

Share on your Social
Media



C and C++ Tutorial

Published On: August 1, 2024

C and C++ Tutorial

C is a high-level, procedural, general-purpose programming language. Whereas C++, a general-purpose programming language, was created as an improvement on C to incorporate the object-oriented paradigm. Learn C and C++ tutorials from scratch to begin your software development career.

[Download C and C++ Tutorial PDF](#)

Introduction to C and C++

Developers have utilized the general-purpose programming language C extensively. Because of its great capabilities, C has been used to create databases, operating systems, apps, and more.

Meanwhile, cross-platform programming languages like C++ can be utilized to develop high-performance software. Bjarne Stroustrup created C++ as a language extension for C. Graphical user interfaces, embedded systems, and operating systems are all made with C++.

In this C and C++ tutorial, we cover the following:

- Overview of C and C++
- Variables and Consonants in C and C++
- Data Types in C and C++
- Operators in C and C++

Featured Articles



Want to know
more about
becoming an
expert in IT?

[Click Here to Get Started](#)

100%
Placement
Assurance

AUTHORISED
CERTIFICATION
PARTNER

IBT

Related Courses at SLA

Related Posts



C and C++ Tutorial

Published On: August 1, 2024

C and C++ Tutorial C is a high-level, procedural, general-purpose programming language. Whereas C++, a...



Quick Enquiry

C and C++ Interview Questions

Overview of C and C++

The C programming language is renowned for its effectiveness and simplicity. Since it provides you with a basic understanding of programming, it is the ideal option for you to begin with.

You can easily learn other languages, such as Java, C++, C#, Python, etc., with the aid of C. Compared to Python and Java, C is a faster programming language.

Low-level programming is supported, and the C code can be compiled for a range of computer systems. Here is a summary of some of the main benefits of C:

- Simple to pick up.
- Language with versatility that works with a variety of technologies and applications.
- A language for mid-level programming.
- Programming Language with Structure.

Hello World with C

```
#include <stdio.h>

int main() {

printf("Hello World! I Don't Give a Bug");

return 0;

}
```

Features of C Programming Language

A few key features of the C language demonstrate its strength and capability:

- Simplicity and Efficiency
- Fast Speed
- Portable
- Memory Management
- Pointers
- Structured Language



ASP DOTNET Tutorial

Published On: July 31, 2024

ASP DOTNET Tutorial Microsoft created the web framework known as ASP.NET. It is employed in...



Artificial Intelligence Tutorial

Published On: July 30, 2024

Artificial Intelligence Tutorial Artificial intelligence (AI) is significant since it enhances many facets of society...



Appium Testing Tutorial

Published On: July 30, 2024

Appium Testing Tutorial Designed to make the UI automation of many app platforms easier, Appium...

Applications of C Language

C code executes just as quickly as assembly language code, it was once referred to as a system development language. The following is how C is used:

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities

C++

C++ is the most widely used and well-liked programming language. It is an object-oriented programming language that incorporates all of the OOPs ideas, including inheritance, encapsulation, and abstraction, giving programs a clear structure and enabling code reuse that reduces development costs and increases security.

It is platform-neutral and can be used to develop programs that are cross-platform compatible.

You can choose to study C++ as your first programming language because it's a simple language to learn.

Programmers can easily transition to C++ due to its syntax's resemblance to that of C, Java, and C#.

Features of C++

The general-purpose programming language C++ was created as an object-oriented paradigm addition to the C language. It is a compiled

language that is imperative.

Among the features of C++ are the following:

- Simple and Popular
- High-Level Language
- Object-Oriented Programming
- Machine Independent
- Case-sensitive
- Compiler Based
- Dynamic Memory Allocation
- Memory Management
- Multi-threading

Hello World with C++

```
#include <iostream>

int main() {

    std::cout << "Hello, World!" << std::endl;

    return 0;

}
```

Similarities Between C and C++

- The syntax of the two languages is comparable.
- Both languages have the same code structure.
- The two languages are comparable in terms of compilation.
- Their fundamental syntax is the same. C++ contains almost all of the operators and keywords found in C, and they all perform the same function.
- The underlying syntax of C++ is the same as that of C; however, it is slightly enhanced.
- Both have extremely similar basic memory models for the hardware.
- Both languages use the same concepts for stack, heap, file-scope, and static variables.

