Lesson Plan:- CSEPC201

TH:1- PROGRAMMING WITH C++

3rd Semester CSE- Winter 2026

Name of the Faculty- Pratibha Patnaik **Date-8th July 2026**

Unit I – Introduction to C++ and Basic Programming (12 Classes)	
1	Introduction to Object-Oriented Programming: Features and Principles
2	User-defined Types: Structures and Unions
3	Concepts: Polymorphism and Encapsulation
4	Getting Started with C++: Syntax and Basic Program
5	Data Types, Variables, and Operators in C++
6	Control Structures: if, else, switch, loops
7	Functions: Syntax, Parameters, and Return Types
8	Functions with Default Arguments, Recursion
9	*Namespaces and Scope Resolution*
10	Strings and Character Arrays
11	One-dimensional and Two-dimensional Arrays
12	Pointers and Pointer Arithmetic
Unit II – Abstraction & Inheritance (11 Classes)	
13	Classes and Objects: Definitions, Access Specifiers
14	Constructors: Default, Parameterized
15	Destructors and Object Initialization
16	Member Functions, Inline Functions
17	Friend Functions and Static Members
18	References and Their Use
19	Inheritance: Basics and Types
20	Single and Multilevel Inheritance
21	Multiple and Hybrid Inheritance
22	Virtual Base Class and Constructor/Destructor Call Order

23	Derived Class Constructors and Base Initialization
Unit III – Polymorphism (9 Classes)	
24	Introduction to Polymorphism: Static vs Dynamic Binding
25	Function Overloading and Ambiguities
26	Base Class Pointer and Object Slicing
27	Late Binding and Method Overriding
28	Virtual Functions and Runtime Polymorphism
29	Pure Virtual Functions and Abstract Classes
30	Abstract Classes: Design and Usage
31	Polymorphism Examples and Applications
32	Recap and Quick Quiz on Polymorphism
Unit IV – *Operator Overloading (9 Classes)*	
33	Operator Functions: Syntax and Usage
34	This Pointer and Its Applications
35	Overloading Unary and Binary Operators
36	Member vs Non-member Operator Overloads
37	Overloading Assignment and Comparison Operators
38	Overloading I/O Operators (<<, >>)
39	Examples and Programs on Operator Overloading
40	Common Mistakes in Operator Overloading
41	*Practice Exercises and Review*
Unit V – Exception Handling (4 Classes)	
42	Basics: try, throw, catch
43	Exceptions with Classes and Inheritance
44	Function Exception Specification, unexpected()
45	Final Review + Unit V Exercises / MCQ Test

Pratibha Patnaik 8/7/26
Signature of the Faculty

8/7/26
HOD, CSE

8/7/26
Academic Coordinator

Lesson Plan:- CSEPC 203
TH:2- PROGRAMMING WITH PYTHON

3rd Semester CSE- Winter 2026

Name of the Faculty- Prabeen Kumar Pattnaik **Date-1st July 2026**

Unit I – Introduction to Python (8 Classes)	
1	**Introduction to Python:** Overview of Python: Features and Applications
2	Setting Up the Python Environment
3	Python Installation, IDEs
4	Python Installation, IDEs
5	Python Syntax: Variables, Data Types
6	Operators Writing, Executing
7	Operators Writing, Executing
8	Debugging Python Scripts
Unit II – Control Structures and Functions (8 Classes)	
9	**Control Structures and Functions**
10	Conditional Statements: if, else, elif
11	Loops: for, while and Nested Loops
12	Programs on Loops
13	Functions: Defining, Calling, and Scope of Variables
14	Functions: Defining, Calling, and Scope of Variables
15	Introduction to Lambda Functions
16	Recursion
Unit III – Data Structures in Python: (8 Classes)	
17	Lists, Tuples, Sets
18	Dictionaries: Operations and Applications
19	Dictionaries: Operations and Applications
20	List Comprehensions and Dictionary Comprehensions
21	Working with Strings: Methods and Manipulation
22	Working with Strings: Methods and Manipulation
23	Introduction to Python's Collections Module

