

Duration: 80 Hours
Live Class: 4 Months

Month 1: C++ Fundamentals & Basic Data Structures

Week 1: Introduction to C++ Programming

- Introduction to C++ and its IDEs (Visual Studio, VS Code)
- Basic Syntax, Data Types, Variables, Input/Output
- Conditional Statements and Loops (If-Else, Switch, For, While, Do-While)
- Introduction to DSA: What is DSA, Why it's Important?

Week 2: Functions and Recursion

- Functions in C++: Call by Value, Call by Reference
- Recursion Basics
- Problem Solving using Recursion (Factorial, Fibonacci)
- Practice Problem Set on Functions

Week 3: Arrays and Searching Algorithms

- Arrays: Declaration, Initialization, Traversing
- Binary Search vs Linear Search
- Sorting Algorithms: Selection Sort, Insertion Sort, Bubble Sort
- Hands-on Practice: Solving Sorting and Searching Problems

Week 4: Object-Oriented Programming (OOP)

- Introduction to OOP Concepts: Classes, Objects, Constructors
- Encapsulation, Abstraction, Inheritance
- Polymorphism: Function Overloading and Overriding
- Mini Project: Build a Simple Bank Management System

Month 2: Intermediate C++ and Data Structures

Week 1: Advanced OOP Concepts

- Operator Overloading, Friend Functions
- Dynamic Memory Allocation (new, delete)
- Smart Pointers (Unique, Shared, Weak Pointers)
- Practice Project: Implementing a Custom String Class

Week 2: Linked Lists

- Introduction to Linked Lists: Singly and Doubly Linked Lists
- Circular Linked Lists
- Insertion, Deletion, and Traversal in Linked Lists
- Hands-on Practice: Solving Linked List Problems

Week 3: Stacks and Queues

- Introduction to Stack and Queue Data Structures
- Implementing Stack and Queue Using Arrays and Linked Lists
- Applications of Stack (Parenthesis Matching, Infix to Postfix)
- Queue Variations: Circular Queue, Priority Queue
- Problem Solving on Stacks and Queues

Week 4: Complexity Analysis and Recursion

- Understanding Time and Space Complexity (Big O Notation)
- Tail Recursion and Recursion vs Iteration
- Advanced Recursive Problems (N-Queens, Sudoku Solver)
- Practice Problems on Complexity and Recursive Algorithms

Month 3: Advanced Data Structures and Algorithms

Week 1: Trees and Binary Search Trees (BST)

- Introduction to Trees: Binary Trees, Binary Search Trees (BST)
- Tree Traversal Techniques (In-order, Pre-order, Post-order)
- Insertion, Deletion, and Searching in BST
- Solving Tree-Based Problems

Week 2: Heaps and Priority Queues

- Introduction to Heaps (Min-Heap, Max-Heap)
- Heapify Process and Heapsort Algorithm
- Implementing Priority Queue Using Heap
- Hands-on Practice with Heaps and Priority Queues

Week 3: Graph Theory

- Graph Representation: Adjacency Matrix, Adjacency List
- Breadth First Search (BFS) and Depth First Search (DFS)
- Shortest Path Algorithms: Dijkstra's, Bellman-Ford
- Graph Applications in Real Life (Social Networks, Web Crawling)

Week 4: Greedy Algorithms & Dynamic Programming (DP)

- Introduction to Greedy Algorithms: Activity Selection, Kruskal's Algorithm
- Introduction to DP: Memoization and Tabulation
- Solving Basic DP Problems (Knapsack, Fibonacci, Coin Change)
- Advanced DP Problems and Optimization Techniques

Month 4: Advanced Topics and Interview Preparation

Week 1: Advanced Dynamic Programming

- DP on Trees: Longest Path, Diameter of a Tree
- DP on Graphs: Floyd-Warshall, Topological Sorting
- Hands-on Practice on Advanced DP Problems

Week 2: Backtracking Algorithms

- Introduction to Backtracking: Permutations, Combinations
- Solving Backtracking Problems: N-Queens, Rat in a Maze
- Hands-on Backtracking Projects

Week 3: Advanced C++ Topics

- Standard Template Library (STL): Vectors, Sets, Maps, Algorithms
- Lambda Functions, Function Pointers
- File Handling in C++
- Project: Implementing a Student Database System Using File Handling

Week 4: Interview Preparation & Capstone Project

- Mock Interviews: Common DSA and C++ Interview Questions
- Competitive Coding Practice
- Capstone Project: Build a Complete Library Management System Using C++ and DSA
- Review and Final Q&A Session

Key Highlights for Students:

1. **Project-Based Learning:** Every concept comes with hands-on coding projects.
2. **Career-Centric Curriculum:** Focus on industry-relevant problem-solving.
3. **Competitive Coding Focus:** Weekly problems from competitive coding platforms.
4. **Capstone Project:** A large-scale project to demonstrate mastery in DSA and C++.
5. **Mock Interviews:** Preparation for technical interviews and coding rounds.

Projects:

- **Banking Management System**
- **Library Management System**