Advance DSA – Competitive Programing using C++

An all-encompassing course designed to solidify your programming fundamentals while fostering a profound, practical grasp of Data Structures and Algorithms. This knowledge is your steppingstone to achieving success in prominent multinational corporations with lucrative compensation packages.

Course Highlights:

- Covers **C++ Programming**.
- Builds Strong Foundation in **Design and Analysis of Algorithms**
- Covers all **Data Structures** used in **modern applications** with **practical applications, implementations and problem solving.**
- Covers **MySQL Database** make you Job Ready
- **Problem Solving** for Interview Preparation, **Mock Interviews**, **Competitive Programming** to AIM big packages

Course in Details:

1. Let's Start with Learning C++ in Depth

- Installing C++ and setting up Development Environments
- Running First Hello World! Program and Understanding Basics of programming
- Understanding Basic Structure of C++ program
- Understanding Compilation and execution process
- Learn C++ Syntax, Variables, Data Types to start solving problems using C++
- Input and Output in c++
- Operators and expressions

2. Exploring Control Statements in C++

- Decision Statements using if-else and Switch Cases
- Looping in C++ using while, do-while, and for loops
- Understanding use Break, continue and flag variables in loops with practical hands on real world problems

3. **Functions and Prototypes**
    - Understanding functions and there use in programming
    - Writing and Calling of Functions in C++
    - Writing reusable piece of code using functions in C++
    - Function Overloading
    - C++ standard Libraries

4. **Object Oriented Programming (OOPs) using C++**
    - Introduction to OOPs concepts
    - Classes and Objects
    - Encapsulation, inheritance, and polymorphism
    - Constructors and Destructors
    - Operator Overloading

5. **Standard Template Library (STL)**
    - Containers(vectors, lists, maps, etc.)
    - Iterators and Algorithms
    - Generic Programming with STL

6. **Advanced C++ Features**
    - Pointers
    - Exception Handing
    - Smart Pointers
    - File I/0 and streams
    - Memory Management
    - Preprocessor Directives
    - Template and Generic programming

7. **Best Practices**
    - Coding Style and conventions
    - Debugging and error handing
    - Code Optimization
    - Testing and quality assurance

8. **Introduction to Data Structures and Algorithms**
    - Analysis of Algorithms
    - Order of Growth
    - Asymptotic Notations – Big Oh, Omega, Theta Notation
    - Analysis of Common Loops
    - Analysis of Recursion

- Space complexity
- Practice Problems

9. Mathematics

- Count digits, Palindrome, Armstrong Number
- Prime Factors, Prime Numbers, Factorial of a Number
- GCD, LCM, Sieve of Eratosthenes
- Practice Problems

10.Bit Manipulation

- Bitwise Operators – AND, OR, XOR, NOT, LEFT SHIFT, RIGHT SHIFT
- Practice Problems

11.Recursion

- Introduction and Application of Recursion
- Factorial, N-th Fibonacci number, Prime using Factorial
- Sum of first N numbers, print n to 1, 1 to n, Sum of digits, Rod Cutting
- Checking Palindrome, Tail Recursion, Subsets of a set
- Tower of Hanoi
- Practice Problem

12.Arrays

- Introduction to Arrays, Memory Representation
- Types of Arrays – Fixed Sized Arrays and Dynamic Sized Arrays
- Operations on Arrays – Searching, Insertions, Deletion, Reversing
- Practice Problems

13.Searching

- Linear Searching
- Binary Search
- Practice Problems

14.Sorting

- Bubble Sort, Selection Sort, Insertion Sort
- Merge Sort, Quick Sort, bucket Sort
- Heap Sort, Cycle Sort, Radix Sort
- Practice Problems

15.Matrix

- Introduction to Matrix in C++

- Multidimensional Matrix, Pass Matrix as Arguments
- Printing Matrix in a snake pattern
- Transposing, Rotating a matrix
- Spiral Traversal
- Matrix Multiplication
- Practice Problems

16.Hashing

- Introduction and Application of Hashing
- Working and examples on various Hash Functions
- Introduction and Various techniques on Collision Handling
- Chaining and its implementation
- Open Addressing and its Implementation
- Chaining V/S Open Addressing
- C++ - Unordered Set and Unordered Map
- Practice Problem

17.Strings

- Discussion of String DS
- String in CPP
- Rabin Karp Algorithm
- KMP Algorithm
- Practice Problems

18.Linked List

- Introduction and Implementation in CPP
- Singly Linked List and It's Operations – Insertion, Deletion, Traversal
- Doubly Linked List
- Circular Linked List
- Loop Problems – Floyd Cycle Detection
- Practice Problems

19.Stack

- Introduction and Implementation of Stack Data Structure
- Applications of Stack
- Operations – Insertion, Deletion, Traversal
- Infix, Prefix and Postfix Evaluation
- Practice Problems

20.Queue

- Introduction and Implementation

- C++ STL Queue
- Operations – Insertion, Deletion, Traversal
- Circular Queue
- Practice Problem

21.Deque

- Introduction and Application
- Implementation In C++ STL
- Practice Problems

22.Tree

- Introduction, Application
- Binary Tree
- Tree Traversal – Inorder, Preorder, Postorder, Level Order, Spiral

23.Binary Search Tree (BST)

- Background, Introduction and Application
- Implementation of Search in BST
- Insertion in BST
- Deletion in BST
- Searching in BST
- Practice problems

24.AVL Tree

- Introduction to AVL Trees
- Applications of AVL Trees
- AVL Tree Operations
- Balance Restoration

25.Heap

- Introduction to Heaps
- Heap Representation
- Heap Operations
- Heap Sort
- Applications of Heaps

26.Tries

- Understanding concepts of Tries
- Basic Terminology and Definitions
- Trie Structure, Nodes and Edges in Trie
- Common use cases of Tries in real word applications
- Autocomplete and spell checking

- Efficiency and Optimization or Tries
- Practice Problems

27.Graph

- Introduction to Graph
- Graph Representation – Adjacency Matrix, Adjacency List in CPP
- Adjacency Matrix VS List
- Breadth-First Search
- Depth First Search
- Shortest Path in Directed Acyclic Search Graph
- Prim's Algorithm/Minimum Spanning Tree
- Dijkstra's Shortest Path Algorithm
- Bellman-Ford Shortest Path Algorithm
- Kruskal's algorithms
- Kosaraju's Algorithm
- Bridges in Graph
- Trajan's Algorithm
- Practice Problems

28.Greedy

- Introduction
- Activity Selection Problem
- Fractional Knapsack
- Job Sequencing Problem
- Huffman Coding
- Practice Problems

29.Backtracking

- Concepts of Backtracking
- Rat In a Maze
- N Queen Problem
- Sudoku Problem
- Practice Problems

30.Dynamic Programming

- Introduction
- Dynamic Programming – Memorization and Tabulation
- 0-1 Knapsack Problem
- Matrix Chain Multiplication
- Practice Problems

31.Major Dynamic Programming Problems

- Fibonacci Sequence
- 0/1 Knapsack Problem
- Longest Common Subsequence (LCS)
- Longest Increasing Subsequence (LIS)
- Edit Distance (Levenshtein Distance)
- Coin Change Problem
- Subset Sum Problem
- Matrix Chain Multiplication
- Rod Cutting Problem
- Palindromic Substring Problem