



Lesson plan for CSEPC 201/TH1/Programming with C++
Academic Session 2026-27 (W)
Department of Computer Science & Engineering
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SRI KAUSHIK MURMU
SUBJECT:: CSEPC 201/TH1 Programming with C++	NO.OF DAYS/PER WEEK CLASS ALLOTTED:3	SEMESTER FROM DATE:01/07/2026TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1 ST	1 ST	Introduction to object-oriented programming: user defined types, structures, unions, polymorphism, encapsulation.
	2 ND	Getting started with C++ syntax, datatype, variables
	3 RD	strings
2 ND	1 ST	functions, default values in functions
	2 ND	recursion
	3 RD	recursion, namespaces
3 RD	1 ST	operators
	2 ND	flow control
	3 RD	arrays
4 TH	1 ST	arrays
	2 ND	pointers
	3 RD	pointers
5 TH	1 ST	Abstraction mechanism: Classes, private, public
	2 ND	constructors, destructors
	3 RD	member data, member functions
6 TH	1 ST	inline function, friend functions
	2 ND	static members, and references
	3 RD	Inheritance: Class hierarchy, derived classes
7 TH	1 ST	single inheritance
	2 ND	multiple, multilevel
	3 RD	hybrid inheritance, role of virtual base class constructor and destructor execution
8 TH	1 ST	1ST INTERNAL ASSESSMENT
	2 ND	base initialization using derived class constructors
	3 RD	Polymorphism: Binding
9 TH	1 ST	Static binding
	2 ND	Dynamic binding
	3 RD	Static polymorphism
10 TH	1 ST	Function Overloading
	2 ND	Ambiguity in function overloading
	3 RD	Dynamic polymorphism: Base class pointer
11 TH	1 ST	object slicing, late binding, method overriding with virtual functions
	2 ND	pure virtual functions, abstract classes
	3 RD	Operator Overloading: This pointer
12 TH	1 ST	applications of this pointer
	2 ND	applications of this pointer
	3 RD	Operator function
13 TH	1 ST	member and non-member operator function
	2 ND	2nd INTERNAL ASSESSMENT
	3 RD	operator overloading
14 TH	1 ST	I/O operators
	2 ND	I/O operators
	3 RD	Exception handling: Try, throw, and catch

15 th	1 ST	exceptions and derived classes
	2 ND	function exception declaration
	3 RD	unexpected exceptions



Lesson plan for :: CSEPC 201/TH2/PROGRAMMING WITH PYTHON
Academic Session 2026-27 (W)
Department of Computer Science & Engineering,
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SRI ABHIRAM BEHERA
SUBJECT:: CSEPC 201/TH2 PROGRAMMING WITH PYTHON	NO.OF DAYS/PER WEEK CLASS ALLOTTED:3	SEMESTER FROM DATE:01/07/2026TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1ST	1ST	Overview of Python: Features and Applications
	2ND	Setting Up the Python Environment (Python I installation, IDEs)
	3RD	Python Syntax: Variables, Data Types
2ND	1ST	Operators Writing
	2ND	Executing
	3RD	Debugging
3RD	1ST	Python Scripts
	2ND	Python Scripts
	3RD	Control Structures and Functions: Conditional Statements: if, else,
4TH	1ST	elif Loops: for
	2ND	while, and Nested Loops
	3RD	Functions: Defining
5TH	1ST	Calling, and Scope of Variables
	2ND	Introduction to Lambda Functions
	3RD	Recursion
6TH	1ST	Recursion
	2ND	Data Structures in Python: Lists, Tuples, Sets
	3RD	Tuples, Sets
7TH	1ST	Dictionaries: Operations and Applications
	2ND	List Comprehensions and Dictionary Comprehensions
	3RD	List Comprehensions and Dictionary Comprehensions
8TH	1ST	1ST INTERNAL ASSESSMENT
	2ND	Working with Strings: Methods and Manipulation
	3RD	Introduction to Python's Collections Module
9TH	1ST	File Handling and Modules: File Operations
	2ND	Reading, Writing, and Appending Files
	3RD	Reading, Writing, and Appending Files
10TH	1ST	Working with CSV and JSON Files
	2ND	Python Modules: Built-In Modulese.g., math, os, datetime
	3RD	Creating AND Using Custom Modules
11TH	1ST	Using Custom Modules
	2ND	Object-Oriented Programming (OOP) in Python: Understanding Classes and Objects
	3RD	Concepts of Encapsulation, Inheritance, and Polymorphism
12TH	1ST	Concepts of Encapsulation, Inheritance,and Polymorphism
	2ND	Concepts of Encapsulation, Inheritance, and Polymorphism
	3RD	Working with Magic Methods and Operator Overloading

