Government Polytechnic, Balangir

Department of Computer Science Engineering

LESSON PLAN 2026-27(WINTER)

NAME OF THE TEACHER : ANJALI MUNDA, GUEST FACULTY

Subject: **DATA STRUCTURES** (Course Code:CSEPC 205/TH3)
Program: Diploma in Computer Science and Engineering
Semester: 3rd
Total Contact Hours: 45
Total Marks: 100
Assessment: Internal Assessment – 30, End Term – 70

After completion of the course, the students will be able to:
CO1: Explain fundamental data structure concepts, classifications, and algorithm analysis.
CO2: Apply linear data structures such as stacks, queues, and their variations.
CO3: Develop linked list structures, including singly, doubly, and circular linked lists.
CO4: Implement non-linear data structures like trees and perform operations such as insertion, deletion, and traversal.
CO5: Describe graph representations and traversal techniques for efficient data organization.

Period	Unit	Topic / Sub-Topic	Learning Objectives	Activity	Course Outcome	Learning Methodology	Homework
1	I	Basic Terminology	Define standard data structure terms.	Concept Mapping	CO1	Lecture & Discussion	Read textbook introduction on data formats.
2	I	Classification of Data Structure	Classify structures into linear and non-linear types.	Think-Pair-Share	CO1	Lecture & Visuals	List real-world examples for each class.
3	I	Operations on Data Structure	Identify core operations like traversal, insertion, and deletion.	Scenario Analysis	CO1	Demonstration	Write a summary of core operations.
4	I	Asymptotic analysis introduction	Explain the need for algorithmic efficiency measurement.	Problem Solving	CO1	Lecture & Discussion	Review basic mathematical functions.
5	I	Asymptotic notation	Apply Big-O notation to simple algorithms.	Code Tracing	CO1	Lecture & Coding	Calculate complexity for a for loop.
6	I	Worst-case analysis of algorithms	Differentiate between best, average, and worst-case scenarios.	Scenario Simulation	CO1	Demonstration	Identify worst-case scenarios in daily tasks.
7	I	Algorithm analysis practice	Evaluate the time complexity of nested loops.	Dry Run Exercises	CO1	Problem Solving	Complete a worksheet on time complexity.
8	I	Unit I Review & Synthesis	Consolidate foundational concepts and analysis techniques.	Interactive Q&A	CO1	Peer Review	Review notes for Unit II transition.

9	II	Stacks: Introduction & Array Representation	Define stack properties (LIFO) and map them to arrays.	Memory Visual Drawing	CO2	Lecture & Visuals	Sketch an array memory map for a stack.
10	II	Operations on a Stack	Implement Push and Pop operations theoretically.	Algorithm Tracing	CO2	Demonstration	Write pseudocode for stack overflow.
11	II	Applications of Stacks: Infix-to-Postfix Transformation	Convert algebraic expressions using stack rules.	Board Work	CO2	Problem Solving	Convert 5 given infix expressions.
12	II	Applications of Stacks: Evaluating Postfix Expressions	Calculate final values from postfix notations.	Group Calculation	CO2	Problem Solving	Solve 5 postfix evaluation problems.
13	II	Queues: Introduction & Array Representation	Define queue properties (FIFO) and array mapping.	Concept Comparison	CO2	Lecture	Compare stack and queue structures.
14	II	Operations on a Queue	Formulate Enqueue and Dequeue logic.	Algorithm Tracing	CO2	Demonstration	Write pseudocode for queue underflow.
15	II	Types of Queues: Circular Queue	Solve the limitation of linear queues using modulo arithmetic.	Circular Array Drawing	CO2	Lecture & Visuals	Trace a circular queue operation on paper.
16	II	Types of Queues: DeQueue	Explain double-ended queue operations.	Scenario Analysis	CO2	Demonstration	List practical applications of DeQueues.
17	II	Applications of Queues: Round Robin Algorithm	Apply queue concepts to OS process scheduling.	Task Scheduling Game	CO2	Problem Solving	Simulate Round Robin for 3 mock processes.
18	II	Unit II Review & Practice	Validate understanding of linear static structures.	Quiz	CO2	Assessment	Prepare for dynamic allocation topics.
19	III	Linked Lists: Singly Linked List Intro	Contrast static arrays with dynamic linked nodes.	Concept Mapping	CO3	Lecture & Discussion	Read chapter on dynamic memory.
20	III	Representation in Memory	Diagram node structures (data + pointer).	Memory Mapping	CO3	Visual Demonstration	Sketch a 4-node linked list.
21	III	Operations on a Single Linked List: Traversal	Trace pointers to visit every node sequentially.	Pointer Tracing	CO3	Problem Solving	Write traversal pseudocode.
22	III	Operations on a Single Linked List: Insertion	Re-link pointers to add nodes at various positions.	Link Disconnect/Connect	CO3	Lecture & Coding	Draw pointer changes for insertion.

