C++ for Beginners: Learn to Code Like a Pro

What is C?

- C is a structural or procedural oriented programming language which is machine-independent.
- can be used to develop from the operating systems
- developed by Dennis Ritchie at the Bell Laboratories

What is C++?

- c++ is a special-purpose programming language developed by **Bjarne Stroustrup** at Bell Labs
- very similar to C language
- it is so compatible with C that it can run 99% of C programs without changing any source of code

C	C++
procedural style programming	both **procedural and object oriented**
Data is less secured	you can use modifiers for class members to make it inaccessible for outside user
follows the top-down approach	follows the **bottom-up approach**
does not support function overloading	C++ supports function overloading
scanf() and printf() are mainly used for input/output	**cin and cout** to perform input and output operations
Operator overloading is not possible	Operator overloading is possible
C does not provide the feature of namespace	C++ supports the feature of namespace
Exception handling is not easy in C. It has to perform using other functions.	C++ provides exception handling using Try and Catch block.

C++ Features

- Simple
- Machine Independent or Portable
- Rich Library
- Memory Management
- Object-Oriented
- Errors are easily detected
- Modeling Real-World Problems
- Standard Template Library (STL)
- Problem Solving Flexibility

C++ Applications

- Systems Programming
- Game Development
- Embedded Systems
- High-Performance Applications
- Graphics and Multimedia
- Financial Software
- Networking
- Library Development

A general breakdown of a basic C++ code structure

Preprocessor Directives:

- `#include`**: Used to include libraries or header files.**
- `#define`**: Used for defining constants or macros.**

Namespace Declaration:

- `using namespace std;`**: Often used to avoid typing** `std::` **before standard library objects.**

Main Function:

- `int main()`: **Entry point of the program.**
- **Contains statements that define the behavior of the program.**

Variables:

- Declaration: `int x;`
- **Initialization:** `int x = 10;`

Data Types:

- Basic types: `int`, `float`, `double`, `char`, **etc.**
- **Derived types:** `array`, `pointer`, `reference`, `structure`, `union`, `enumeration`.

Operators:

- Arithmetic: `+`, `-`, `*`, `/`, `%`.
- **Comparison:** `==`, `!=`, `>`, `<`, `>=`, `<=`.
- **Logical:** `&&`, `||`, `!`.

Control Structures:

- Conditional Statements: `if`, `else if`, `else`.
- **Looping Constructs:** `for`, `while`, `do-while`.
- **Switch Case:** `switch`, `case`, `break`, `default`.

Functions:

- Declaration: `return_type function_name(parameters);`
- **Definition:** `return_type function_name(parameters) { // function body }`
- **Recursive functions are also supported.**

Input/Output:

- Standard input/output: `cin`, `cout`.
- **File input/output:** `ifstream`, `ofstream`.

Comments:

- Single-line comments: `//`.
- **Multi-line comments:** `/* */`.

NOTE : `iostream` **is a standard header file that provides input and output functionality. It stands for "input-output stream." The** `iostream` **header includes declarations for** `cin`, `cout`, `cerr`, `clog`, **which are used for standard input (keyboard) and output (console).**

Difference between initialise, declare, define

Declare:

- Meaning: Declaring a variable or function means announcing its existence to the compiler. It tells the compiler about the name and type of the variable or function but does not allocate memory for variables or provide the implementation for functions.
- Syntax:
    - Variable Declaration: `int x;`
    - **Function Declaration:** `int add(int a, int b);`
- Usage: Declarations are often seen in header files (.h or .hpp) or at the beginning of source files (.cpp) to let the compiler know about the entities that will be used or defined later in the code.

Define:

- Meaning: Defining a variable or function means allocating memory for variables or providing the implementation for functions. It includes both declaration and initialization.
- Syntax:
    - Variable Definition: `int x = 10;`
    - **Function Definition:** `int add(int a, int b) { return a + b; }`
- Usage: Definitions are typically found in source files (.cpp) and provide the actual implementation of variables or functions declared elsewhere.

Initialize:

- Meaning: Initializing a variable means assigning it a value at the time of declaration or later in the code. Initialization happens after the variable has been declared and, optionally, defined.
- Syntax:
    - Variable Initialization at Declaration: `int x = 10;`
    - **Variable Initialization Later:** `x = 20;`
- Usage: Initialization sets the initial value of a variable, either at the time of declaration or later in the program.

- Declaration is about announcing the existence of an entity to the compiler.
- Definition includes both declaration and provision of storage or implementation for the entity.

- Initialization assigns an initial value to a declared variable. It can happen at the time of declaration or later in the code.

Ternary Operator

The ternary operator, also known as the conditional operator, is a shorthand way of writing an if-else statement in a single line.a condition followed by a question mark (?), an expression to evaluate if the condition is true, and a colon (:) followed by an expression to evaluate if the condition is false

condition ? expression1 : expression2