13 TH	1 ST	2nd INTERNAL ASSESSMENT
	2 ND	Exception Handling in Python
	3 RD	Advanced Python and Applications: Introduction to Libraries: NumPy
14 TH	1 ST	Introduction to Libraries: NumPy , Pandas, Matplotlib Basics of Web Scraping:
	2 ND	Using requests and BeautifulSoup, Scripting for Automation
	3 RD	Working with os and shutil Modules
15 th	1 ST	Mini-Project: Developing a Python Script for a Real-World Problem
	2 ND	Mini-Project: Developing a Python Script for a Real-World Problem
	3 RD	Mini-Project: Developing a Python Script for a Real-World Problem



Lesson plan for :: CSEPC 205/TH3/DATA STRUCTURES
Academic Session 2026-27 (W)
Department of Computer Science & Engineering,
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SMT SUMITRA MAHAPATRA
SUBJECT:: CSEPC 205/TH:3/ DATA STRUCTURES	NO.OF DAYS/PER WEEK CLASS ALLOTTED:3	SEMESTER FROM DATE:14/07/2026TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1ST	1ST	Introduction to Data StructuresBasic Terminology
	2ND	Introduction to Data Structures: Basic Terminology
	3RD	Classification of Data Structure
2ND	1ST	Classification of Data Structure
	2ND	Operations on Data Structure, Asymptotic
	3RD	Operations on Data Structure, Asymptotic
3RD	1ST	worst-case analysis of algorithms
	2ND	worst-case analysis of algorithms
	3RD	Linear Data Structures: Stacks-Introduction to Stacks
4TH	1ST	Array Representation of Stacks
	2ND	Operations on a Stack
	3RD	Applications of Stacks Infix-to-Postfix Transformation
5TH	1ST	evaluating Postfix Expressions
	2ND	Queues: Introduction to Queues
	3RD	Array Representation of Queues
6TH	1ST	Operations on a Queue
	2ND	Types of Queues-DeQueue, Circular Queue,
	3RD	Applications of Queues-Round Robin Algorithm.
7TH	1ST	Linked Lists: Singly Linked List,
	2ND	Linked Lists Representation in Memory
	3RD	Linked Lists Representation in Memory
8TH	1ST	1st INTERNAL ASSESSMENT
	2ND	Operations on a Single Linked List
	3RD	Circular Linked Lists
9TH	1ST	Circular Linked Lists
	2ND	Doubly Linked Lists
	3RD	Doubly Linked Lists
10TH	1ST	Doubly Linked Lists, Linked List Representation
	2ND	Doubly Linked Lists, Linked List Representation
	3RD	Operations of Stack, Linked List Representation
11TH	1ST	Operations of Stack, Linked List Representation
	2ND	Operations of Queue.
	3RD	Non-Linear Data Structures: Tree Basic Terminologies
12TH	1ST	Non-Linear Data Structures: Trees-Basic Terminologies
	2ND	Definition and Concepts of Binary Trees
	3RD	Definition and Concepts of Binary Trees
13TH	1ST	Representations of a Binary Tree using Arrays
	2ND	2nd INTERNAL ASSESSMENT
	3RD	Representations of a Binary Tree using Linked Lists,
14TH	1ST	Operations on a Binary Tree-Insertion
	2ND	Binary Tree Deletion, Traversals, Types of Binary Trees
	3RD	Binary Tree Deletion, Traversals, Types of Binary Trees
15th	1ST	GRAPHS: Graph Terminologies
	2ND	Representation of Graphs- Set, Linked, Matrix, Graph

		Traversals
	3 RD	Representation of Graphs- Set, Linked, Matrix, Graph Traversals



Lesson plan for ::CSEPC 207/TH4/ DIGITAL ELECTRONICS AND COMPUTER ORGANISATION
Academic Session 2026-27 (W)