Difference Between C and C++

C	C++

C does not enable inheritance, encapsulation, or polymorphism, object-oriented programming is not supported in C.	C++ is an object-oriented programming language; it allows inheritance, encapsulation, and polymorphism.
As a procedural programming language, C separates data and functions.	In C++, data and functions are combined into a single object.
C does not define functions inside of structures.	In C++, functions can be utilized inside of structures.
Access modifiers are not present in C structures.	There are access modifiers in C++ structures.
".c" is the file extension.	The file extension is either ".cpp," ".c++," ".cc," or ".cxx."
The C language contains 32 keywords.	C++ contains 97 keywords.

In the language C, a variable is a named memory address that is used to store and retrieve data. The variable can be used to hold many kinds of data and can be reused an unlimited number of times to store different sorts of data.

The name and type of a variable are specified in the C declaration syntax.

Syntax:

data_type variable_name = value; // for single variable

or

data_type variable_name1, variable_name2; //

for multiple variable

Explanation:

data_type: The kind of data that can be kept in a variable.

variable_name: The user-provided variable name.

value: The value that the user has assigned to the variable.

Example:

```
#include <stdio.h>

int main()

{

    int defined_var;

    printf("Defined_var: %d\n", defined_var);

    defined_var = 12;

    int ini_var = 25;

    printf("Value of defined_var after initialization:
%d\n", defined_var);

    printf("Value of ini_var: %d", ini_var);

    return 0;

}
```

Output

Defined_var: 0

Value of defined_var after initialization: 12

Value of ini_var: 25

Rules for Naming Variable in C Programming Language

- The only characters allowed in a variable name are underscores, numbers, and letters.
- Only an alphabet or an underscore may appear at the beginning of a variable name. A digit cannot be used to begin it.

- The variable name cannot contain any white space.
- There cannot be any reserved words or keywords in a variable name.

Variable Types in C

The following categories apply to the C variables:

Local Variables: In C, a variable declared inside a function or code block is referred to as a local variable.

Global Variables: In C, a variable defined outside of a function or code block is referred to as a global variable.

Static Variables: In C, a variable declared using the static keyword is referred to as a static variable.

Automatic Variables: By default, all local variables are automatically assigned values. Another name for them is auto variables. The auto keyword can be used to declare the auto variables if necessary.

Extern Variables: Numerous C files can share external variables. With the extern keyword, we may declare an external variable.

Register Variables: In C, variables saved in the CPU register as opposed to more traditional storage locations like RAM are referred to as register variables.

[**C and C++ Training**](#)

Constants in C

In C, a variable that is defined in the program and cannot be changed is called a constant. Once the constant variables are defined, we are unable to alter their value.

The const keyword in the C language is used to specify a constant. The const keyword, which

declares a variable as a constant, is added to the beginning of the variable declaration. It is also referred to as a const type qualifier.

Syntax:

```
const data_type var_name = value;
```

Example:

```
#include <stdio.h>

int main()
{
    const int int_const = 25;

    const char char_const = 'A';

    const float float_const = 15.66;

    printf("Printing value of Integer Constant: %d\n",
        int_const);

    printf("Printing value of Character Constant: %c\n",
        char_const);

    printf("Printing value of Float Constant: %f",
        float_const);

    return 0;
}
```

Output

Printing value of Integer Constant: 25

Printing value of Character Constant: A

Printing value of Float Constant: 15.660000

Types of Constants in C

The data type of the variables and the constant are the same. The list of different types of constants is as follows:

- Integer Constant
- Character Constant

- Floating Point Constant
- Double Precision Floating Point Constant
- Array Constant
- Structure Constant

Example: Constant Macro

```
#include <stdio.h>

#define pi 3.14

int main()

{

    printf("The value of pi: %.2f", pi);

    return 0;

}
```

Output

The value of pi: 3.14

C++ Variables

A memory location in C++ is referred to as a variable. It is a program's fundamental storage unit.

- Variable values are subject to alteration while a program is running.
- All operations performed on a variable affect the memory address it refers to; a variable is merely a name assigned to a memory location.
- Every variable in C++ needs to be declared before it can be used.

Variable Declaration in C++

A standard declaration of a variable looks like this:

// Declaring a single variable

type variable_name;

// Declaring multiple variables:

*type variable1_name, variable2_name,
variable3_name;*

Explanation:

datatype: The kind of data that this variable is capable of holding.

variable_name: The variable's name.

value: This is the starting value that the variable contains.

