🔥 30-Day C++ Learning Plan

📌 Week 1: C++ Basics (Syntax & Fundamentals)

Focus: Understand C++ syntax, I/O operations, and basic programming concepts.

🗓️ Day 1-2: Introduction & Setup

- Install C++ compiler (GCC or MSVC)
- Setup VS Code or CodeBlocks
- Write your first C++ program (Hello World)
- Basic Input/Output (cin, cout)
- Variables & Data Types (int, char, float, double, bool)

🗓️ Day 3: Operators & Conditionals

- Arithmetic, Logical, and Relational operators
- If-else, Switch-case statements

🗓️ Day 4: Loops & Functions

- For, While, and Do-While loops
- Function declaration, recursion

🗓️ Day 5: Arrays & Strings

- 1D and 2D arrays
- String functions (getline(), substr(), find(), stoi())

🗓️ Day 6: Pointers & Memory Management

- Introduction to pointers
- Dynamic memory allocation (new, delete)

🗓️ Day 7: Reference Variables & STL Intro

- Understanding references
- Introduction to Standard Template Library (STL)

📌 Week 2: Object-Oriented Programming (OOP)

Focus: Get comfortable with OOP concepts, as it's crucial for coding interviews.

🗓️ **Day 8: Classes & Objects**

- Defining classes & objects
- Access specifiers (public, private, protected)

🗓️ **Day 9: Constructors & Destructors**

- Parameterized and default constructors
- Destructor

🗓️ **Day 10: Inheritance & Polymorphism**

- Single, Multiple, and Multilevel inheritance
- Function overloading & overriding

🗓️ **Day 11: Encapsulation & Abstraction**

- Getters & Setters
- Abstract classes

🗓️ **Day 12: Operator Overloading**

- Overloading +, -, ==, etc.

🗓️ **Day 13: File Handling in C++**

- Read/Write operations using fstream

🗓️ **Day 14: Exception Handling**

- Try-catch blocks
- Handling runtime errors

📌 **Week 3: STL & Advanced Concepts**

Focus: Learn STL, which is essential for competitive programming and DSA.

🗓️ **Day 15: Introduction to STL**

- Containers, Algorithms, and Iterators

🗓️ **Day 16: Vectors & Pairs**

- vector<int>, vector<string>, pair<int, int>

🗓️ **Day 17: Sets & Maps**

- set, unordered_set, map, unordered_map

🗓️ **Day 18: Stacks & Queues**

- stack, queue, priority_queue

🗓️ **Day 19: Linked List (Singly & Doubly)**

- Creating a linked list
- Insertion, Deletion, Traversal

🗓️ **Day 20: Recursion & Backtracking**

- Basics of recursion
- Generating permutations

🗓️ **Day 21: Greedy & Sorting Algorithms**

- Bubble Sort, Selection Sort, Insertion Sort
- Quick Sort, Merge Sort

📌 **Week 4: Practice & Problem Solving**

Focus: Apply what you learned by solving problems on LeetCode & CodeChef.

🗓️ **Day 22: Array & String Problems**

- Two Sum
- Reverse a String
- Maximum Subarray (Kadane's Algorithm)

🗓️ **Day 23: Linked List Problems**

- Reverse a Linked List
- Detect Cycle in a Linked List

🗓️ **Day 24: Stack & Queue Problems**

- Implement Stack using Queue
- Implement Queue using Stack

🗓️ **Day 25: Tree & Graph Basics**

- BFS & DFS Traversal

🗓️ **Day 26: DP & Recursion Problems**

- Fibonacci using DP

- Coin Change Problem

🗓️ **Day 27: STL-Based Problems**

- Most Frequent Element in an Array
- Sorting Using Comparator

🗓️ **Day 28-30: Mock Contests & LeetCode Practice**

- Solve 5 problems/day from easy-medium level
- Participate in CodeChef or LeetCode contests

🎯 **After 30 Days:**

✅ You'll be ready to start **DSA & LeetCode** with a strong C++ foundation.
✅ You'll be familiar with **STL**, recursion, OOP, and problem-solving techniques.

Would you like me to suggest specific LeetCode problems for DSA once you finish C++? 🚀

🔗 **Connect with Me**

GitHub: https://github.com/iankushsingh