Department of Computer Science & Engineering,
 UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SRI KAUSHIK MURMU
SUBJECT:: CSEPC 207/ TH:4 DIGITAL ELECTRONICS AND COMPUTERORGANISATION	NO.OF DAYS/PER WEEK CLASS ALLOTTED:3	SEMESTER FROM DATE:01/07/2026TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1ST	1ST	Introduction to Digital Electronics: Difference Between Analog and Digital Signals
	2ND	Introduction to Digital Electronics: Difference Between Analog and Digital Signals
	3RD	Number Systems: Binary, Octal, Decimal, and Hexa-decimal
2ND	1ST	Number Systems: Binary, Octal, Decimal, and Hexadecimal
	2ND	Conversion Between Number Systems
	3RD	Conversion Between Number Systems
3RD	1ST	Binary Arithmetic Boolean Algebra :
	2ND	Basic Operations, Laws, and Simplification
	3RD	Logic Gates and Circuits: Logic Gates: AND, OR, NOT,
4TH	1ST	Logic Gates NAND, NOR, XOR, XNOR
	2ND	Logic Gates NAND, NOR, XOR, XNOR
	3RD	Design and Simplification of Logic Circuits Using Boolean Algebra,
5TH	1ST	Design and Simplification of Logic Circuits Using Boolean Algebra,
	2ND	Karnaugh Maps (K-Maps) for Simplification Practical Applications of Logic Gates in Real-World Circuits
	3RD	Karnaugh Maps (K-Maps) for Simplification Practical Applications of Logic Gates in Real-World Circuits
6TH	1ST	Combinational and Sequential Circuits: Combinational Circuits: Multiplexers
	2ND	Demultiplexers, Encoders, and Decoders
	3RD	Demultiplexers, Encoders, and Decoders
7TH	1ST	Sequential Circuits: Flip-Flops (SR, JK, D, T) and Their Applications
	2ND	Sequential Circuits: Flip-Flops (SR, JK, D, T) and Their Applications
	3RD	Counters: Synchronous and Asynchronous Counters Registers
8TH	1ST	1ST INTERNAL ASSESSMENT
	2ND	Shift Registers: Types and Uses
	3RD	Fundamentals of Computer Organization: Basic Structure of a Computer
9TH	1ST	Fundamentals of Computer Organization: Basic Structure of a Computer
	2ND	CPU, Memory, Input/Output Devices
	3RD	CPU, Memory, Input/Output Devices
10TH	1ST	Instruction Cycle: Fetch, Decode
	2ND	Execute Memory Organization: Types of Memory (RAM, ROM, Cache, Virtual Memory),
	3RD	Execute Memory Organization: Types of Memory (RAM, ROM, Cache, Virtual Memory),

11 TH	1 ST	Introduction to Buses: Address Bus, Data Bus, and Control Bus
	2 ND	Processor Architecture and Control: Unit Introduction to Microprocessors and Microcontrollers
	3 RD	Processor Architecture and Control: Unit Introduction to Microprocessors and Microcontrollers
12 TH	1 ST	Basics of Arithmetic Logic Unit (ALU) and Control Unit
	2 ND	Basics of Arithmetic Logic Unit (ALU) and Control Unit
	3 RD	Instruction Set Architecture (ISA): RISC vs CISC Pipelining
13 TH	1 ST	Performance Optimization in Processors
	2 ND	2 nd INTERNAL ASSESSMENT
	3 RD	Input/Output Systems and Advanced Topics: I/O Devices and Interfaces
14 TH	1 ST	Keyboard, Mouse, Printers, and Storage Devices Interrupts and DMA (Direct Memory Access),
	2 ND	Keyboard, Mouse, Printers, and Storage Devices Interrupts and DMA (Direct Memory Access),
	3 RD	Overview of Modern Trends: Multicore Processors, GPUs, and Embedded Systems
15 th	1 ST	Overview of Modern Trends: Multicore Processors, GPUs, and Embedded Systems
	2 ND	Mini- Project: Design a Simple Digital Circuit or Simulate a Basic CPU Operation
	3 RD	Mini- Project: Design a Simple Digital Circuit or Simulate a Basic CPU Operation



