C++ COURSE OUTLINE

Learning C++ can be divided into several phases, starting from the basics and moving towards advanced topics. Here's a structured list of topics and subtopics you should cover to gain a comprehensive understanding of C++:

1. Introduction to C++

- Basic Syntax

- Structure of a C++ program
- Comments
- Data types
- Variables and Constants
- Input and Output (cin, cout)

- Operators

- Arithmetic operators
- Relational operators
- Logical operators
- Bitwise operators
- Assignment operators
- Miscellaneous operators (sizeof, comma, etc.)

2. Control Structures

-Conditional Statements

- if, if-else, nested if-else
- switch-case

- Loops

- for loop
- while loop
- do-while loop

- Jump Statements

- break

- continue
- goto

3. Functions

- Function Basics

- Function declaration and definition
- Function call
- Function arguments and return value

- Advanced Functions

- Default arguments
- Function overloading
- Inline functions
- Recursive functions

4. Arrays and Strings

- Arrays

- One-dimensional arrays
- Multi-dimensional arrays

- Strings

- Character arrays
- String class

5. Pointers and References

- Pointers

- Pointer basics
- Pointer arithmetic
- Pointers and arrays
- Pointers to pointers

- References

- Reference basics

- Passing by reference
- Reference vs Pointer

6. Dynamic Memory Management

- Memory Allocation

- new and delete operators
- Dynamic arrays

- Smart Pointers (C++11)

- unique_ptr
- shared_ptr
- weak_ptr

7. Object-Oriented Programming (OOP)

- Classes and Objects

- Class definition
- Access specifiers (public, private, protected)
- Member functions
- Constructors and Destructors

- Advanced OOP

- Inheritance (Single, Multiple, Multilevel, Hierarchical)
- Polymorphism
  - Function overloading
  - Operator overloading
  - Virtual functions and pure virtual functions
- Encapsulation
- Abstraction

8. Advanced Features

- Exception Handling

- try, catch, throw

- Templates

- Function templates
- Class templates

- Standard Template Library (STL)

- Containers (vector, list, map, set, etc.)
- Iterators
- Algorithms (sort, find, etc.)
- Function objects

9. File Handling

- File Operations

- File streams (ifstream, ofstream, fstream)
- Reading from and writing to files
- File modes (ios::in, ios::out, ios::app, etc.)

10. Advanced C++ Concepts

- Multithreading (C++11)

- std::thread
- Mutexes and locks
- Condition variables

- Networking (using libraries like Boost)

- Basics of sockets
- TCP/IP and UDP communication

11. Best Practices and Design Patterns

- Best Practices

- Code organization and modularization
- Documentation and commenting
- Code optimization techniques

- Design Patterns

- Singleton

  - Factory
  - Observer
  - Strategy
  - Decorator

12. Modern C++ (C++11 and beyond)

- **New Features**
  - auto keyword
  - Lambda expressions
  - Range-based for loop
  - nullptr
  - std::move and rvalue references
  - constexpr
  - Variadic templates
  - std::chrono (date and time)

Learning Resources

- **Books**
  - "The C++ Programming Language" by Bjarne Stroustrup
  - "Effective C++" by Scott Meyers
  - "C++ Primer" by Stanley B. Lippman
- **Online Courses**
  - Coursera, Udemy, edX
- **Documentation and References**
  - [C++ Reference](http://www.cplusplus.com)
  - [ISO C++](https://isocpp.org)

Practical Experience

- **Projects**
  - Build small projects to apply what you learn

- Contribute to open-source projects

- Coding Practice

- Solve problems on platforms like LeetCode, HackerRank, Codeforces