# Mastering C++20

1. **Introduction to C++**
   * Overview of C++ Programming Language
   * Setting Up the Development Environment
   * Writing Your First Program
   * Compiling and Running C++ Code

* **Basic Syntax and Structure**
  + Data Types and Variables
  + Input and Output
  + Operators and Expressions
  + Control Structures (if, else, switch, loops)
* **Functions in C++**
  + Function Declarations and Definitions
  + Function Parameters and Return Types
  + Pass by Value vs Pass by Reference
  + Function Overloading
  + Lambda Functions (C++11/14/17/20 Enhancements)
* **Object-Oriented Programming (OOP)**
  + Classes and Objects
  + Constructors and Destructors
  + Encapsulation, Inheritance, and Polymorphism
  + Access Specifiers (public, private, protected)
  + Member Functions and Data Members
  + Static Members
  + Friend Functions and Classes
* **Memory Management**
  + Stack vs Heap Memory
  + Dynamic Memory Allocation (new, delete)
  + Smart Pointers (unique\_ptr, shared\_ptr, weak\_ptr)
* **C++ Standard Library (STL)**
  + Overview of the STL
  + Containers (vector, list, map, set, unordered\_map, etc.)
  + Iterators
  + Algorithms
  + Function Objects and Lambdas in Algorithms
* **Templates and Generic Programming**
  + Function Templates
  + Class Templates
  + Template Specialization
  + Concepts (C++20)
* **Advanced Object-Oriented Concepts**
  + Virtual Functions and Polymorphism
  + Abstract Classes
  + Interfaces in C++
  + Multiple Inheritance
  + Copy Constructors, Assignment Operators, and the Rule of 5
* **Error Handling**
  + Exceptions
  + Try, Catch, and Throw
  + Standard Exception Classes
  + noexcept Specifier (C++11)

* **Modern C++ Features (C++11 - C++20)**
  + Move Semantics and Rvalue References (C++11)
  + constexpr and consteval Functions (C++11/20)
  + Range-based Loops (C++11)
  + std::optional and std::variant (C++17)
  + Concepts (C++20)
  + Ranges (C++20)
  + Coroutines (C++20)
  + Modules (C++20)
  + Three-way Comparison (Spaceship Operator)
* **Multithreading and Concurrency**
  + Introduction to Multithreading
  + std::thread and std::async
  + Mutexes and Locks (std::mutex, std::unique\_lock)
  + Condition Variables
  + Atomic Operations
  + Coroutines (C++20)
  + Parallel Algorithms (C++17/20)
* **File Handling**
  + File Streams (ifstream, ofstream, fstream)
  + Reading and Writing Files
  + Working with Binary Files
* **C++20 New Features**
  + Concepts and Constraints
  + Ranges and Views
  + Coroutines
  + Modules
  + Calendar and Time Zones (Chrono library updates)
  + Improved constexpr
* **C++ Best Practices**
  + Code Readability and Formatting
  + Writing Efficient Code (Time and Space Complexity)
  + Memory Management Best Practices
  + Using the Standard Library Effectively
  + Avoiding Undefined Behavior
* **Debugging and Testing**
  + Debugging Techniques (gdb, lldb)
  + Unit Testing with Google Test
  + Performance Profiling Tools
* **Advanced C++ Techniques**
  + Metaprogramming
  + SFINAE (Substitution Failure Is Not An Error)
  + Template Metaprogramming
  + Variadic Templates
  + Advanced Template Patterns (CRTP, Tag Dispatching)

* **Design Patterns in C++**
  + Singleton
  + Factory
  + Observer
  + Strategy
  + Visitor
* **Optimization Techniques**
  + Compiler Optimizations (Flags and Options)
  + Manual Optimizations (Loop Unrolling, Inlining)
  + Memory and Cache Optimization
  + Using Profiling Tools for Optimization
* **Project Structure and Build Systems**
  + Organizing Code for Large Projects
  + CMake Basics
  + C++ Modules (C++20)
  + Managing Dependencies
* **Working with External Libraries**
  + Using Third-Party Libraries
  + Linking and Dependencies
  + Header-Only Libraries vs Compiled Libraries
* **C++ in Real-World Applications**
  + C++ for Game Development (Introduction to SDL, SFML)
  + C++ in Systems Programming
  + C++ in Networking and Web Applications (Boost, ASIO)
  + C++ for Embedded Systems
* **Becoming a C++ Hero**
  + Contributing to Open Source Projects
  + Staying Updated with C++ Standards
  + Advanced C++ Libraries (Boost, Eigen, Poco)
  + Building Cross-Platform Applications

————————

This table of contents guides a learner from beginner-level topics such as basic syntax to expert-level subjects like metaprogramming and optimization techniques, with a focus on C++20 features.

#software/languages/cpp