Lesson plan for :: CSEPC 209/TH5/ ALOGORITHMS
Academic Session 2026-27 (W)
Department of Computer Science & Engineering,
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SRI ABHIRAM BEHERA
SUBJECT:: CSEPC 209/TH:5 ALGORITHMS	NO.OF DAYS/PER WEEK CLASS ALLOTTED:3	SEMESTER FROM DATE:01/07/2026TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1 ST	1 ST	Introduction to AlgorithmsDefinition of algorithm
	2 ND	Introduction to Algorithms: Definition of algorithm
	3 RD	Criteria of algorithms – Input/output, finiteness
2 ND	1 ST	Criteria of algorithms – Input/output, finiteness
	2 ND	Criteria of algorithms- definiteness, effectiveness,
	3 RD	Criteria of algorithms- definiteness, effectiveness,
3 RD	1 ST	Criteria of algorithms- definiteness, effectiveness,
	2 ND	writing an algorithm with pseudocode, algorithms and programs
	3 RD	Algorithmic Complexity: Concept of algorithmic complexity
4 TH	1 ST	space complexity, time complexity
	2 ND	space complexity, time complexity
	3 RD	worst case, average case and best
5 TH	1 ST	worst case, average case and best
	2 ND	case analysis, Big-O notation,
	3 RD	case analysis, Big-O notation,
6 TH	1 ST	Finding the complexity of an algorithm
	2 ND	Recursive algorithms: Concept of iteration and recursion
	3 RD	examples of recursive algorithms
7 TH	1 ST	Fibonacci series, factorial, Tower-of-Hanoi problem
	2 ND	Complexities of recursive algorithms
	3 RD	Conversion of recursive algorithm to iterative algorithm
8 TH	1 ST	1st INTERNAL ASSESSMENT
	2 ND	Algorithm Paradigms: Greedy, Divide and Conquer
	3 RD	Algorithm Paradigms: Greedy, Divide and Conquer
9 TH	1 ST	Algorithm Paradigms: Greedy, Divide and Conquer
	2 ND	Branch and Bound, Dynamic
	3 RD	Branch and Bound, Dynamic
10 TH	1 ST	Branch and Bound, Dynamic
	2 ND	Programming and Backtracking.
	3 RD	Programming and Backtracking.
11 TH	1 ST	Sorting: The sorting problem. Bubble sort
	2 ND	Selection sort, Insertion sort
	3 RD	Merge sort, Quicksort
12 TH	1 ST	Heap sort, Radix sort
	2 ND	Searching: Symbol Tables, Binary Search Trees
	3 RD	Searching: Symbol Tables, Binary Search Trees
13 TH	1 ST	Balanced Search Trees. Hashing, Hash Tables
	2 ND	2nd INTERNAL ASSESSMENT
	3 RD	Graphs: Definition of a directed and undirected graph

14 TH	1 ST	Paths, Cycles, spanning trees
	2 ND	Directed Acyclic Graphs
	3 RD	Topological Sorting. Minimum Spanning Tree algorithms
15 TH	1 ST	Topological Sorting. Minimum Spanning Tree algorithms
	2 ND	Shortest Path algorithms: Dijkstra's algorithm
	3 RD	Flow-based algorithms



Lesson plan for :: CSEPC 211/PR1/_Programming with C++ Lab
Academic Session 2026-27 (W)
Department of Computer Science & Engineering,
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SRI KAUSHIK MURMU
SUBJECT:: CSEPC 211 PR:1 Programming with C++ Lab	NO.OF DAYS/PER WEEK CLASS ALLOTTED:4	SEMESTER FROM DATE:01/07/2026 TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1ST	1ST	Familiarization with C++ programming environment (Editor, Compiler, etc.)
	2ND	Familiarization with C++ programming environment (Editor, Compiler, etc.)
	3RD	Familiarization with C++ programming environment (Editor, Compiler, etc.)
	4TH	Familiarization with C++ programming environment (Editor, Compiler, etc.)
2ND	1ST	Programs using, I/O statements and various operators in C++
	2ND	Programs using, I/O statements and various operators in C++
	3RD	Programs using, I/O statements and various operators in C++
	4TH	Programs using, I/O statements and various operators in C++
3RD	1ST	Programs using C++ expression evaluation and precedence
	2ND	Programs using C++ expression evaluation and precedence
	3RD	Programs using C++ expression evaluation and precedence
	4TH	Programs using C++ expression evaluation and precedence
4TH	1ST	Programs using C++ decision-making statements and branching statements
	2ND	Programs using C++ decision-making statements and branching statements
	3RD	Programs using C++ decision-making statements and branching statements
	4TH	Programs using C++ decision-making statements and branching statements
5TH	1ST	Programs using loop statements in C++
	2ND	Programs using loop statements in C++
	3RD	Programs using loop statements in C++
	4TH	Programs using loop statements in C++
6TH	1ST	Programs to demonstrate applications of n-dimensional arrays in C++
	2ND	Programs to demonstrate applications of n-dimensional arrays in C++
	3RD	Programs to demonstrate applications of n-dimensional arrays in C++
	4TH	Programs to demonstrate applications of n-dimensional arrays in C++