24	Introduction to Python's Collections Module
Unit IV – File Handling and Modules: (7 Classes)	
25	File Operations, Reading, Writing
26	File Operations, Reading, Writing
27	Appending Files Working with CSV and JSON Files
28	Appending Files Working with CSV and JSON Files
29	Python Modules: Built-In Modules (e.g., math, os, datetime)
30	Creating and Using Custom Modules
31	Creating and Using Custom Modules
Unit V – Object-Oriented Programming (OOP) in Python: (7 Classes)	
32	Understanding Classes and Objects
33	Concepts of Encapsulation, Inheritance
34	Inheritance, Concepts of Polymorphism,
35	Programming on Inheritance and Polymorphism
36	Working with Magic Methods
37	Operator Overloading
38	Exception Handling in Python
Unit VI – Advanced Python and Applications: (7 Classes)	
39	Introduction to Libraries: NumPy, Pandas
40	Matplotlib Basics of Web Scraping: Using requests and Beautiful Soup
41	Matplotlib Basics of Web Scraping: Using requests and Beautiful Soup
42	Scripting for Automation: Working with os and shutil Modules
43	Scripting for Automation: Working with os and shutil Modules
44	Mini-Project: Developing a Python Script for a Real-World Problem
45	Mini-Project: Developing a Python Script for a Real-World Problem

Signature of the Faculty

HOD, CSE

Academic Coordinator

Lesson Plan: - TH: 4- DIGITAL ELECTRONICS AND COMPUTER ORGANISATION

3rd Semester CSE- Winter 2026

Name of the Faculty- Soma Das **Date-1st July 2026**

Unit I – Introduction to Digital Electronics: (8 Classes)	
1	Difference Between Analog and Digital Signals
2	Number Systems: Binary, Octal, Decimal, and Hexadecimal
3	Number Systems: Binary, Octal, Decimal, and Hexadecimal
4	Conversion Between Number Systems
5	Conversion Between Number Systems
6	Binary Arithmetic Boolean Algebra: Basic Operations, Laws
7	Simplification
8	Simplification
Unit II – Logic Gates and Circuits: (7 Classes)	
9	***Logic Gates:****AND, OR, NOT, NAND, NOR, XOR, XNOR*
10	Design and Simplification of Logic Circuits Using Boolean Algebra,
11	Design and Simplification of Logic Circuits Using Boolean Algebra,
12	Karnaugh Maps (K-Maps) for Simplification
13	Karnaugh Maps (K-Maps) for Simplification
14	Karnaugh Maps (K-Maps) for Simplification
15	Practical Applications of Logic Gates in Real-World Circuits
Unit III – Linked Lists: Combinational and Sequential Circuits: (8 Classes)	
16	Combinational and Sequential Circuits
17	Combinational Circuits: Multiplexers, Demultiplexers, Encoders, and Decoders
18	Combinational Circuits: Multiplexers, Demultiplexers, Encoders, and Decoders
19	Sequential Circuits: Flip-Flops (SR, JK, D, T)
20	Sequential Circuits: Flip-Flops (SR, JK, D, T)
21	Counters: Synchronous and Asynchronous Counters
22	Registers and Shift Registers: Types and Uses
23	Registers and Shift Registers: Types and Uses

