

Learn to code — free 3,000-hour curriculum

NOVEMBER 4, 2021 / #C PROGRAMMING

# C vs C++ – What's The Difference?



Dionysia Lemonaki

The C and C++ programming languages power a large part of the world's products, applications, and websites.

Each helped lay the foundation for the creation of many popular programming languages, such as Java. They also support many languages that you might use regularly for your programming projects, such as Python.

In this article you'll find a general and beginner friendly overview of the two languages along with their main similarities and differences.

## The origins of C and C++

### A history of the C programming language

Ken Thompson and Dennis Ritchie had been working for quite a few years on the MULTICS (Multiplexed Information and Computing Service) project at AT&T Bell Laboratories.

Learn to code — free 3,000-hour curriculum

While doing so, he ended up writing a nearly complete operating system, Unix, from scratch in assembly language.

While working on MULTICS, both Thompson and Ritchie had been writing system software and programming utilities using higher-level languages. And they'd seen how much easier the whole process was, compared to the cryptic and hard to decipher assembly language.

Ritchie joined Thompson to help port Unix onto a newer machine – the PDP-11.

During that period they experimented with various higher level languages that could help get the job done.

They used BCPL (Basic Combined Programming Language), which was used heavily during the MULTICS era. After trying it out, Thompson ended up writing a new language – the B programming language.

The B language was similar to BCPL but was a more simple and stripped down version.

But B wasn't powerful enough and didn't take full advantage of the new features and power of the PDP-11.

Learn to code — free 3,000-hour curriculum



Thompson (sitting) and Ritchie working together at a PDP-11. [Image and text credit from Wikipedia](#)

Dennis Ritchie started to improve the B language and ended up creating the C programming language.

C is a portable language, meaning programs written in it can be transferred and used on a variety of machine architectures. It's very fast and easy to compile and has direct mapping to machine code, giving the programmer access to low level functionalities.

They ended up re-writing the Unix operating system in C in 1972.

Since C was portable and was the language Unix was implemented on, developers started adopting it and using it widely. This led to the success of the Unix operating system, and in turn the C language became popular.

Learn to code — free 3,000-hour curriculum

programming language' in 1977, which created a standard for how the

language is supposed to be used. This book popularized the language even more. C is hugely significant in the history of computing and its creation lead to the creation of many other programming languages. For that, it is often referred to as the '*mother*' of all programming languages.

## History of C++

In 1979, the researcher Bjarne Stroustrup was hired at AT&T Bell Laboratories.

During the 1970s the complexity and computational power of computers increased and limitations in the C programming language started to crop up.

In the early 1980s, Bjarne Stroustrup created a new language which was influenced by two things:

- The Object Oriented Programming capabilities of another language, Simula, which offered a different approach to programming compared to C. Code could be abstracted and better-organized and anything could be represented using classes.
- The systems programming language, C, which offered the ability to get really close to the machine hardware and do demanding low level computational tasks.

Those two ideas combined allowed for higher level abstraction

Learn to code — free 3,000-hour curriculum

In 1984 'C with classes' was renamed to C++.

So, C++ is a superset of C, meaning that it was an *extension* of C and is based on it. C++ just provides additional capabilities to the C language.

## Similarities between C and C++

Below are some of the similarities between C and C++.

### Syntax and code structure

The overall syntax of the two languages is very similar. The operators and keywords used in C are also used in C++ to achieve the same things. But C++ has more keywords than C, and it has an extended grammar.

Inline comments, `//`, and block comments, `*/ */`, look the same.

In addition, every statement ends with a semicolon, `;`.

Conditionals, looping, initializing and declaring variables – they all look similar between the two languages.

Both C and C++ have a `main()` method, which kickstarts every program, and both include header files at the top of the respective files, with `#include`.

### Compiled programming languages

Both C and C++ are compiled programming languages.

Learn to code — free 3,000-hour curriculum

It takes the source code that a programmer wrote in a higher level programming language and *translates* it into another language that the computer can understand.

This form is first assembly code which gets translated again to machine code – the native language of all computers.

Machine language is a set of instructions which are understood directly by a computer's CPU (Central Processing Unit).

Once the source code has been translated to machine code, a binary executable file, `a.out`, gets created.