7 TH	1 ST	Programs to demonstrate use of string manipulation functions in C++
	2 ND	Programs to demonstrate use of string manipulation functions in C++
	3 RD	Programs to demonstrate use of string manipulation functions in C++
	4 TH	Programs to demonstrate use of string manipulation functions in C++
8 TH	1 ST	1st INTERNAL ASSESSMENT
	2 ND	Programs to demonstrate parameter passing mechanism in C++
	3 RD	Programs to demonstrate parameter passing mechanism in C++
	4 TH	Programs to demonstrate parameter passing mechanism in C++
9 TH	1 ST	Programs in C++ to demonstrate recursion
	2 ND	Programs in C++ to demonstrate recursion
	3 RD	Programs in C++ to demonstrate recursion
	4 TH	Programs in C++ to demonstrate recursion
10 TH	1 ST	Programs in C++ to demonstrate use of pointers
	2 ND	Programs in C++ to demonstrate use of pointers
	3 RD	Programs in C++ to demonstrate use of pointers
	4 TH	Programs in C++ to demonstrate use of pointers
11 TH	1 ST	Programs in C++ to demonstrate class, object, constructor, destructor.
	2 ND	Programs in C++ to demonstrate class, object, constructor, destructor.
	3 RD	Programs in C++ to demonstrate class, object, constructor, destructor.
	4 TH	Programs in C++ to demonstrate class, object, constructor, destructor.
12 TH	1 ST	Programs in C++ to demonstrate class, object, constructor, destructor.
	2 ND	Programs in C++ to demonstrate class, object, constructor, destructor.
	3 RD	Programs in C++ to demonstrate class, object, constructor, destructor.
	4 TH	Programs in C++ to demonstrate class, object, constructor, destructor.
13 TH	1 ST	Programs in C++ to demonstrate function overloading, operator overloading
	2 ND	2nd INTERNAL ASSESSMENT
	3 RD	Programs in C++ to demonstrate function overloading, operator overloading
	4 TH	Programs in C++ to demonstrate function overloading, operator overloading
14 TH	1 ST	Programs in C++ to demonstrate function overloading, operator overloading
	2 ND	Programs in C++ to demonstrate function overloading, operator overloading
	3 RD	Programs in C++ to demonstrate dynamic memory allocation, file operations
	4 TH	Programs in C++ to demonstrate dynamic memory allocation, file operations
15 TH	1 ST	Programs in C++ to demonstrate dynamic memory allocation, file operations

	2 ND	Programs in C++ to demonstrate dynamic memory allocation, file operations
	3 RD	Programs in C++ to demonstrate dynamic memory allocation, file operations
	4 TH	Programs in C++ to demonstrate dynamic memory allocation, file operations