Unit IV – Fundamentals of Computer Organization: (8 Classes)	
24	Basic Structure of a Computer: CPU, Memory, Input/Output Devices
25	Basic Structure of a Computer: CPU, Memory, Input/Output Devices
26	Instruction Cycle: Fetch, Decode, Execute
27	Instruction Cycle: Fetch, Decode, Execute
28	Memory Organization: Types of Memory (RAM, ROM)
29	Cache, Virtual Memory
30	Cache, Virtual Memory
31	Introduction to Buses: Address Bus, Data Bus, and Control Bus
Unit V – Processor Architecture and Control: (7 Classes)	
32	Unit Introduction to Microprocessors and Microcontrollers
33	Basics of Arithmetic Logic Unit (ALU) and Control Unit
34	Basics of Arithmetic Logic Unit (ALU) and Control Unit
35	Instruction Set Architecture (ISA)
36	RISC vs CISC Pipelining
37	RISC vs CISC Pipelining
38	Performance Optimization in Processors
Unit VI – Input/Output Systems and Advanced Topics: (7 Classes)	
39	I/O Devices and Interfaces: Keyboard, Mouse, Printers, and Storage Devices
40	I/O Devices and Interfaces: Keyboard, Mouse, Printers, and Storage Devices
41	Interrupts and DMA (Direct Memory Access)
42	Overview of Modern Trends: Multicore Processors, GPUs, and Embedded Systems
43	Overview of Modern Trends: Multicore Processors, GPUs, and Embedded Systems
44	Mini- Project: Design a Simple Digital Circuit or Simulate a Basic CPU Operation
45	Mini- Project: Design a Simple Digital Circuit or Simulate a Basic CPU Operation

Signature of the Faculty

HOD, CSE

Academic Coordinator

Lesson Plan – Algorithms (CSEPC 209/TH5)

3rd Semester Computer Science (Winter 2026)

Name of the Faculty-Pratibha Patnaik **Date-8th July 2026**

Class No.	Topic
Unit I – Introduction to Algorithms (8 Classes)	
1	What is an Algorithm? Characteristics and Importance
2	Algorithm vs Program, Algorithm Writing Basics
3	Input/Output, Finiteness, Definiteness, Effectiveness
4	Writing Pseudocode: Format and Conventions
5	Examples: Simple Algorithms (Sum, Max-Min)
6	More Examples: Swapping, Factorial (Non-Recursive)
7	Dry Run and Step-by-Step Execution of Algorithms
8	Practice Exercises and Class Quiz
Unit II – Algorithmic Complexity (8 Classes)	
9	Concept of Time and Space Complexity
10	Best, Worst, and Average Case Analysis
11	Big-O, Big-Ω, Big-Θ Notations
12	Rules to Compute Time Complexity
13	Examples of Complexity Calculation
14	Analyzing Simple Search and Loop-based Algorithms
15	Exercises and Complexity Comparisons
16	Quiz + Discussion on Complexity Problems
Unit III – Recursive Algorithms (6 Classes)	
17	Concept of Recursion vs Iteration
18	Recursive Examples: Fibonacci, Factorial
19	Tower of Hanoi and Analysis
20	Complexity of Recursive Algorithms
21	Converting Recursion to Iteration
22	Practice Problems and Recap

Class No.	Topic
Unit IV – Algorithm Paradigms (8 Classes)	
23	Greedy Approach – Concept and Introduction
24	Examples: Coin Change, Activity Selection
25	Divide and Conquer – Concept & Merge Sort Example
26	Quicksort as Divide and Conquer
27	Dynamic Programming – Concept & Fibonacci/Knapsack
28	Backtracking – N-Queens/Subset Sum
29	Branch and Bound – Conceptual Overview
30	Revision and Comparison of Paradigms
Unit V – Sorting & Searching (8 Classes)	
31	Sorting Overview: Importance and Classification
32	Bubble Sort, Selection Sort
33	Insertion Sort, Comparison with Others
34	Merge Sort and Quicksort (Recap)
35	Heap Sort, Radix Sort Overview
36	Searching Techniques: Linear and Binary Search
37	Binary Search Tree and Balanced BST
38	Hashing and Hash Tables
Unit VI – Graph Algorithms (7 Classes)	
39	Introduction to Graphs: Directed/Undirected, Cycles
40	Graph Representations and Traversals
41	Topological Sorting
42	Minimum Spanning Tree: Kruskal's and Prim's Algorithm
43	Shortest Path: Dijkstra's Algorithm
44	Flow-Based Algorithms: Concept Overview
45	Final Review + Class Quiz

