🚀 COMPLETE C++ + DSA PLACEMENT ROADMAP

Student Name: Utkarsh

Duration: 90 Days (3 Months)

Goal: Placement Ready (Internship + Product Based Companies)

📌 OVERVIEW STRUCTURE

Phase 1 – Complete All Videos (60 Days)

Phase 2 – Advanced Topics (30 Days)

Total Target: 400+ Problems

🎯 PHASE 1 – COMPLETE ALL VIDEOS (60 DAYS)

⏰ DAILY STUDY SCHEDULE (College Days)

Total Study Time: 4–5 Hours Daily

1. 1.5 Hours – Watch New Topic Video
2. 1 Hour – Revise Notes + Code Again Without Seeing
3. 1.5 Hours – Solve Topic Based Problems
4. 30 Minutes – Previous Topic Revision

🗓 WEEK-WISE ROADMAP

🔹 WEEK 1–2: C++ FUNDAMENTALS

Topics:

- Basics (Variables, Data Types)
- Loops
- Conditionals
- Functions
- Pointers
- Arrays
- Strings

Daily Practice:

- 5 Easy Problems

Platforms:

- LeetCode (Easy)
- HackerRank
- CodeStudio

🎯 **Target: 60 Problems**

🔹 WEEK 3: SEARCHING + SORTING

Topics:

- Linear Search

- Binary Search
- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort

Daily Practice:

- 6–7 Problems

Platforms:

- LeetCode Easy
- CodeStudio

🎯 **Target: 40 Problems**

🔹 **WEEK 4: RECURSION + TIME COMPLEXITY**

Topics:

- Basic Recursion
- Recursion on Arrays
- Backtracking Basics
- Time Complexity Analysis

Daily Practice:

- 5 Recursion Problems
- 2 Time Complexity Analysis

Platforms:

- LeetCode
- GFG

🎯 **Target: 35 Problems**

🔹 **WEEK 5: LINKED LIST + STACK + QUEUE**

Topics:

- Singly Linked List
- Doubly Linked List
- Stack (Array + Linked List)
- Queue
- Circular Queue

Daily Practice:

- 6 Problems

Platforms:

- LeetCode (Filter by Topic)

🎯 **Target: 45 Problems**

🔹 **WEEK 6: TREES + BST**

Topics:

- Tree Traversals
- Height / Diameter
- Binary Search Tree
- Level Order
- LCA

Daily Practice:

- 5 Tree Problems

Platform:

- LeetCode (Tree Section)

🎯 **Target: 40 Problems**

🔹 **WEEK 7: HEAP + MAP + SET + GREEDY**

Topics:

- Priority Queue
- Heap
- HashMap
- HashSet
- Greedy Algorithms

Daily Practice:

- 6 Problems

Platforms:

- LeetCode
- GFG

🎯 **Target: 40 Problems**

🔹 **WEEK 8: GRAPH + BACKTRACKING + DP (BASIC)**

Topics:

- BFS
- DFS
- Cycle Detection
- Basic Dynamic Programming
- Backtracking

Daily Practice:

- 4 Medium Problems

Platforms:

- LeetCode Medium
- GFG

🎯 **Target: 35 Problems**

📊 **TOTAL TARGET AFTER 60 DAYS**

Easy: 150

Medium: 120

Hard: 20

Total: 290+ Problems

✅ At this stage, you are placement ready for most companies.

🚀 **PHASE 2 – ADVANCED TOPICS (30 DAYS)**

Advanced Graph

- Dijkstra
- Bellman Ford
- Floyd Warshall
- MST (Kruskal, Prim)

Target: 20 Problems

Advanced Dynamic Programming

- LIS Variations
- Knapsack Variations
- DP on Trees
- Partition DP

Target: 30 Problems

Segment Tree + Fenwick Tree

Target: 15 Problems

Disjoint Set (Union Find)

Target: 10 Problems

Bit Manipulation

Target: 20 Problems

🗓️ **Daily (Advanced Phase)**

- 3 Medium Problems
- 1 Hard Problem (Alternate Days)

Target: 120 More Problems

💻 **BEST PRACTICE PLATFORMS**

LeetCode – Best for Placements

CodeStudio – Structured DSA Sheets

GeeksforGeeks – Concept Clarity
Codeforces – Competitive Improvement

🔥 **DAILY PROBLEM RULE**

Easy Topic → 5–6 Problems
Medium Topic → 3–4 Problems
Hard Topic → 1 (Alternate Days)

🧠 **GOLDEN RULES**

Never:

- Watch video without solving
- Skip revision
- Jump topics randomly

Always:

