

# Alpha Placement Course

Java 



' Let not the fruit of action be your motive to action. Your concern is with action alone, not with the fruit of action. '

- *The Bhagavad Gita*

# What's new?

Our experience tells us that students on average are **3 times** more serious in a batch. The completion of course and watch time of students increases exponentially.

**300+**

Coding Questions

**400+**

videos. Classes on alternate days for next 2.5 months

**Live**

7 Live Mentorship Sessions

# Batch Overview

- ✓ Complete Java
- ✓ Complete Data Structures & Algorithms
- ✓ Live Resume Preparation
- ✓ Live Interview Preparation & Mentorship sessions
- ✓ Coding Questions on all Important Topics (asked by Top Companies)



**400+**

video  
lectures

**300+**

solved questions  
practice

**Duration : 2.5 Months**

Course access is for One Year

# Topics

## Java

| CATEGORY              | CHAPTERS  | OVERVIEW  |
|-----------------------|---|---|
| Basics of Programming | Flowcharts & Pseudocodes<br>Variables & Data Types<br>Conditional Statements<br>Operators | what are flowcharts, pseudocodes, decision making using flowcharts, examples<br>Our first Java program, Variables and data types, Taking input/output, How java code runs?<br>Introduction to if else, else if, Nested conditionals, switch<br>arithmetic, relational, logical & assignment operators |
| Loops & Functions     | For loop, While loop, Do-while loop<br>Patterns<br>Functions                              | For loops, While loops, Do-while loops, Flow of execution of statements, break & continue, examples<br>Introduction to nested loops, basic to advanced patterns solved (butterfly, floyd's triangle, rhombus etc.)<br>Introduction to functions, function calling, Pass by value, scope               |
| Arrays                | Introduction to Arrays<br>Searching & Sorting   | Introduction to arrays, arrays in memory, Passing arrays to functions, interview problems<br>Linear search, Binary search, Selection sort, Bubble sort, Insertion sort, count sort  |
| 2D Arrays & Strings   | 2D Arrays<br>Strings  | 2D arrays, 2D arrays in memory, Examples using 2D Arrays<br>Introduction to strings & StringBuilder, storage of strings and their inbuilt functions   |

# Data Structures & Algorithms (DSA)

| CATEGORY                    | CHAPTERS  | OVERVIEW   |
|-----------------------------|---|--|
| Problem Solving Techniques  | Recursion, Backtracking, Divide & Conquer<br><br>Bit Manipulation<br><br>Time & Space Complexity<br><br>Greedy Algorithms | Introduction to recursion, Principle of mathematical induction, factorial, Fibonacci numbers, Recursion using arrays, Recursion using strings, Recursion using 2D arrays, backtrack, merge sort, quick sort<br><br>Binary number system, bitwise operators, operations on bits, fast exponentiation<br><br>Order complexity analysis, Theoretical complexity analysis, Time complexity analysis of searching and recursive algorithms, Space complexity analysis of merge sort<br><br>Introduction to greedy approach to problem solving, solving classical problems |
| Object-oriented programming | Basic to Advanced OOP   | Objects & Classes, Creating objects, Getters, and setters, Constructors and related concepts, Inbuilt constructor and destructor, Example classes, Static members, Function overloading and related concepts, Abstraction, Encapsulation, Inheritance, Polymorphism, Abstract classes, Interfaces  |
| Linear Data Structures      | ArrayLists<br><br>Linked lists<br><br>Stacks and Queues   | Introduction to java collection framework, arrays, solved questions<br><br>Linked list Introduction, Inserting node in linked list, Deleting node from linked list, Midpoint of linked list, Merge two sorted linked lists, merge sort of a linked list, Reversing a linked list<br><br>Stacks Introduction, Stack using arrays, Dynamic Stack class, Stack using linked list, Inbuilt stack, Queue using arrays, Dynamic queue class, circular queue  |

# Data Structures & Algorithms (DSA)

| CATEGORY                 | CHAPTERS              | OVERVIEW  |
|--------------------------|-----------------------|---|
| Advanced Data Structures | Binary Trees & BST    | Introduction to Binary Trees, Constructing the tree, Binary Tree traversals, Diameter of binary tree, height & LCA of the tree, Introduction to Binary Search Trees, Searching a node in BST, BST class, Inserting and Deleting nodes in BST, Types of balanced BSTs  |
|                          | Heaps/Priority Queues | Introduction to Heaps, Min/Max heaps, Heap Sort, Priority Queues, how to implement priority queues, Introduction to CBT(Complete Binary Trees) and its implementation, Insert and Delete operations in heaps, Implementing priority queues, In-built Priority Queue   |
|                          | Hashing (Maps & Sets) | Introduction to Hashing, Hashmaps, Inbuilt Hashmap, Hashsets, In-built Hashsets, Hash functions, Insert and Delete operation implementation in hashmap/hashset, examples  |
|                          | Tries                 | What are Tries, Creating a Trie node class, Insert, Search and Remove operation in Tries, Types of Tries, Questions on Tries  |
|                          | Graphs                | Introduction to Graphs, Graph Terminology, Graph implementation, Graph Traversals (DFS and BFS), Weighted and Directed Graphs, Minimum Spanning Trees, Cycle Detection in Graphs, Kruskal's algorithm, Prim's Algorithm, Dijkstra's algorithm, Bellman Ford Algorithm & a lot of questions  |
|                          | Segment Trees         | What are segment trees, Creation of segment trees, solving range queries, immutable & mutable   |
| Dynamic Programming      | DP & its Questions    | Fundamentals of Dynamic Programming, Introduction to Memoization, Knapsack using DP, Factorial using DP, Fibonacci numbers using recursion, memoization and tabulation, Longest Common Subsequence (LCS) using recursion, Catalan's number, Edit distance using recursion, memoization and dynamic programming, Matrix Chain Multiplication and much more |

1st Lecture upload will be on **2nd August (9:00 PM)**  
Till then, keep learning & keep exploring ❤