23	III	Operations on a Single Linked List: Deletion	Bypass nodes safely to remove them from the list.	Garbage Collection Discussion	CO3	Demonstration	Draw pointer changes for deletion.
24	III	Circular Linked Lists	Design a list where the tail points to the head.	Architecture Exercise	CO3	Lecture & Visuals	Compare circular vs. singly list diagrams.
25	III	Operations on Circular Linked Lists	Adapt traversal algorithms for circular bounds.	Algorithm Tracing	CO3	Problem Solving	Write a circular traversal loop condition.
26	III	Doubly Linked Lists	Construct nodes with forward and backward pointers.	Concept Mapping	CO3	Lecture	Sketch a doubly linked node.
27	III	Operations on Doubly Linked Lists: Insertion/Deletion	Manage multiple pointer updates simultaneously.	Complex Pointer Tracing	CO3	Problem Solving	Write insertion pseudocode for a doubly list.
28	III	Linked List Representation of Stack	Map LIFO behavior to dynamic node allocation.	Data Structure Integration	CO3, CO2	Lecture & Demo	Differentiate array-stack vs. list-stack.
29	III	Operations of Stack using Linked List	Implement Push/Pop by manipulating the head node.	Board Work	CO3, CO2	Problem Solving	Trace Push/Pop on a mock dynamic stack.
30	III	Linked List Representation of Queue	Map FIFO behavior utilizing front and rear pointers.	Architecture Exercise	CO3, CO2	Lecture & Demo	Sketch front/rear pointers on a linked list.
31	III	Operations of Queue using Linked List	Implement Enqueue/Dequeue with dynamic boundaries.	Algorithm Tracing	CO3, CO2	Problem Solving	Trace dynamic queue operations on paper.
32	III	Unit III Review & Synthesis	Ensure mastery of dynamic pointer manipulation.	Peer Code Review	CO3	Discussion	Complete textbook exercises on lists.
33	IV	Trees: Basic Terminologies	Define root, leaf, edge, depth, and height.	Tree Diagramming	CO4	Lecture & Visuals	Identify terminologies on a provided tree graph.
34	IV	Concepts and Types of Binary Trees	Differentiate full, complete, and strict binary trees.	Classification Activity	CO4	Demonstration	Classify 5 different drawn trees.
35	IV	Representations of a Binary Tree using Arrays	Calculate parent/child indices in sequential storage.	Index Calculation Math	CO4	Problem Solving	Map a simple tree into an array index table.

36	IV	Representations of a Binary Tree using Linked Lists	Construct tree nodes with left/right child pointers.	Memory Visual Drawing	CO4	Lecture & Demo	Sketch a linked representation of a binary tree.
37	IV	Operations on a Binary Tree: Insertion	Navigate branches to place new data appropriately.	Logic Flow Exercise	CO4	Problem Solving	Trace the insertion of 5 numerical values.
38	IV	Operations on a Binary Tree: Deletion	Safely remove *leaf nodes and* nodes with children.	Structure Rebuilding	CO4	Demonstration	Trace deletion and tree restructuring.
39	IV	Traversals of Binary Trees	Execute In-order, Pre-order, and Post-order logic.	Path Tracing Game	CO4	Problem Solving	Perform all 3 traversals on a sample tree.
40	IV	GRAPHS: Graph Terminologies	Define vertices, edges, directed, and undirected graphs.	Network Drawing	CO5	Lecture & Visuals	Identify nodes and edges in a real-world map.
41	IV	Representation of Graphs: Set & Matrix	Model graph connections using Adjacency Matrices.	Matrix Construction	CO5	Problem Solving	Build a matrix for a 4-node graph.
42	IV	Representation of Graphs: Linked	Model graph connections using Adjacency Lists.	List Construction	CO5	Demonstration	Build an adjacency list for the same graph.
43	IV	Graph Traversals: BFS	Explore graphs iteratively using Queue *structures.*	Algorithm Tracing	CO5	Lecture & Coding	Trace BFS on a 6-node graph.
44	IV	Graph Traversals: DFS	Explore graphs deeply using Stack/Recursive structures.	Algorithm Tracing	CO5	Lecture & Coding	Trace DFS on the same 6-node graph.
45	IV	Course Wrap-up & Final Review	Correlate all structures and prepare for the final exam.	Q&A Session	CO1-CO5	Discussion	Review the entire semester syllabus.

Aniali Munda
13/07/26
Signature of Teacher

Deepak Kumar Barik
13.07.2026
Signature of HOD