Rules for Declaring Variable in C++

- The variable's name is made up of underscores, numbers, and letters.
- The variable's name is case-sensitive (for example, Arr and arr are two distinct variables).
- There are no whitespace or special characters (such as #, \$, %, *, etc.) in the variable name.
- Every variable name must start with an underscore (_) or an alphabetic letter.
- C++ keywords (such as float, double, and class) cannot be used as variable names.

Types of Variables

According to the scope of variables in C++, there are three different kinds of variables.

Local Variables: A local variable is one that is defined inside a constructor, method, or block.

Instance Variables: Non-static variables called instance variables are declared in a class outside any constructors, methods, or blocks.

Static Variables: Another name for static variables is class variables. The static keyword is used to define these variables inside a class, outside of any method constructor or block, whereas instance variables are declared similarly.

Constants in C++

Use the const keyword (which will designate the variable as "constant," meaning unchangeable and read-only) when you don't want other people (or

yourself) to alter the values of the variable already in place:

Example

```
const int myNum = 15;

myNum = 10;
```

C++ Developer Salary

Data Types in C and C++

A variable in C must be of a specific data type, as detailed in the Variables chapter, and in order to be shown, it must be used with a format specifier inside the printf() function:

Example:

```
int myNum = 5;           // Integer (whole number)

float myFloatNum = 5.99; // Floating point number

char myLetter = 'D';     // Character

printf("%d\n", myNum);

printf("%f\n", myFloatNum);

printf("%c\n", myLetter);
```

Data Type	Size	Description
int	2 to 4 bytes	handles full numbers only —no decimals.
float	4 bytes	holds fractions with one or more decimals in them. Enough to hold six or seven decimal places.

double	8 bytes	holds fractions with one or more decimals in them. Enough to hold fifteen decimal places.
char	1 byte	holds ASCII values or a single character, letter, or number.

Data Types of C++

C++ supports a wide variety of data types. Programmers can select the one best suited to the needs of the application. Data types specify the size and types of values to be stored. While C++ instructions are the same across all processors, the storage representation and machine instructions used to modify each type of data vary.

The following data types can be used with C++:

Primary or Built-in or Fundamental data type:

These data types are preset or built-in, and the user can declare variables using them directly. Example: char, float, bool, and int. Among the primitive data types in C++ are:

- Integer
- Character
- Boolean
- Floating Point
- Double Floating Point
- Valueless or Void
- Wide Character

Derived data types: The term “derived data types” refers to data types that are derived from built-in or primitive datatypes. These can be divided into four

categories:

- Function
- Array
- Pointer
- Reference

Abstract or user-defined data types: Data types that the user creates are known as user-defined or abstract data types. The following user-defined datatypes are available in C++:

- Class
- Structure
- Union
- Enumeration
- Typedef defined Datatype

Example

```
#include <iostream>

#include <limits.h>

using namespace std;

int main()
{
    cout << "Size of char : " << sizeof(char) << " byte"
        << endl;

    cout << "char minimum value: " << CHAR_MIN << endl;

    cout << "char maximum value: " << CHAR_MAX << endl;

    cout << "Size of int : " << sizeof(int) << " bytes"
        << endl;

    cout << "Size of short int : " << sizeof(short int)
        << " bytes" << endl;

    cout << "Size of long int : " << sizeof(long int)
        << " bytes" << endl;

    cout << "Size of signed long int : "
        << sizeof(signed long int) << " bytes" << endl;

    cout << "Size of unsigned long int : "
```

```
<< sizeof(unsigned long int) << " bytes" << endl;

cout << "Size of float : " << sizeof(float) << " bytes"

<< endl;

cout << "Size of double : " << sizeof(double)

<< " bytes" << endl;

cout << "Size of wchar_t : " << sizeof(wchar_t)

<< " bytes" << endl;

return 0;

}
```

Output

Size of char : 1 byte

char minimum value: -128

char maximum value: 127

Size of int : 4 bytes

Size of short int : 2 bytes

Size of long int : 8 bytes

Size of signed long int : 8 bytes

Size of unsigned long int : 8 bytes

Size of float : 4 bytes

Size of double : 8 bytes

Size of wchar_t : 4 bytes

Operators in C and C++

Values and variables can be operated on using operators. The + operator can be used to add together a variable and a value, or a variable and another variable, even though it is commonly used to add together two values.