Lesson plan for :: CSEPC 213/PR2/PROGRAMMING WITH PYTHON LAB
Academic Session 2026-27 (W)
Department of Computer Science & Engineering,
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SRI ABHIRAM BEHERA
SUBJECT::CSEPC/213/PR:2PROG RAMMING WITH PYTHON LAB	NO.OF DAYS/PER WEEK CLASS ALLOTTED:4	SEMESTER FROM DATE:01/07/2026 TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1ST	1ST	Introduction to Python: Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder)
	2ND	Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder)
	3RD	Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder)
	4TH	Install Python and set up an IDE (e.g., PyCharm, VS Code, Jupyter, Spyder)
2ND	1ST	Data types, and operators
	2ND	Data types, and operators
	3RD	Data types, and operators
	4TH	Debug Python scripts to identify and fix errors
3RD	1ST	Debug Python scripts to identify and fix errors
	2ND	Debug Python scripts to identify and fix errors
	3RD	Control Structures and Functions: Implement conditional statements (if, else, elif) in real-life scenarios
	4TH	Implement conditional statements (if, else, elif) in real-life scenarios
4TH	1ST	Implement conditional statements (if, else, elif) in real-life scenarios
	2ND	Write programs using loops (for, while, and nested loops) to solve repetitive tasks
	3RD	Write programs using loops (for, while, and nested loops) to solve repetitive tasks
	4TH	Write programs using loops (for, while, and nested loops) to solve repetitive tasks
5TH	1ST	Define custom functions, including examples of recursion, use lambda functions for inline operations.
	2ND	Define custom functions, including examples of recursion, use lambda functions for inline operations.
	3RD	Define custom functions, including examples of recursion, use lambda functions for inline operations.
	4TH	Define custom functions, including examples of recursion, use lambda functions for inline operations.
6TH	1ST	Data Structures in Python: Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data
	2ND	Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data
	3RD	Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data

	4 TH	Perform CRUD operations on lists, tuples, sets, and dictionaries, use list comprehensions to filter and transform data
7 TH	1 ST	Perform string manipulations using built-in methods
	2 ND	Perform string manipulations using built-in methods
	3 RD	Introduce Python's collections module with practical examples.
	4 TH	Introduce Python's collections module with practical examples.
8 TH	1 ST	1st INTERNAL ASSESSMENT
	2 ND	Introduce Python's collections module with practical examples.
	3 RD	File Handling and Modules: Write programs to read, write, and append text files
	4 TH	Write programs to read, write, and append text files
9 TH	1 ST	Work with CSV files using Python's csv module
	2 ND	Work with CSV files using Python's csv module
	3 RD	Work with CSV files using Python's csv module
	4 TH	Read and write JSON files to store structured data
10 TH	1 ST	Read and write JSON files to store structured data
	2 ND	Read and write JSON files to store structured data
	3 RD	Explore built-in modules like os, math, and datetime, create and import custom modules.
	4 TH	Explore built-in modules like os, math, and datetime, create and import custom modules.
11 TH	1 ST	Object-Oriented Programming (OOP) in Python: Define classes and create objects with attributes and methods
	2 ND	Define classes and create objects with attributes and methods
	3 RD	Implement encapsulation, inheritance, and polymorphism
	4 TH	Implement encapsulation, inheritance, and polymorphism
12 TH	1 ST	Work with magic methods (e.g., init , str) and operator overloading
	2 ND	Work with magic methods (e.g., init , str) and operator overloading
	3 RD	Work with magic methods (e.g., init , str) and operator overloading
	4 TH	Write programs to handle exceptions using try, except, and finally.
13 TH	1 ST	Write programs to handle exceptions using try, except, and finally.
	2 ND	2nd INTERNAL ASSESSMENT
	3 RD	Use NumPy for numerical operations and Pandas for data analysis
	4 TH	Use NumPy for numerical operations and Pandas for data analysis
14 TH	1 ST	Use NumPy for numerical operations and Pandas for data analysis
	2 ND	Use NumPy for numerical operations and Pandas for data analysis
	3 RD	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).

	4 TH	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).
15 TH	1 ST	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).
	2 ND	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).
	3 RD	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).
	4 TH	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).



