**Intro To Programming**

#### Basics of Programming: - 9 Hours

**FLOWCHARTS**

Introduction to flowcharts, Decision making using flowcharts, Loops, Example problems

**VARIABLES AND DATA TYPES**

First program, Variables and data types, taking input, how data is stored in memory, Arithmetic Operators

**CONDITIONAL STATEMENTS**

Introduction to If else, Relational and logical operators, Nested conditionals

#### Loops and Functions: - 12.5 Hours

**WHILE LOOPS**

While loops, Flow of execution of statements in while loop, Example problems using while loop

**PATTERNS**

Introduction to patterns, Basic Patterns, Square Patterns, Triangular Patterns, Character Patterns, Reverse Triangle, Inverted patterns, Isosceles triangles

**FOR LOOPS**

For loops, Break and Continue, increment - decrement operators

**FUNCTIONS**

Introduction to functions, working of function calling, Variables and its scope, Pass by value

#### Arrays: - 7.5 Hours

**INTRODUCTION TO ARRAYS**

Introduction to arrays, how arrays are stored in memory, Passing arrays to functions

**SEARCHING AND SORTING**

Understanding Binary Search, Selection sort, Bubble sort, Insertion sort, Merging two sorted arrays

#### Strings and 2D Arrays: - 7 Hours

**STRINGS**

Introduction to strings, storage of strings and their inbuilt functions

**2D ARRAYS**

2D arrays, Storage of 2D arrays, Example problems using 2D Arrays

## Basic- DSA = 30 Days

#### Problem Solving Techniques: - 18 Hours = 8 Days

**RECURSION**

Introduction to recursion, Principle of mathematical induction, Fibonacci numbers, Recursion using arrays, Recursion using strings, Recursion using 2D arrays

**TIME AND SPACE COMPLEXITY**

Order complexity analysis, Theoretical complexity analysis, Time complexity analysis of searching and recursive algorithms, Theoretical space complexity, Space complexity analysis of merge sort

#### Object Oriented Programming: - 12 Hours = 5 Days

**BASICS OF OOPS**

Introduction to OOPS, Creating objects, Getters and setters, Constructors and related concepts, Inbuilt constructor and destructor, Example classes

**ADVANCE CONCEPTS OF OOPS**

Static members, Function overloading and related concepts, Abstraction, Encapsulation, Inheritance, Polymorphism, Virtual functions, Abstract classes, Exception handling

#### Linear Data Structures: - 16 Hours = 7 Days

**LINKED LISTS**

Introduction to linked list, 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

**STACKS AND QUEUES**

Introduction to stacks, Stack using arrays, Dynamic Stack class, Stack using linked list, Inbuilt stack, Queue using arrays, Dynamic queue class, Queue using linked list, Inbuilt queue

#### Trees: - 18.5 Hours = 10 Days

**GENERIC TREES**

Introduction to Trees, Making a tree node class, Taking a tree as input and printing, Tree traversals, Destructor for tree node class

**BINARY TREES**

Introduction to Binary Trees, Taking a binary tree as input and printing, Binary Tree traversals, Diameter of binary tree

**BINARY SEARCH TREES**

Introduction to Binary Search Trees, Searching a node in BST, BST class, Inserting and Deleting nodes in BST, Types of balanced BSTs

## Advance – DSA = 25-30 Days

#### Advanced Data Structures: - 32 Hours = 15 Days

**PRIORITY QUEUES**

Introduction to Priority Queues, Ways to implement priority queues, Introduction to heaps, Introduction to Complete Binary Trees and its implementation, Insert and Delete operations in heaps, Implementing priority queues, Heap sort, Inbuilt Priority Queue

**HASHMAPS**

Introduction to HashMap, Inbuilt HashMap, Hash functions, Collision handling, Insert and Delete operation implementation in HashMap, Load factor, Rehashing

**TRIES**

Introduction to Tries, Making a Trie Node class, Insert, Search and Remove operation implementation in Tries, Types of Tries, Huffman Coding

**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

#### Dynamic Programming: - 14 Hours = 7 Days

**INTRODUCTION TO DYNAMIC PROGRAMMING**

Introduction to Memoization, Introduction to Dynamic Programming, Fibonacci numbers using recursion, memoization and dynamic programming

**APPLICATIONS OF DYNAMIC PROGRAMMING**

Longest Common Subsequence (LCS) using recursion, memoization and dynamic programming, Edit distance using recursion, memoization and dynamic programming, Knapsack problem using recursion, memoization and dynamic programming