based algorithms.	7

REFERENCES:

1.	Narasimha Karumanchi, Data Structures And Algorithms Made Easy: Data Structures And Algorithmic Puzzles, 2nd Edition, CareerMonk Publications, India, 2011.
2.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 3rd Edition, MIT Press, Cambridge, 2009.
3.	Reema Thareja, Data Structures Using C, 2nd Edition, Oxford University Press India, New Delhi, 2014.
4.	Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, Fundamentals of Data Structure in C, 2nd Edition, University Press, India, 2008.
5.	Gajendra Sharma, Design & Analysis of Algorithms, 1st Edition, Khanna Publishing House, New Delhi, 2016.
6.	Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Pearson Education, United States, 2011.

PROGRAMMING WITH C++ Lab

L	T	P		
0	0	4		Course Code: CSEPC 211
Total Contact Hours				**Practical Assessment**
Practical		: 60Hrs	**Total Marks: 50**	End Term Exam : 15
				Progressive Assessment : 35
Pre-Requisite		**: Nil**		
Credit		**: 2**		Category of Course : PC

RATIONALE:

Programming with C++ is a foundation course for any would-be IT professional. It gives exposure to the basic techniques of computer programming in current technological scenario. It is most essential for any curriculum of Computer Science and Engineering. This course provides the student with the skill to write computer programming with C++ language.

Learning Objectives:

After completing this course, students will be able to:

- Implement object-oriented programming (OOP) concepts like encapsulation, inheritance, and polymorphism.
- Use control structures, functions, and recursion to solve programming problems efficiently.
- Work with C++ Standard Library components, including file handling and STL.
- Apply problem-solving techniques through hands-on coding exercises and small projects.
- Develop skills in writing, compiling, debugging, and executing C++ programs.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	Familiarization with C++ programming environment (Editor, Compiler, etc.)	4
II	Programs using I/O statements and various operators in C++	4
III	Programs using C++ expression evaluation and precedence	4
IV	Programs using C++ decision-making statements and branching statements	4
V	Programs using loop statements in C++	4
VI	Programs to demonstrate applications of n-dimensional arrays in C++	4
VII	Programs to demonstrate use of string manipulation functions in C++	4
VIII	Programs to demonstrate parameter passing mechanism in C++	4

IX	Programs in C++ to demonstrate recursion	4
X	Programs in C++ to demonstrate use of pointers	4
XI	Programs in C++ to demonstrate class, object, constructor, destructor.	8
XII	Programs in C++ to demonstrate function overloading,operator overloading.	6
XIII	Programs in C++ to demonstrate dynamic memory allocation,file operations	6

REFERENCES:

1.	Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education
2.	ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education
3.	C++: The Complete Reference - Schildt, McGraw-Hill Education (India)
4.	C++ and Object Oriented Programming - Jana, PHI Learning.
5.	Object Oriented Programming with C++ - Rajiv Sahay, Oxford
6.	Mastering C++ - Venugopal, McGraw-Hill Education (India)

PROGRAMMING WITH PYTHON LAB

L	T	P			
0	0	4		**Course Code: CSEPC 213**	
Total Contact Hours			**Total Marks: 50**	**Practical Assessment**	
Practical	: 60Hrs			End Term Exam	: 15
				Progressive Assessment	: 35
Pre-Requisite	**: Nil**				
Credit	**: 2**			Category of Course	: PC

RATIONALE:

Python is a versatile and beginner-friendly scripting language, renowned for its simplicity and readability. It enables rapid development of applications in fields like web development, data analysis, automation, and artificial intelligence. Learning Python equips students with a powerful toolset to solve complex problems and adapt to evolving technology demands.

LEARNING OUTCOMES:

After completion of the course, the students will be able to**:**

- Describe Python's features, applications, and fundamental concepts such as control structures, functions, and recursion.
- Develop Python programs using built-in data structures, string manipulations, file handling, and modules for practical problem-solving.
- Implement object-oriented programming concepts, including classes, objects, inheritance, and exception handling, to structure programs effectively.
- Evaluate advanced Python libraries such as NumPy, Pandas, and Matplotlib to process, analyze, and visualize data for specific applications.
- Design a mini-project to integrate Python programming concepts, demonstrating real-world applications and problem-solving capabilities.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Python:** • Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder) • Write simple Python scripts to demonstrate variable declarations • Data types, and operators • Debug Python scripts to identify and fix errors.	10
II	**Control Structures and Functions:** • Implement conditional statements (if, else, elif) in real-life scenarios	10