- Code yourself
- Revise weekly
- Maintain notebook
- Track solved problems

🎯 **FINAL STRUCTURE (0 → Placement Ready)**

Step 1 – Complete Videos (60 Days)
Step 2 – Solve 300 Problems
Step 3 – Advanced Topics (30 Days)
Step 4 – Mock Interviews
Step 5 – Resume + Core Subjects Revision

📌 **FINAL TARGET**

Total Problems: 400+
Consistency: 90 Days
Outcome: Internship + Placement Ready

COMPLETE DSA C++ COURSE STRUCTURE

Total Videos: 76+

Sequence: Foundation → Intermediate → Advanced

Goal: Placement-Level Mastery

✅ **SECTION 1: C++ FUNDAMENTALS (Programming Basics)**

1. C++ Input & Operators (Lecture 2) – 1:27:47
2. Fundamentals of Programming – Session (One Shot) – 1:48:46
3. Pattern Printing – Part 1 – 1:39:56
4. Pattern Printing – Part 2 – 1:36:24
5. Loops – Part 1 – 1:37:10
6. Loops – Part 2 – 2:05:11
7. Conditionals – Part 1 – 2:24:19
8. Conditionals – Part 2 – 1:12:38

✅ **SECTION 2: FUNCTIONS & POINTERS**

9. Functions – 2:31:53
10. Pointers – 1:55:01

✅ **SECTION 3: ARRAYS**

11. Arrays – Part 1 – 1:36:00
12. Arrays – Part 2 – 2:33:28
13. Arrays – Part 3 – 3:31:49

✅ **SECTION 4: 2D ARRAYS**

14. 2D Arrays – Part 1 – 1:55:56
15. 2D Arrays – Part 2 – 1:51:49
16. 2D Arrays – Part 3 – 2:10:48

✅ **SECTION 5: STRINGS**

17. Strings – Part 1 – 1:14:34
18. Strings – Part 2 – 1:59:46

✅ **SECTION 6: TIME & SPACE COMPLEXITY**

19. Time & Space Complexity – Part 1 – 2:01:37
20. Time & Space Complexity – Part 2 – 1:10:07

✅ **SECTION 7: SEARCHING**

21. Searching – Part 1 – 1:50:31
22. Searching – Part 2 – 2:38:48

23. Searching – Part 3 – 1:52:24

✅ **SECTION 8: SORTING**

24. Bubble Sort – 1:14:46
25. Selection Sort & Insertion Sort – 1:17:55
26. Problems on Sorting – 1:50:13
27. Merge Sort – 2:09:12
28. Quick Sort – 2:04:09
29. Cyclic Sort – 1:28:58

✅ **SECTION 9: RECURSION**

30. Recursion – Part 1 – 1:47:38
31. Recursion – Part 2 – 2:31:15
32. Recursion – Part 3 – 3:01:23
33. Recursion – Part 4 – 4:28:14
34. Recursion – Session (One Shot) – 1:40:41

✅ **DATA STRUCTURES SECTION**

🔹 **LINKED LIST**

35. Linked List – Part 1 – 3:57:05
36. Linked List – Part 2 – 2:35:03
37. Linked List – Part 3 – 3:52:29
38. Linked List – Part 4 – 4:05:53

🔹 **STACKS**

39. Stacks – Part 1 – 2:02:29
40. Stacks – Part 2 – 2:07:49
41. Stacks – Part 3 – 1:58:07
42. Stacks – Part 4 – 1:53:51

🔹 **QUEUES**

43. Queues – Part 1 – 2:02:14
44. Queues – Part 2 – 1:21:28
45. Queues – Part 3 – 1:36:14

🔹 **BINARY TREES**

46. Binary Trees – Part 1 – 2:18:31
47. Binary Trees – Part 2 – 2:29:34
48. Binary Trees – Part 3 – 2:25:09

🔹 **BINARY SEARCH TREES**

49. Binary Search Trees – Part 1 – 2:03:46
50. Binary Search Trees – Part 2 – 55:08
51. Binary Search Trees – Part 3 – 1:27:48

🔹 **HEAPS**

52. Heaps – Part 1 – 3:11:03
53. Heaps – Part 2 – 3:09:52
54. Heaps – Part 3 – 1:40:29

🔹 **MAPS & SETS**

55. Maps & Sets – Part 1 – 1:33:43
56. Maps & Sets – Part 2 – 2:08:45
57. Maps & Sets – Part 3 – 2:40:02

✅ **ADVANCED SECTION**

🔹 **GREEDY**

58. Greedy – Part 1 – 2:45:12
59. Greedy – Part 2 – 1:07:33
60. Greedy – Part 3 – 2:16:46

🔹 **BACKTRACKING**

61. Backtracking – Part 1 – 2:25:17
62. Backtracking – Part 2 – 1:41:51

🔹 **GRAPHS**

63. Introduction to Graphs – 2:22:53
64. Graphs – Part 1 – 40:34
65. Graphs – Part 2 – 1:48:11
66. Graphs – Part 3 – 2:23:14
67. Graphs – Part 4 – 2:37:27
68. Graphs – Part 5 – 1:01:50
69. Problem on Graphs – Part 1 – 2:43:10
70. Problem on Graphs – Part 2 – 2:04:04