Lesson plan for :: CSEPC 215/PR3/DATA STRUCTURES AND ALGORITHMS LAB
Academic Session 2026-27 (W)
Department of Computer Science & Engineering,
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SMT SUMITRA MAHAPATRA
SUBJECT::CSEPC/215/PR:3/DATA STRUCTURE AND ALGORITHMS LAB	NO.OF DAYS/PER WEEK CLASS ALLOTTED:4	SEMESTER FROM DATE:14/07/2026 TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1ST	1ST	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.
	2ND	Write a program to analyze and compare the time complexity of basic operations (e.g., searching, insertion) on arrays and linked lists.
	3RD	Linear Data Structures: Implement stack operations (push, pop, peek) using arrays and linked lists
	4TH	Implement stack operations (push, pop, peek) using arrays and linked lists
2ND	1ST	Implement stack operations (push, pop, peek) using arrays and linked lists
	2ND	Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation)
	3RD	Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation)
	4TH	Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation)
3RD	1ST	Implement queue operations (enqueue, dequeue) using arrays and linked lists
	2ND	Implement queue operations (enqueue, dequeue) using arrays and linked lists
	3RD	Implement queue operations (enqueue, dequeue) using arrays and linked lists
	4TH	Write programs for types of queues: circular queues and dequeue
4TH	1ST	Write programs for types of queues: circular queues and dequeue
	2ND	Write programs for types of queues: circular queues and dequeue
	3RD	Linked Lists: Implement singly linked list operations (insertion, deletion, traversal)
	4TH	Implement singly linked list operations (insertion, deletion, traversal)
5TH	1ST	Implement singly linked list operations (insertion, deletion, traversal)
	2ND	Implement singly linked list operations (insertion, deletion, traversal)
	3RD	Write programs to create and manipulate circular and doubly linked lists
	4TH	Write programs to create and manipulate circular and doubly linked lists

6 TH	1 ST	Write programs to create and manipulate circular and doubly linked lists
	2 ND	Write programs to create and manipulate circular and doubly linked lists
	3 RD	Implement stack and queue operations using linked lists
	4 TH	Implement stack and queue operations using linked lists
7 TH	1 ST	Non-Linear Data Structures: Implement binary tree operations (insertion, deletion, traversal)
	2 ND	Implement binary tree operations (insertion, deletion, traversal)
	3 RD	Implement binary tree operations (insertion, deletion, traversal)
	4 TH	Implement binary tree operations (insertion, deletion, traversal)
8 TH	1 ST	1st INTERNAL ASSESSMENT
	2 ND	Develop programs for types of binary trees (binary search tree, AVL tree)
	3 RD	Develop programs for types of binary trees (binary search tree, AVL tree)
	4 TH	Develop programs for types of binary trees (binary search tree, AVL tree)
9 TH	1 ST	Implement graph representations (adjacency list, adjacency matrix)
	2 ND	Implement graph representations (adjacency list, adjacency matrix)
	3 RD	Implement graph representations (adjacency list, adjacency matrix)
	4 TH	Implement graph representations (adjacency list, adjacency matrix)
10 TH	1 ST	Basic graph traversals (BFS, DFS)
	2 ND	Basic graph traversals (BFS, DFS)
	3 RD	Basic graph traversals (BFS, DFS)
	4 TH	Basic graph traversals (BFS, DFS)
11 TH	1 ST	Sorting: Implement sorting algorithms using bubble sort
	2 ND	Implement sorting algorithms using bubble sort
	3 RD	Implement sorting algorithms using selection sort
	4 TH	Implement sorting algorithms using selection sort
12 TH	1 ST	Implement sorting algorithms using insertion sort
	2 ND	Implement sorting algorithms using insertion sort
	3 RD	Implement sorting algorithms using merge sort
	4 TH	Implement sorting algorithms using merge sort
13 TH	1 ST	Implement sorting algorithms using quick sort
	2 ND	2nd INTERNAL ASSESSMENT
	3 RD	Searching: Write programs for searching using binary search trees (BST)
	4 TH	Write programs for searching using binary search trees (BST)
14 TH	1 ST	Write programs for searching using binary search trees (BST)
	2 ND	Write programs for searching using hash tables
	3 RD	Write programs for searching using hash tables
	4 TH	Write programs for searching using hash tables
15 TH	1 ST	Implement symbol table operations using balanced search trees.

	2 ND	Implement symbol table operations using balanced search trees.
	3 RD	Implement symbol table operations using balanced search trees.
	4 TH	Implement symbol table operations using balanced search trees.



