Introduction

Master the backbone of programming with our 3-Month Data Structures This course is tailored for aspiring developers, competitive programmers, and anyone looking to excel in technical interviews. With a mix of theory, hands-on coding, and real-world applications, you'll develop a solid understanding of data structures and their importance in solving complex problems.

Month 1: Fundamentals of Data Structures

Week 1: Introduction to Data Structures

- What Are Data Structures and Why Are They Important?
- Classifications: Linear vs. Non-Linear
- Complexity Analysis: Time and Space Complexity Basics
- Big-O Notation Overview
- Hands-on Projects:
- Analyze the Complexity of Simple Algorithms.

Week 2: Arrays and Linked Lists

- Arrays: Representation, Operations (Insert, Delete, Search)
- 2D Arrays and Applications
- Linked Lists:
- Singly Linked List
- Doubly Linked List
- Circular Linked List
- Hands-on Projects:
- Create a Music Playlist Using Linked Lists.

Week 3: Stacks and Queues

- Stack: LIFO Principle, Implementation, and Applications
- Queue: FIFO Principle, Circular Queue, and Deque

- Priority Queue Using Heaps
- Hands-on Projects:
- Design a Browser’s Back/Forward Navigation Using Stacks.
- Implement a Ticket Booking System Using Queues.

Week 4: Recursion and Its Role in Data Structures

- What Is Recursion?
- Recursive Algorithms in Sorting and Searching
- Backtracking and Problem-Solving Techniques
- Hands-on Projects:
- Solve the Tower of Hanoi Problem.

Month 2: Advanced Data Structures

Week 1: Trees

- Binary Trees: Traversals (Inorder, Preorder, Postorder)
- Binary Search Trees (BST)
- AVL Trees: Self-Balancing BST
- Applications of Trees in Real-World Scenarios
- Hands-on Projects:
- Build a Family Tree Visualizer.

Week 2: Graphs

- Representation: Adjacency Matrix and Adjacency List
- Graph Traversals: BFS and DFS
- Shortest Path Algorithms: Dijkstra’s and Floyd-Warshall
- Minimum Spanning Tree: Kruskal’s and Prim’s Algorithms
- Hands-on Projects:

- Build a Route Planner for a City Using Graphs.

Week 3: Hashing and Hash Tables

- Hash Functions and Collision Resolution Techniques
- Applications of Hashing in Real Life
- Hands-on Projects:
- Develop a Dictionary Application Using Hash Tables.

Week 4: Heaps

- Min-Heap and Max-Heap Implementation
- Applications in Priority Queues
- Hands-on Projects:
- Build a Task Scheduler Using Heaps.

Month 3: Real-World Applications and Problem Solving

Week 1: Sorting and Searching Algorithms

- Sorting: Bubble Sort, Merge Sort, Quick Sort, and Heap Sort
- Searching: Linear Search, Binary Search, and Interpolation Search
- Hands-on Projects:
- Compare and Visualize Sorting Algorithms.

Week 2: Dynamic Programming and Greedy Algorithms

- Introduction to Dynamic Programming: Concepts and Applications
- Greedy Algorithms and Their Use Cases
- Hands-on Projects:
- Solve the Knapsack Problem Using DP and Greedy Techniques.

Week 3: Advanced Topics and Real-Life Use Cases

- Trie: Implementation and Applications in Autocomplete

- Disjoint Set (Union-Find): Applications in Networking
- Segment Trees: Range Queries and Updates
- Hands-on Projects:
- Implement an Autocomplete System for a Search Engine.

Week 4: Capstone Project and Interview Preparation

- Capstone Project Options:
- Develop a Pathfinding Algorithm for Maze Solving.
- Implement a Social Network Graph Analysis Tool.
- Mock Interviews Focused on Data Structures and Algorithm Questions
- Tips for Solving Competitive Programming Challenges

Key Features of the Program

- Problem-Solving Focus: Learn to think like a programmer and solve problems efficiently.
- Real-World Applications: Each concept tied to practical use cases.
- Mentorship: Get one-on-one guidance from experienced mentors.
- Interview-Ready: Comprehensive preparation for coding interviews with industry-standard questions.

Join Now to unlock the power of Data Structures and become a problem-solving expert!