Pratibha Patnaik 8/7/26
Signature of the Faculty

8/7/26
HOD, CSE

ACADEMIC CO-ORDINATOR

Lesson Plan: - CSEPC 211
PR-1:- Programming with C++ Lab
3rd Semester CSE- Winter 2026

Name of the Faculty- Pratibha Patnaik **Date-8th July 2026**

Unit I (4 Classes)	
1	Familiarization with C++ programming environment (Editor, Compiler, etc.)
2	Familiarization with C++ programming environment (Editor, Compiler, etc.)
3	Familiarization with C++ programming environment (Editor, Compiler, etc.)
4	Familiarization with C++ programming environment (Editor, Compiler, etc.)
Unit II (4 Classes)	
5	Programs using, I/O statements and various operators in C++
6	Programs using, I/O statements and various operators in C++
7	Programs using, I/O statements and various operators in C++
8	Programs using, I/O statements and various operators in C++
Unit III (4 Classes)	
9	Programs using C++ expression evaluation and precedence
10	Programs using C++ expression evaluation and precedence
11	Programs using C++ expression evaluation and precedence
12	Programs using C++ expression evaluation and precedence
Unit IV (4 Classes)	
13	Programs using C++ decision-making statements and branching statements
14	Programs using C++ decision-making statements and branching statements
15	Programs using C++ decision-making statements and branching statements
16	Programs using C++ decision-making statements and branching statements
Unit V (4 Classes)	
17	Programs using loop statements in C++
18	Programs using loop statements in C++
19	Programs using loop statements in C++
20	Programs using loop statements in C++
Unit VI (4 Classes)	
21	Programs to demonstrate applications of n-dimensional arrays in C++
22	Programs to demonstrate applications of n-dimensional arrays in C++

23	Programs to demonstrate applications of n-dimensional arrays in C++
24	Programs to demonstrate applications of n-dimensional arrays in C++
Unit VII (4 Classes)	
25	Programs to demonstrate use of string manipulation functions in C++
26	Programs to demonstrate use of string manipulation functions in C++
27	Programs to demonstrate use of string manipulation functions in C++
28	Programs to demonstrate use of string manipulation functions in C++
Unit VIII (4 Classes)	
29	Programs to demonstrate parameter passing mechanism in C++
30	Programs to demonstrate parameter passing mechanism in C++
31	Programs to demonstrate parameter passing mechanism in C++
32	Programs to demonstrate parameter passing mechanism in C++
Unit IX (4 Classes)	
33	Programs in C++ to demonstrate recursion
34	Programs in C++ to demonstrate recursion
35	Programs in C++ to demonstrate recursion
36	Programs in C++ to demonstrate recursion
Unit X (4 Classes)	
37	Programs in C++ to demonstrate use of pointers
38	Programs in C++ to demonstrate use of pointers
39	Programs in C++ to demonstrate use of pointers
40	Programs in C++ to demonstrate use of pointers
Unit XI (8 Classes)	
41	Programs in C++ to demonstrate class, object, constructor, destructor.
42	Programs in C++ to demonstrate class, object, constructor, destructor.
43	Programs in C++ to demonstrate class, object, constructor, destructor.
44	Programs in C++ to demonstrate class, object, constructor, destructor.
45	Programs in C++ to demonstrate class, object, constructor, destructor.
46	Programs in C++ to demonstrate class, object, constructor, destructor.
47	Programs in C++ to demonstrate class, object, constructor, destructor.
48	Programs in C++ to demonstrate class, object, constructor, destructor.
Unit XII (6 Classes)	
49	Programs in C++ to demonstrate function overloading, operator overloading.
50	Programs in C++ to demonstrate function overloading, operator overloading.

