Professional Data Structures and Algorithms (DSA) Course Outline

Module 1: Introduction to Data Structures and Algorithms

- What is DSA? Why is it important?
- Time Complexity and Big-O Notation
- Space Complexity Analysis
- Best Practices for Writing Efficient Code

Module 2: Arrays and Strings

- Introduction to Arrays & Static vs. Dynamic Arrays
- Searching Algorithms: Linear Search, Binary Search
- Sorting Algorithms: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort
- String Manipulation Techniques (Reversal, Anagrams, Palindromes)

Module 3: Linked Lists

- Introduction to Singly and Doubly Linked Lists
- Operations: Insertion, Deletion, Searching, Traversing
- Circular Linked Lists
- Detecting and Removing Loops in Linked Lists

Module 4: Stacks and Queues

- Understanding LIFO and FIFO Concepts
- Stack Implementation Using Arrays and Linked Lists
- Queue and Deque Implementation
- Applications: Balanced Parentheses, Expression Evaluation, LRU Cache

Module 5: Recursion and Backtracking

- Understanding Recursion and Base Cases
- Tail Recursion vs. Head Recursion
- Backtracking Techniques (N-Queens Problem, Subset Sum)
- Optimizing Recursive Solutions with Memoization

Module 6: Trees and Binary Search Trees (BSTs)

- Introduction to Trees and Binary Trees
- Binary Search Tree (BST): Insertion, Deletion, Searching
- Tree Traversal Techniques: Inorder, Preorder, Postorder
- Balanced Trees (AVL Trees, Red-Black Trees)
- Applications of Trees (Huffman Encoding, Segment Trees)

Module 7: Heaps and Priority Queues

- Introduction to Heaps (Min-Heap, Max-Heap)
- Heap Operations: Insertion, Deletion, Heapify
- Priority Queue Implementation
- Applications of Heaps (Dijkstra's Algorithm, Job Scheduling)

Module 8: Graphs and Graph Algorithms

- Representation of Graphs (Adjacency Matrix & List)
- Graph Traversal Algorithms: BFS, DFS
- Shortest Path Algorithms: Dijkstra's, Bellman-Ford
- Minimum Spanning Tree: Kruskal's and Prim's Algorithms
- Topological Sorting and Cycle Detection

Module 9: Dynamic Programming (DP)

- Introduction to DP and Recursion vs. DP
- Classical DP Problems: Fibonacci, Knapsack, Longest Common Subsequence
- Bottom-Up vs. Top-Down Approaches
- Advanced DP Problems (Matrix Chain Multiplication, Palindromic Substrings)

Module 10: Advanced Topics and Interview Preparation

- Bit Manipulation Techniques
- Sliding Window & Two-Pointer Techniques
- Tries (Prefix Trees) and Their Applications
- Disjoint Set Union (DSU) and Kruskal's Algorithm
- Competitive Programming Strategies

Final Module: Capstone Project & Certification

- Implementing a Real-World DSA-Based Application
- Coding Challenges and Mock Interviews
- Performance Optimization Techniques
- Course Completion Certification