## **C++ Roadmap: Beginner to Advanced**

### **📌 Phase 1: Fundamentals (Beginner Level)**

🔹 **1. Introduction to C++** ✅ (Completed)

* History, features, and applications of C++
* Setting up the environment
* Writing and compiling your first program

🔹 **2. Basic Syntax & Data Types**

* Variables, constants, and keywords
* Data types: int, float, char, bool, etc.
* Input/output operations (cin, cout)

🔹 **3. Operators & Expressions**

* Arithmetic, relational, logical, bitwise, assignment, and ternary operators
* Operator precedence and associativity

🔹 **4. Control Flow (Decision Making & Loops)**

* if, if-else, switch-case
* Loops: for, while, do-while
* Break, continue, and nested loops

🔹 **5. Functions & Scope**

* Function definition, return types, and parameters
* Function overloading & default arguments
* Scope, lifetime, and storage classes (auto, static, extern, register)

### **📌 Phase 2: Intermediate Concepts**

🔹 **6. Arrays & Strings**

* One-dimensional & multi-dimensional arrays
* String manipulation and string class

🔹 **7. Pointers & Dynamic Memory Allocation**

* Pointers: Basics, arithmetic, and pointer-to-pointer
* new and delete operators
* Dynamic arrays and memory leaks

🔹 **8. Structures & Unions**

* Defining and using struct and union
* Difference between structure and class

🔹 **9. Object-Oriented Programming (OOP)**

* Classes and objects
* Access specifiers: public, private, protected
* Constructors and destructors
* this pointer

🔹 **10. OOP Concepts - Advanced**

* Inheritance: Single, multiple, multilevel, hierarchical, hybrid
* Polymorphism: Function overloading & overriding
* Abstract classes & interfaces (virtual functions)

### **📌 Phase 3: Advanced Topics**

🔹 **11. Standard Template Library (STL)**

* Introduction to STL
* Containers: Vector, List, Stack, Queue, Map, Set
* Iterators and algorithms

🔹 **12. File Handling in C++**

* Reading & writing files using fstream
* Handling binary files

🔹 **13. Exception Handling**

* Try, catch, and throw statements
* Custom exception classes

🔹 **14. Multi-threading & Concurrency**

* Basics of multi-threading
* Mutex and race conditions

🔹 **15. Advanced Concepts**

* Smart pointers (unique\_ptr, shared\_ptr, weak\_ptr)
* Move semantics & rvalue references
* Lambda expressions

### **📌 Phase 4: Mastery & Real-World Applications**

🔹 **16. Data Structures & Algorithms in C++**

* Linked lists, stacks, queues, trees, graphs
* Sorting & searching algorithms

🔹 **17. Competitive Programming in C++**

* Efficient coding techniques
* Solving problems on Leetcode, Codeforces, etc.

🔹 **18. System Programming & Embedded C++**

* Low-level programming with memory management
* Embedded systems concepts

🔹 **19. Game Development & Graphics**

* Introduction to SFML, SDL, or OpenGL
* Basics of game loops and rendering

🔹 **20. Building Real-World C++ Projects**

* Console-based projects
* GUI-based applications
* Full-stack C++ applications (backend systems)

### **📌 Learning Resources & Practice**

✅ **Books**: "The C++ Programming Language" by Bjarne Stroustrup, "Effective C++" by Scott Meyers  
✅ **Online Courses**: Udemy, Coursera, Codecademy  
✅ **Practice Websites**: Leetcode, Codeforces, HackerRank, GeeksforGeeks  
✅ **Project-Based Learning**: GitHub, Open-source contributions