Complete DSA with JAVA

Duration: 80 Hours
Live Class: 4 Months

Month 1: Java Fundamentals & Basics of DSA

Week 1: Introduction to Java Programming

- Overview of Java and its Ecosystem
- Setting up Java Development Environment (JDK, IDEs like IntelliJ, Eclipse)
- Data Types, Variables, Operators, Input/Output
- Control Flow (If-Else, Switch, Loops)
- Introduction to DSA: What and Why?

Week 2: Functions & Arrays

- Methods and Functions in Java
    a) Pass by Value, Recursion Basics
- Arrays: Declaration, Initialization, Multi-Dimensional Arrays
    a) Array Traversal and Manipulation
- Sorting Algorithms: Selection Sort, Bubble Sort, Insertion Sort

Week 3: Object-Oriented Programming (OOP) Basics

- Classes and Objects, Constructors
- OOP Principles: Encapsulation, Abstraction
- Access Modifiers and Static Keyword

Week 4: Advanced OOP Concepts

- Inheritance, Polymorphism, Method Overloading/Overriding
- Interfaces, Abstract Classes
- Hands-on Projects: Small OOP-Based Projects (e.g., Library Management System)

Month 2: Intermediate Java & Data Structures

Week 1: Strings and Their Operations

- String Manipulation and Built-in Methods
- StringBuilder and StringBuffer
- Introduction to Regular Expressions
- Solving Basic String-Based Problems (e.g., Palindrome, Anagram)

Week 2: Introduction to Data Structures

- Complexity Analysis (Time and Space Complexity)
- Arrays and Linked Lists (Singly and Doubly Linked Lists)
- Stack and Queue in Java (with Use Cases)
- Solving Problems with Stacks and Queues (Parenthesis Matching, Balanced Expressions)

Week 3: Recursion & Backtracking

- Understanding Recursion in Depth
- Introduction to Backtracking (e.g., N-Queens, Sudoku Solver)
- Solving Array, String, and Linked List Problems with Recursion

Week 4: Sorting and Searching Algorithms

- QuickSort, MergeSort, Binary Search, and their Optimizations
- Applications of Binary Search (e.g., Rotated Sorted Arrays)
- Solving Advanced Array Problems

Month 3: Advanced Data Structures

Week 1: Sets, HashMaps, and Trees

- Introduction to HashMap, HashSet
- Collision Handling in Hashing
- Trees: Binary Trees, Binary Search Trees (BST)
- Hands-on Practice: Tree Traversals (In-order, Pre-order, Post-order)

Week 2: Advanced Trees and Heaps

- AVL Trees, Red-Black Trees
- Priority Queue (Heap Implementation)
- Applications of Heaps (Heap Sort, Kth Largest Element)
- Solving Problems Using Trees and Heaps

Week 3: Graphs & Their Applications

- Introduction to Graphs (Representation: Adjacency Matrix, List)
- Depth First Search (DFS), Breadth First Search (BFS)
- Shortest Path Algorithms: Dijkstra’s, Bellman-Ford
- Real-World Graph Problems (e.g., Social Network Analysis, Web Crawling)

Week 4: Greedy Algorithms & Dynamic Programming

- Greedy Methodologies: Activity Selection, Knapsack Problem
- Dynamic Programming: Introduction and Basic Problems (e.g., Fibonacci, Coin Change)
- Solving Real-World Problems Using Dynamic Programming

Month 4: Advanced Java and Problem-Solving

Week 1: Java Collections Framework (JCF)

- Introduction to Collections: List, Set, Map
- ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap
- Collections Utility Methods (Sorting, Searching, etc.)
- Hands-on Practice: Real-World Application Using JCF

Week 2: Multithreading and Concurrency

- Introduction to Threads in Java
- Synchronization and Inter-Thread Communication
- Concurrency Utilities (Executor Framework, Future, Callable)
- Project: Implementing a Multithreaded Application

Week 3: File Handling and Serialization

- File I/O in Java (Reading and Writing Files)
- Serialization and Deserialization in Java
- Handling Different Types of Files (Text, CSV, Binary)
- Project: Build a File-Based Database System

Week 4: Capstone Project & Competitive Coding

- Capstone Project: Build a Complete Java Application
- Practice Competitive Coding
- Review of Data Structures and Algorithms with Mock Tests
- Interview Preparation: Common DSA Problems in Java

Projects:

- **Library Management System**
- **Banking System with OOP Principles**

Key Highlights for Students:

1. **Industry-Relevant Curriculum**: Focus on solving real-world challenges using Java.
2. **Hands-on Projects**: Every concept is backed by a practical project.
3. **Competitive Coding Focus**: Preparing students for coding competitions.
4. **Interview Preparation**: DSA-focused sessions to clear technical rounds.
5. **Capstone Project**: A large-scale project to showcase in portfolios.