🔹 **DYNAMIC PROGRAMMING**

71. Dynamic Programming – Part 1 – 2:20:24
72. Dynamic Programming – Part 2 – 1:48:54

73. Dynamic Programming – Part 3 – 2:25:47
74. Dynamic Programming – Part 4 – 5:04:10
75. Dynamic Programming – Part 5 – 2:19:54
76. Dynamic Programming – Part 6 – 1:30:00

📊 COURSE SUMMARY

Total Videos: 76+
Coverage Level: Beginner → Advanced
Topics Covered: Complete DSA Foundation + Advanced

🚀 PHASE 2 – ADVANCED DSA ROADMAP

(Topics Not Covered in Your DSA C++ Course)
Duration: 45 Days
Goal: Product-Based Company Level
Level After Completion: Strong Placement Ready

🗓️ DAILY STUDY STRUCTURE (PHASE 2)

Total Study Time: 3–4 Hours Daily
• 1 Hour – Learn Concept
• 2 Hours – Solve Problems
• 30 Minutes – Revision
Daily Problem Target:

- 3 Medium Problems
- 1 Hard Problem (Every 2 Days)

Platforms:

- LeetCode
- GeeksforGeeks (GFG)
- CodeStudio

📘 WEEK 1 – BIT MANIPULATION

Topics Covered:

1. XOR Basics
2. Power of Two
3. Single Number Problems
4. Subsets Using Bitmask
5. Important Bit Tricks

Target:

Solve 25 Problems
(From LeetCode Bit Manipulation Section)

📘 WEEK 2 – DISJOINT SET (UNION FIND)

Topics Covered:

1. Union by Rank
2. Path Compression
3. Cycle Detection
4. Connected Components

Target:

Solve 15–20 Problems
(Graph + DSU Based Problems)

📘 WEEK 3 – TRIE (Important for Interviews)

Topics Covered:

1. Insert / Search
2. Prefix Matching
3. Word Search
4. Replace Words

Target:

Solve 15 Problems

📘 WEEK 4 – SEGMENT TREE + FENWICK TREE

Topics Covered:

1. Segment Tree Basics
2. Range Sum Query
3. Lazy Propagation
4. Fenwick Tree (Binary Indexed Tree)

Target:

Solve 20 Problems

📘 WEEK 5 – ADVANCED GRAPH

Topics Covered:

1. Dijkstra Algorithm
2. Bellman-Ford Algorithm
3. Floyd-Warshall Algorithm
4. Kruskal's Algorithm (Minimum Spanning Tree)
5. Prim's Algorithm
6. Advanced Topological Sort
7. Strongly Connected Components

Target:
Solve 25 Problems

📘 WEEK 6 – ADVANCED DYNAMIC PROGRAMMING

Topics Covered:

1. LIS Variations
2. DP on Trees
3. Bitmask DP
4. Digit DP
5. Matrix Chain Multiplication
6. DP Optimization Techniques

Target:
Solve 30 Problems

📊 TOTAL PHASE 2 TARGET

Bit Manipulation – 25
Disjoint Set – 20
Trie – 15
Segment Tree – 20
Advanced Graph – 25
Advanced DP – 30
Total: 135 Problems

🏆 FINAL RESULT AFTER BOTH PHASES

Phase 1 Problems: ~300
Phase 2 Problems: ~135
Total Problems Solved: 435+
This is strong Product-Level preparation.

💻 PLATFORM STRATEGY

LeetCode → Interview Focus
CodeStudio → Structured Practice
GeeksforGeeks → Concept Clarity
Codeforces → Speed & Logical Thinking

📌 OPTIONAL (Elite Level Topics)

If you want to go beyond standard placement level, add:
• Meet in the Middle
• Mo's Algorithm
• Binary Search on Answer

- Monotonic Stack
- Advanced Sliding Window

🎯 FINAL STATUS AFTER COMPLETION

✓ Strong in Core DSA

✓ Confident in Medium & Hard Problems

✓ Ready for Product-Based Interviews

✓ 400+ Problems Solved

✓ Strong Algorithmic Thinking