Example

```
int sum1 = 100 + 50;
```

```
int sum2 = sum1 + 250;
```

```
int sum3 = sum2 + sum2;
```

The C operators are separated into the following groups:

Arithmetic operators: Common mathematical operations are carried out using arithmetic operators. They are +, -, *, /, %, ++, and - as in mathematics.

Assignment operators: Variable values are assigned using assignment operators. They will be denoted with "=". They are =, +=, -=, *=, /=, %=, &=, |=, ^=, >>=, and <<=.

Comparison operators: When comparing two values, or variables, comparison operators are employed. This is crucial to programming because it facilitates decision-making and answer-finding.

A comparison's return value can be either 1 or 0, denoting true (1) or false (0). Boolean values are what these values are called. They are ==, !=, >, <, >=, and <=.

Logical operators: By combining several conditions, logical operators can be utilized to ascertain the logic between variables or values. They are && (and), || (or), and ! (not).

Bitwise operators: At the bit level, bit operators operate. In C, they are utilized for bitwise operations. They are bitwise AND (&), bitwise OR (|), bitwise XOR (^), left shift (<<), right shift (>>), and bitwise NOT (~).

C++ Operators

Like C, an operator is a symbol that can be used to carry out particular logical or mathematical operations on a value. They are every programming language's cornerstone. The built-in operators in C++ offer the required functionality.

Example: `int c = a + b;`

There are six different categories of operators in C++:

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Ternary or Conditional Operators

Arithmetic Operators: On the operands, these operators are used to carry out mathematical or arithmetic operations. For example, the operators "+" and "-" are used for addition and subtraction, respectively, and "*" is used for multiplication.

There are two types of arithmetic operators:

- **Unary Operators:** These operators only use one operand when operating or working. Operators for increment (++) and decrement (--).
- **Binary Operator:** Operators that operate or work with two operands are known as binary operators. Take addition (+), subtraction (-), etc. as examples.

Relational Operators: When comparing the values of two operands, these operators are employed.

For instance, '>' determines whether or not one operand is greater than the other, among other things. A Boolean value, such as true or false, is returned as the result. They are ==, !=, >, <, >=, and <=.

Logical Operators: These operators can be applied to combine two or more restrictions or conditions, or they can be used to enhance the way the initial condition under consideration was evaluated. A Boolean value, such as true or false, is returned as the result. They are && (and), || (or), and ! (not).

Bitwise Operators: The operands can be subjected to bit-level processing using these operators. The computation is done on the operands after the operators have been translated to bit-level.

For quicker processing, mathematical operations like addition, subtraction, multiplication, etc., can be carried out at the bit level. They are bitwise AND (&), bitwise OR (|), bitwise XOR (^), left shift (<<), right shift (>>), and bitwise NOT (~).

Assignment Operators: These operators are employed to give a variable a value. A value is the operand on the right side of the assignment operator, and a variable is the operand on the left.

The compiler will report an error if the value on the right is not the same data type as the variable on the left. They are =, +=, -=, *=, /=, %=, &=, |=, ^=, >>=, and <<=.

Conditional or Ternary Operators (?:): Based on the condition, this operator returns the value.

Expression1? Expression2: Expression3

Conclusion

This C and C++ tutorial covers basic concepts of two programming languages, such as variables, constants, data types, operators, and examples. Stay in touch to get new updates in our tutorials and enroll in the [C and C++ training in Chennai](#) for comprehensive learning.

Share on your Social
Media



Navigation

About Us

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.

Landmark: Karnataka Bank Building

Phone: [+91 86818 84318](tel:+918681884318)

Email: enquiry@softlogicsys.in

Map: [Google Maps Link](#)

OMR

No. EI-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai – 600 130.

Landmark: Adj. to AGS Cinemas

Phone: [+91 89256 88858](tel:+918925688858)

Email: info@softlogicsys.in

Map: [Google Maps Link](#)

Courses

Python

Software Testing

Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

[Blog Posts](#)

[Careers](#)

[Contact](#)

[Placement Training](#)

[Corporate Training](#)

[Hire With Us](#)

[Job Seekers](#)

[SLA's Recently Placed Students](#)

[Reviews](#)

[Sitemap](#)

Important Links

[Disclaimer](#)

[Privacy Policy](#)

[Terms and Conditions](#)

Social Media Links



Review Sources

[Google](#)

[Trustpilot](#)

[Glassdoor](#)

[Mouthshut](#)

[Sulekha](#)

[Justdial](#)

[Ambitionbox](#)

[Indeed](#)

[Software Suggest](#)

