# Coding Ninjas

# Data Structures
## & Algorithms in C++

"Learning to write programs stretches your mind, and helps you think better, creates a way of thinking about things that I think is helpful in all domains."

**~Bill Gates**

# About Coding Ninjas

At Coding Ninjas, our mission is to continuously innovate the best ways to train the next generation of developers and transform how tech education is delivered. Training is designed and provided by professional developers turned educators who have experience working at bigwigs like Facebook, Amazon, Google etc. and are Stanford, IIT, IIIT alumni.

Coding Ninjas teaches 17+ Programming courses in Foundation, Advanced, Data & Development courses such as Machine Learning, Data Science, Web Development, Android and more.

# Doubt Support

We have developed a very scalable solution using which we are able to solve 4000+ doubts every single day with the help of 500+ doubts on the platform itself with an average rating of 4.8 out of 5.

# Placement Cell

**50,000+**
Students taught
so far

**78%+**
Percentage
placement

**2500+**
Students placed
in top MNCs

**300+** Placement Partners
**7.6L** Average Salary

Number of placement partners and average salary of students

**100+**
Students received
International job offers

## Ankush Singla
### Co-Founder & Instructor

Ankush holds a Bachelor's degree in Computer Science from India's most premier institute- IIT Delhi and a Master's degree in Computer Science from Stanford University.

He is a coding enthusiast and has worked with bigwigs like Amazon and Facebook in the past.

### Live Mentor Support & Student Experience Team

Dedicated TAs and Student experience team to make sure that your doubts get resolved quickly and you don't miss your deadlines.

### Want A Break? Pause Your Course

Take a short break when you need it. Pause your course for upto 60 days. Resume when you are ready

### Get An Industry Recognised Certificate

Get awarded with an industry recognised certificate after you complete your programming course

### Be A Part Of The Learning Community

Slack groups to meet your batchmates. Learn from your peers about resources, doubts and more!

# Programme Overview

O **Course Overview**

You will learn the most commonly used data structures and algorithms that are necessary to solve problems by programming which target multiple job opportunities like software developer, software engineer, product/data roles, etc..

O **Features**

**30K+**
Students Enrolled

**350+**
Problems

**60+**
hours of video content

# WHY
## Data Structures
### & Algorithms?

O It is essential to learn the basics of the most popular programming languages (C++) to become an expert

O Data structures and algorithms is all about organizing theinformation and finding the most efficient approach to solve aproblem.

O Learning these concepts will in turn help you to improve your problem-solving skills and solve any real-world problemsusing technology

# Companies Hiring

amazon | Adobe | Google

Flipkart | PayPal | Microsoft

# Course Outcome

○ This course will enhance your basics of Programming and Data Structures & Algorithm concepts of C++.

# Placement after the course

Ayush Raghuwanshi
L&T Infotech

Abhishek Chauhan
zomato

Vardan Sharma
Flipkart

Vijay Kumar
Cognizant

# Course Offerings

| Basic | Standard | Combo | Premium |
|---|---|---|---|
| Introduction to Programming | x | Introduction to Programming | Introduction to Programming |
| x | Data structures and algorithms | Data structures and algorithms | Data structures and algorithms |
| Dedicated Placement Cell with curated job openings | Dedicated Placement Cell with curated job openings | Dedicated Placement Cell with curated job openings | Dedicated Placement Cell with curated job openings |
| x | x | x | Premium Hirist Account |
| x | x | x | Mock interview/Industry mentor guidance sessions |
| x | x | x | Workshops for building your resume and Linkedin/Github profiles |
| x | x | x | Curated interview problems |
| x | x | x | DSA based product companies Mock Test Series |
| x | x | x | 2 months Free Course Extension |
| **2** Months **15⁺** Hours **10⁺** Assignments **100⁺** Problems | **4** Months **45⁺** Hours **30⁺** Assignments **200⁺** Problems | **6** Months **60⁺** Hours **40⁺** Assignments **300⁺** Problems | **6** Months **60⁺** Hours **40⁺** Assignments **400⁺** Problems |

# Detailed Curriculum

## Introduction to Programming

| Topics | Sub-topics | Details |
|---|---|---|
| Basics of Programming | 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 | 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 | 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 | Strings | Introduction to strings, storage of strings and their inbuilt functions |
| | 2D Arrays | 2D arrays, Storage of 2D arrays, Example problems using 2D Arrays |

# Data Structures

| Topics | Sub-topics | Details |
|---|---|---|
| Problem Solving Techniques | 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 | Basics of OOP | Introduction to oops, Creating objects, Getters, and setters, Constructors and related concepts, Inbuilt constructor and destructor, Example classes |
| | Advance concepts of OOP | Static members, Function overloading and related concepts, Abstraction, Encapsulation, Inheritance, Polymorphism, Virtual functions, Abstract classes, Exception handling |
| Linear Data Structures | 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 | 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 |
| Advanced Data Structures | 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 |

| Topics | Sub-topics | Details |
|---|---|---|
| | Hashmaps | Introduction to Hashmaps, 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 | Introduction to Dynamic Programming | Introduction to Memoization, Introduction to Dynamic Programming, Fibonacci numbers using recursion, memoization and dynamic programming |
| | Applications of Dynamic | 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 |

# CODING NINJAS

**1800-123-3598**

**contact@codingninjas.com**

**codingninjas.com**

**Follow us on**