Lesson plan for :: CSEPC 217/PR4/DIGITAL ELECTRONICS LAB

Academic Session 2026-27 (W)

Department of Computer Science & Engineering,
UCP Engineering School, Berhampur

DISCIPLINE:CSE	SEMESTER:3RD	NAME OF THE TEACHING FACULTY: SRI KAUSHIK MURMU
SUBJECT::CSEPC/217/PR:4/DIGITAL ELECTRONICS LAB	NO.OF DAYS/PER WEEK CLASS ALLOTTED:4	SEMESTER FROM DATE:14/07/2026TO DATE: 05/11/2026 NO.OF WEEKS:15
WEEK	PERIOD	TOPICS
1 ST	1 ST	To verify the truth tables for all logic gates: NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
	2 ND	To verify the truth tables for all logic gates: NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
	3 RD	To verify the truth tables for all logic gates: NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
	4 TH	To verify the truth tables for all logic gates: NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
2 ND	1 ST	To verify the truth tables for all logic gates: NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
	2 ND	To verify the truth tables for all logic gates: NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS Logic gates and TTL Logic Gates.
	3 RD	Implement and realize Boolean Expressions with Logic Gates.
	4 TH	Implement and realize Boolean Expressions with Logic Gates.
3 RD	1 ST	Implement and realize Boolean Expressions with Logic Gates.
	2 ND	Implement and realize Boolean Expressions with Logic Gates.
	3 RD	Implement and realize Boolean Expressions with Logic Gates.
	4 TH	Implement and realize Boolean Expressions with Logic Gates.
4 TH	1 ST	Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using ICs
	2 ND	Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using ICs
	3 RD	Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using ICs
	4 TH	Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using ICs
5 TH	1 ST	Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using ICs
	2 ND	Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using ICs
	3 RD	Implement parallel and serial full-adder using ICs.
	4 TH	Implement parallel and serial full-adder using ICs.
6 TH	1 ST	Implement parallel and serial full-adder using ICs.

	2 ND	Implement parallel and serial full-adder using ICs.
	3 RD	Implement parallel and serial full-adder using ICs.
	4 TH	Implement parallel and serial full-adder using ICs.
	4 TH	Implement parallel and serial full-adder using ICs.
7 TH	1 ST	Design and development of multiplexer and De-multiplexer using multiplexer ICs
	2 ND	Design and development of multiplexer and De-multiplexer using multiplexer ICs
	3 RD	Design and development of multiplexer and De-multiplexer using multiplexer ICs
	4 TH	Design and development of multiplexer and De-multiplexer using multiplexer ICs
8 TH	1 ST	1ST INTERNAL ASSESSMENT
	2 ND	Design and development of multiplexer and De-multiplexer using multiplexer ICs
	3 RD	Verification of the function of SR, D, JK, and T Flip Flops.
	4 TH	Verification of the function of SR, D, JK, and T Flip Flops.
9 TH	1 ST	Verification of the function of SR, D, JK, and T Flip Flops.
	2 ND	Verification of the function of SR, D, JK, and T Flip Flops.
	3 RD	Verification of the function of SR, D, JK, and T Flip Flops.
	4 TH	Verification of the function of SR, D, JK, and T Flip Flops.
10 TH	1 ST	Verification of the function of SR, D, JK, and T Flip Flops.
	2 ND	Verification of the function of SR, D, JK, and T Flip Flops.
	3 RD	Design controlled shift registers.
	4 TH	Design controlled shift registers.
11 TH	1 ST	Design controlled shift registers.
	2 ND	Design controlled shift registers.
	3 RD	Design controlled shift registers.
	4 TH	Design controlled shift registers.
12 TH	1 ST	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
	2 ND	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
	3 RD	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
	4 TH	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
13 TH	1 ST	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
	2 ND	2ND INTERNAL ASSESSMENT
	3 RD	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
	4 TH	Construct a Single Digit Decade Counter (0-9) with a 7-segment display.
14 TH	1 ST	To design a programmable Up-Down Counter with a 7-segment display.
	2 ND	To design a programmable Up-Down Counter with a 7-segment display.
	3 RD	To design a programmable Up-Down Counter with a 7-segment display.
	4 TH	To design a programmable Up-Down Counter with a 7-segment display.
15 TH	1 ST	To design a programmable Up-Down Counter with a 7-segment display.
	2 ND	To design a programmable Up-Down Counter with a 7-segment display.

	3 RD	To design a programmable Up-Down Counter with a 7-segment display.
	4 TH	To design a programmable Up-Down Counter with a 7-segment display.