51	Programs in C++ to demonstrate function overloading, operator overloading.
52	Programs in C++ to demonstrate function overloading, operator overloading.
53	Programs in C++ to demonstrate function overloading, operator overloading.
54	Programs in C++ to demonstrate function overloading, operator overloading.
Unit XIII (6 Classes)	
55	Programs in C++ to demonstrate dynamic memory allocation, file operations
56	Programs in C++ to demonstrate dynamic memory allocation, file operations
57	Programs in C++ to demonstrate dynamic memory allocation, file operations
58	Programs in C++ to demonstrate dynamic memory allocation, file operations
59	Programs in C++ to demonstrate dynamic memory allocation, file operations
60	Programs in C++ to demonstrate dynamic memory allocation, file operations

Prathibha Pothasi
8/7/26
Signature of the Faculty

8/7/26
HOD, CSE

Academic Coordinator

Lesson Plan: - CSEPC 213

PR:2- PROGRAMMING WITH PYTHON LAB

3rd Semester CSE- Winter 2026

Name of the Faculty- Prabeen Kumar Pattnaik **Date-1st July 2026**

Unit I – Introduction to python: Data types, and operators• **(10 Classes)**	
1	Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder)
2	Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder)
3	Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder)
4	Write simple Python scripts to demonstrate variable declarations
5	Write simple Python scripts to demonstrate variable declarations
6	Data types, and operators
7	Data types, and operators
8	Data types, and operators
9	Debug Python scripts to identify and fix errors.
10	Debug Python scripts to identify and fix errors.
Unit II – Control Structures and function (10 Classes)	
11	Implement conditional statements (if, else, elif) in real-life scenarios
12	Implement conditional statements (if, else, elif) in real-life scenarios
13	Implement conditional statements (if, else, elif) in real-life scenarios
14	Write programs using loops (for, while, and nested loops) to solve repetitive tasks
15	Write programs using loops (for, while, and nested loops) to solve repetitive tasks
16	Write programs using loops (for, while, and nested loops) to solve repetitive tasks
17	Write programs using loops (for, while, and nested loops) to solve repetitive tasks
18	Define custom functions, including examples of recursion, use lambda functions for inline operations.
19	Define custom functions, including examples of recursion, use lambda functions for inline operations.
20	Define custom functions, including examples of recursion, use lambda functions for inline operations.

Unit III –Data structures in python: (10 Classes)	
21	Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data
22	Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data
23	Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data
24	Perform string manipulations using built-in methods
25	Perform string manipulations using built-in methods
26	Perform string manipulations using built-in methods
27	Introduce Python's collections module with practical examples.
28	Introduce Python's collections module with practical examples.
29	Introduce Python's collections module with practical examples.
30	Introduce Python's collections module with practical examples
Unit IV – File Handing and modules: (10 Classes)	
31	Write programs to read, write, and append text file
32	Write programs to read, write, and append text file
33	Write programs to read, write, and append text file
34	Work with CSV files using Python's csv module
35	Work with CSV files using Python's csv module
36	Read and write JSON files to store structured data
37	Read and write JSON files to store structured data
38	Explore built-in modules like os, math, and datetime, create and import custom modules.
39	Explore built-in modules like os, math, and datetime, create and import custom modules.
40	Explore built-in modules like os, math, and datetime, create and import custom modules.
Unit V –Object-Oriented programming(OOP): (10 Classes)	
41	Define classes and create objects with attributes and methods
42	Define classes and create objects with attributes and methods
43	Define classes and create objects with attributes and methods
44	Implement encapsulation, inheritance, and polymorphism
45	Implement encapsulation, inheritance, and polymorphism
46	Work with magic methods (e.g., init , str) and operator overloading,
47	Work with magic methods (e.g., init , str) and operator overloading,

48	Work with magic methods (e.g., init , str) and operator overloading.
49	Write programs to handle exceptions using try, except, and finally.
50	Write programs to handle exceptions using try, except, and finally.
Unit VI – Advanced python and Application (10 Classes)	
51	Use NumPy for numerical operations and Pandas for data analysis
52	Use NumPy for numerical operations and Pandas for data analysis
53	Use NumPy for numerical operations and Pandas for data analysis
54	Use NumPy for numerical operations and Pandas for data analysis
55	Use NumPy for numerical operations and Pandas for data analysis
56	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).
57	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).
58	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).
59	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).
60	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).