	• Write programs using loops (for, while, and nested loops) to solve repetitive tasks • Define custom functions, including examples of recursion, use lambda functions for inline operations.	
III	**Data Structures in Python:** • Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data • Perform string manipulations using built-in methods • Introduce Python’s collections module with practical examples.	**10**
IV	**File Handling and Modules:** • Write programs to read, write, and append text files • Work with CSV files using Python's csv module • Read and write JSON files to store structured data • Explore built-in modules like os, math, and datetime, create and import custom modules.	**10**
V	**Object-Oriented Programming (OOP) in Python:** • Define classes and create objects with attributes and methods • Implement encapsulation, inheritance, and polymorphism • Work with magic methods (e.g., __init__, __str__) and operator overloading, • Write programs to handle exceptions using try, except, and finally.	**10**
VI	**Advanced Python and Applications:** • Use NumPy for numerical operations and Pandas for data analysis • Mini-Project: Develop a Python script to solve a real-world problem (e.g., a data analysis script, a file organizer, or a basic web scraper).	**10**

REFERENCES:

1.	Yashavant Kanetkar, Let Us Python, 2nd Edition, BPB Publications, India, 2020.
2.	Reema Thareja, Python Programming: Using Problem Solving Approach, 1st Edition, Oxford University Press, India, 2017.
3.	R. Nageswara Rao, Core Python Programming, 1st Edition, Dreamtech Press, India, 2018.
4.	Satish Jain & Shashi Singh, Python Programming for Beginners, 1st Edition, BPB Publications, India, 2019.
5.	A. K. Sinha, Python Programming: A Modular Approach with Graphics, Database, Mobile and Web Applications, 1st Edition, BPB Publications, India, 2021.
6.	Yashavant Kanetkar, Let Us Python, 2nd Edition, BPB Publications, India, 2020.

DATA STRUCTURES LAB

L	T	P		
0	0	4		**Course Code: CSEPC 215**
Total Contact Hours				**Practical Assessment**
Theory	: 60Hrs		**Total Marks: 100**	End Term Exam : 15
				Progressive Assessment : 35
Pre-Requisite	**: Nil**			
Credit	**: 2**			Category of Course : PC

RATIONALE:

Data Structures is the backbone of computer science and software engineering. They enable efficient data storage, retrieval, and manipulation, while algorithms provide systematic methods to solve complex computational problems. Mastering DSA equips students with problem-solving skills essential for developing optimized, scalable, and robust software systems.

LEARNING OUTCOMES:

After completion of the course, the students will be able to

- Explain basic terminologies and operations on data structures.
- Perform asymptotic and worst-case analysis of algorithms.
- Implement linear data structures.
- Demonstrate trees and graphs to solve problems.
- Implement sorting and searching algorithms.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Data Structures:** • Write a program to analyze and compare the time complexity of basic operations (e.g., searching, insertion) on arrays and linked lists.	2
II	**Linear Data Structures:** • Implement stack operations (push, pop, peek) using arrays and linked lists • Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation) • Implement queue operations (enqueue, dequeue) using arrays and linked lists • Write programs for types of queues: circular queues and dequeue	10
III	**Linked Lists:**	10

	• Implement singly linked list operations (insertion, deletion, traversal). • Write programs to create and manipulate circular and doubly linked lists • Implement stack and queue operations using linked lists.	
IV	**Non-Linear Data Structures:** • Implement binary tree operations (insertion, deletion, traversal) • Develop programs for types of binary trees (binary search tree, AVL tree)	8
V	**Sorting:** • Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.	10
VI	**Searching:** • Write programs for searching using binary search trees (BST) and hash tables • Implement symbol table operations using balanced search trees.	10
VII	**Graphs:** • Implement graph representations (adjacency list, adjacency • matrix) and basic graph traversals (BFS, DFS).	10

REFERENCES:

1.	Seymour Lipschutz , Schaum's Outlines - Data Structures With C, Tata-MacGraw-Hill
2.	Y. Langsam, M. J. Augenstein, A. M. Tanenbaum, Data Structures Using C and C++, Prentice Hall of India.

DIGITAL ELECTRONICS LAB

L	T	P		
0	0	4		Course Code: CSEPC 217
Total Contact Hours				**Practical Assessment**
Theory		: 60Hrs	**Total Marks: 50**	End Term Exam : 15
				Progressive Assessment : 35
Pre-Requisite		**: Nil**		
Credit		**: 2**		Category of Course : PC

RATIONALE:

Digital Electronics form the foundation of modern computing, enabling the design and functioning of digital systems. They cover essential concepts like logic circuits, memory, and system architecture, bridging hardware and software integration. Acquiring skill on these topics equips students to understand and develop efficient, reliable computing systems.