Signature of the Faculty

HOD, CSE

Academic Coordinator

Lesson Plan: - CSEPC 215

PR:3- DATA STRUCTURES AND ALGORITHMS LAB

3rd Semester CSE- Winter 2026

Name of the Faculty- Pratibha Patnaik **Date-** 8th July 2026

Unit I – Introduction to data structure :(2 Classes)	
1	Write a program to analyze and compare the time complexity of basic operations (e.g., searching, insertion) on arrays and linked lists.
2	Write a program to analyze and compare the time complexity of basic operations (e.g., searching, insertion) on arrays and linked lists.
Unit II – Linear Data Structures: (12 Classes)	
3	Implement stack operations (push, pop, peek) using arrays and linked lists
4	Implement stack operations (push, pop, peek) using arrays and linked lists
5	Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation)
6	Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation)
7	Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation)
8	Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation)
9	Implement queue operations (enqueue, dequeue) using arrays and linked *listsImplement queue operations (enqueue, dequeue) using arrays and linked lists*
10	Implement queue operations (enqueue, dequeue) using arrays and linked lists
11	Implement queue operations (enqueue, dequeue) using arrays and linked lists
12	Write programs for types of queues: circular queues and dequeue
13	Write programs for types of queues: circular queues and dequeue
14	Write programs for types of queues: circular queues and dequeue
Unit III – Linked Lists: (10 Classes)	
15	Implement singly linked list operations (insertion, deletion, traversal).
16	Implement singly linked list operations (insertion, deletion, traversal).
17	Implement singly linked list operations (insertion, deletion, traversal).
18	Write programs to create and manipulate circular and doubly linked lists
19	Write programs to create and manipulate circular and doubly linked lists

20	Write programs to create and manipulate circular and doubly linked lists
21	Write programs to create and manipulate circular and doubly linked lists
22	Implement stack and queue operations using linked lists.
23	Implement stack and queue operations using linked lists.
24	Implement stack and queue operations using linked lists.
Unit IV – Non-Linear Data Structure: (16 Classes)	
25	Implement binary tree operations (insertion, deletion, traversal)
26	Implement binary tree operations (insertion, deletion, traversal)
27	Implement binary tree operations (insertion, deletion, traversal)
28	Implement binary tree operations (insertion, deletion, traversal)
29	Develop programs for types of binary trees (binary search tree, AVL tree)
30	Develop programs for types of binary trees (binary search tree, AVL tree)
31	Develop programs for types of binary trees (binary search tree, AVL tree)
32	Develop programs for types of binary trees (binary search tree, AVL tree)
33	Implement graph representations (adjacency list, adjacency matrix
34	Implement graph representations (adjacency list, adjacency matrix
35	Implement graph representations (adjacency list, adjacency matrix
36	Implement graph representations (adjacency list, adjacency matrix
37	Implement graph representations (adjacency list, adjacency matrix)
38	Basic graph traversals (BFS, DFS)
39	Basic graph traversals (BFS, DFS)
40	Basic graph traversals (BFS, DFS)
Unit V – Sorting: (10 Classes)	
41	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
42	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
43	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
44	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
45	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
46	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
47	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge

	sort, quicksort.
48	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
49	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
50	Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.
Unit VI – Searching : **(10 Classes)**	
51	Write programs for searching using binary search trees (BST) and hash tables
52	Write programs for searching using binary search trees (BST) and hash tables
53	Write programs for searching using binary search trees (BST) and hash tables
54	Write programs for searching using binary search trees (BST) and hash tables
55	Write programs for searching using binary search trees (BST) and hash tables
56	Implement symbol table operations using balanced search trees.
57	Implement symbol table operations using balanced search trees.
58	Implement symbol table operations using balanced search trees.
59	Implement symbol table operations using balanced search trees.
60	Implement symbol table operations using balanced search trees.

Prediction & Petrait 8/7/22
Signature of the Faculty

6/7/22
HOD, CSE

Academic Coordinator

Lesson Plan: - CSEPC 211

PR:4 DIGITAL ELECTRONICS LAB

3rd Semester CSE- Winter 2026

Name of the Faculty- Soma Dash **Date- 1st July 2026**

Unit I – (6 Classes)	
1	To verify the truth tables for all logic gates: NOT, OR, AND, NAND using CMOS Logic gates and TTL Logic Gates.
2	To verify the truth tables for all logic gates: NOT, OR, AND, NAND using CMOS Logic gates and TTL Logic Gates.
3	To verify the truth tables for all logic gates: NOT, OR, AND, NAND using CMOS Logic gates and TTL Logic Gates.
4	To verify the truth tables for all logic gates: NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
5	To verify the truth tables for all logic gates: NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
6	To verify the truth tables for all logic gates: NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
Unit II –: (6 Classes)	
7	Implement and realize Boolean Expressions with Logic Gates.
8	Implement and realize Boolean Expressions with Logic Gates.
9	Implement and realize Boolean Expressions with Logic Gates.
10	Implement and realize Boolean Expressions with Logic Gates.
11	Implement and realize Boolean Expressions with Logic Gates.
12	Implement and realize Boolean Expressions with Logic Gates.
Unit III – (10 Classes)	
13	Implement Half Adder, Full Adder using ICs
14	Implement Half Adder, Full Adder using ICs
15	Implement Half Adder, Full Adder using ICs
16	Implement Half Subtractor, Full Subtractor using ICs
17	Implement Half Subtractor, Full Subtractor using ICs
18	Implement Half Subtractor, Full Subtractor using ICs
Unit IV –: (6 Classes)	
19	Implement parallel using ICs.

20	Implement parallel using ICs.
21	Implement parallel using ICs.
22	serial full-adder using ICs.
23	serial full-adder using ICs.
24	serial full-adder using ICs.
Unit V – (10 Classes)	
25	Design and development of multiplexer using multiplexer ICs
26	Design and development of multiplexer using multiplexer ICs
27	Design and development of multiplexer using multiplexer ICs
28	Design and development of De-multiplexer using multiplexer ICs
29	Design and development of De-multiplexer using multiplexer ICs
30	Design and development of De-multiplexer using multiplexer ICs
Unit VI – (8 Classes)	
31	Verification of the function of SR, D, JK, and T Flip Flops.
32	Verification of the function of SR, D, JK, and T Flip Flops.
33	Verification of the function of SR, D, JK, and T Flip Flops.
34	Verification of the function of SR, D, JK, and T Flip Flops.
35	Verification of the function of SR, D, JK, and T Flip Flops.
36	Verification of the function of SR, D, JK, and T Flip Flops.
37	Verification of the function of SR, D, JK, and T Flip Flops.
38	Verification of the function of SR, D, JK, and T Flip Flops.
Unit VII-:(6 Classes)	
39	Design controlled shift registers.
40	Design controlled shift registers.
41	Design controlled shift registers.
42	Design controlled shift registers.
43	Design controlled shift registers.
44	Design controlled shift registers.
Unit VIII-:(8 Classes)	
45	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
46	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
47	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
48	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
49	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.

50	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
51	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
52	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
Unit IX-:(8 Classes)	
53	To design a programmable Up-Down Counter with a 7-segment display.
54	To design a programmable Up-Down Counter with a 7-segment display.
55	To design a programmable Up-Down Counter with a 7-segment display.
56	To design a programmable Up-Down Counter with a 7-segment display.
57	To design a programmable Up-Down Counter with a 7-segment display.
58	To design a programmable Up-Down Counter with a 7-segment display.
59	To design a programmable Up-Down Counter with a 7-segment display.
60	To design a programmable Up-Down Counter with a 7-segment display.

Signature of the Faculty

HOD, CSE

Academic Coordinator