LEARNING OUTCOMES:

After completion of the course, the students will be able to:

- Analyze the truth tables of all basic logic gates using CMOS and TTL logic.
- Implement Boolean expressions using logic gates.
- Design arithmetic circuits such as adders and subtractors using ICs.
- Construct sequential circuits, including flip-flops, shift registers, and counters.
- Develop digital circuits with a 7-segment display for counting applications.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	To verify the truth tables for all logic gates: NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.	6
II	Implement and realize Boolean Expressions with Logic Gates.	6
III	Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using ICs	6
IV	Implement parallel and serial full-adder using ICs.	6
V	Design and development of multiplexer and De-multiplexer using multiplexer ICs	6
VI	Verification of the function of SR, D, JK, and T Flip Flops.	8
VII	Design controlled shift registers.	6
VIII	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.	8
IX	To design a programmable Up-Down Counter with a 7-segment display.	8

REFERENCES:

1.	Digital Principles & Applications by Albert Paul Malvino & Donald P. Leach, McGraw Hill Education; Eighth edition, ISBN: 978-9339203405.
2.	Digital Electronics by Roger L. Tokheim, McGraw-Hill Education (ISE Editions); International 2 Revised ed edition, ISBN: 978-0071167963.
3.	Digital Electronics – an Introduction to Theory and Practice by William H. Gothmann, Prentice Hall India Learning Private Limited; 2 edition, ISBN: 978-8120303485.
4.	Fundamentals of Logic Design by Charles H. Roth Jr., Jaico Publishing House; First edition, ISBN: 978-8172247744.
5.	Digital Electronics by R. Anand, Khanna Publications, New Delhi (Edition 2018), ISBN: 978-93-82609445.

SUMMER INTERNSHIP - I

L	T	P		
0	0	0		**Course Code: SI 201**
Total Contact Hours				
Practical	: 30Hrs		**Total Marks: 50**	End Term Exam : 15
				Progressive Assessment : 35
Pre-Requisite	**: Nil**			
Credit	**: 2**			Category of Course : SI

Duration: 3-4 weeks during summer vacation after 2nd Semester.

RATIONALE:

Summer Internship - I is to offer a structured and practical learning experience that prepares individuals for their future careers, helps them make informed career choices, and equips them with the skills and knowledge necessary to succeed in their chosen field. This course provides opportunities to students for hands-on industry experience.

LEARNING OUTCOMES:

After completion of the course, the students will be able to:

- Apply theoretical knowledge gained in their academic coursework to real-world
- situations.
- Develop and refine specific skills relevant to the field.
- Gains hands-on experience in a professional network by interacting with mentors
- and industry professionals.
- Learn to manage their time effectively.
- Clarify career goals.

DETAILED COURSE CONTENTS

SUGGESTED ACTIVITIES:

I Orientation: • Introduction to the organization’s mission, values, and culture. • Familiarization with workplace policies, procedures, and safety guidelines. • Orientation to the team and organizational structure.
II Project-Based Learning: • Description of the main project or tasks the intern will be working on during the internship. • Detailed project goals and objectives. • Training and guidance on project-specific tools, technologies, or • methodologies.
III Technical and Skill Development: • Training sessions or workshops to enhance technical skills relevant to the internship role

(e.g., programming languages, software tools, laboratory techniques). • Soft skills development, including communication, teamwork, problem solving, and time management
IV Mentorship and Supervision: • Regular meetings with a designated mentor or supervisor for guidance, feedback, and support. • Mentorship objectives and expectations.
V Professional Development: • Sessions on professional etiquette, networking, and building a personal brand • Resume writing and interview preparation workshops.
VI Industry and Field-Specific Knowledge: • Lectures, seminars, or presentations on industry trends, best practices, and emerging technologies. • Guest speakers from the field to share insights and experiences.
VII Reporting and Documentation: • Training on how to document project progress, results, and findings. • Practice in creating reports, presentations, or other deliverables.
VIII Ethics and Professionalism: • Discussions on ethical considerations within the field. • Scenarios and case studies related to ethical decision-making
IX Feedback and Evaluation: • Regular performance evaluations and feedback sessions. • Self-assessment and goal-setting exercises.
X Networking and Industry Exposure: • Opportunities to attend industry conferences, webinars, or networking events. • Encouragement to connect with professionals in the field.

NOTE

As per AICTE guidelines, in Summer Internship-I, students are required to be involved in Inter/ Intra Institutional Activities viz;

- Training with higher Institutions;
- Soft skill training organized by Training and Placement Cell of the respective institutions;
- contribution at incubation/ innovation /entrepreneurship cell of the institute;
- participation in conferences/ workshops/ competitions etc.;
- Learning at Departmental Lab/ Tinkering Lab/ Institutional workshop;
- Working for consultancy/ research project within the institutes and
- Participation in all the activities of Institute’s Innovation Council for eg: IPR workshop/Leadership Talks/ Idea/ Design/ Innovation/ Business Completion/ Technical Expos etc.