## Differences between C and C++

Now let's look at a few of the differences between the two languages.

### Input and Output methods

C and C++ use different ways to output information to the console and receive information from the user.

In C, `scanf()` is used for user input, whereas `printf()` is used for outputting data.

In C++, `std::cin >>` is used for getting user input and `std::cout <<` is used to output data.

### The programming paradigm

---

Learn to code — free 3,000-hour curriculum

C is a **procedural oriented language** and its emphasis is on functions.

Programs are divided into a set of functions and they consist of step-by-step instructions, or commands, to be executed in sequential order.

This style of programming specifies *how* to do something, giving structured steps for how computational tasks will be carried out, following a top-down approach.

This style of programming can get quite messy and error-prone when programs grow in size. It leads to a lot of copying and pasting throughout the file and updating many functions when there is a change.

Besides being a procedural language, C++ is also an **Object Oriented Programming language**, which is based on the concept of dividing a program into objects.

Everything is organized and divided into smaller groups of related parts or *objects*, which are an instance of a *class*, following a bottom-up approach.

Object Oriented Programming is based on four principles: *encapsulation, abstraction, inheritance and polymorphism*.

This style of programming creates more readable and usable code that is easier to maintain, while also providing better data security.

## The STL

C++ offers the STL – Standard Template Library – and C doesn't

Learn to code — free 3,000-hour curriculum  
components for implementing added, built-in functionality.

One such component is containers such as Vectors, that store collections of objects.

## Namespaces

Namespaces are a feature available in C++ and not C.

They are containers used to organize code into logical groups of identifiers and similar objects under a name, within a scope.

They prevent name collisions when multiple libraries are present, and prevent conflict with names of other namespaces within a program.

One example of a namespace is `std::`.

One way to use a namespace and introduce it into a scope is by using the `using` keyword, for example `using namespace std;`.

## Exception handling

C doesn't offer a way to handle exceptions in programs which help prevent errors.

C++, on the other hand, supports exception handling by introducing `try` and `catch` blocks.

## File extension

The file extension for a file that contains C code is `.c`, whereas the file extension for C++ files is `.cpp`.



Learn to code — free 3,000-hour curriculum

## WHERE ARE C AND C++ USED:

C is commonly used for very demanding, low-level computational tasks where speed, efficiency, and close access to the machine are a must.

C assumes that programmers know what they are doing and gives them freedom.

It is therefore the language of choice for operating systems, embedded devices, systems programming, kernels and drivers, compiler development, and the growing industry of IoT (Internet of Things) applications.

C++ again allows the programmer close access to and manipulation of the machine, while providing efficiency and high-performance for large-scale systems. At the same time, it is higher-level with enough abstraction away from the machine.

C++ is a popular language of choice for creating game engines and computer graphics and applications, VR applications, web browsers such as Google Chrome, Mozilla Firefox, and Safari, and web browser extensions. The Google Search Engine is also built in C++.

## How to learn C and C++

Below is a list of some resources to help you get started on your C and C++ learning journey.

To learn C:

---

Learn to code — free 3,000-hour curriculum

---

## Beginners

- [C Programming Tutorial for Beginners](#)

To learn C++:

- [The C++ Programming Language](#)
- [C++ Tutorial for Beginners](#)
- [Learn Object Oriented Programming \(OOP\) in C++](#)
- [Learn Modern C++ by Building an Audio Plugin](#)

## Conclusion

Thanks for making it to the end, and I hope you have found this article helpful.

You learned about the origins of C and C++ and their historical background. You then saw a few of their similarities and differences, how each language is used, and some resources for you to get started learning the languages.

Happy coding!



**Dionysia Lemonaki**

Learning something new everyday and writing about it

---

Learn to code — free 3,000-hour curriculum

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers.

Get started

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) nonprofit organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

Trending Guides

Big O Notation	HTML Link
SQL Outer Join	Bayes Rule
Python For Loop	Python Map
What is JavaScript?	HTML Italics
Learn How To Code	Python SQL
Chrome Bookmarks	HTML Bold
Concatenate Excel	GraphQL VS Rest
C# String to Int	If Function Excel
Git Switch Branch	HTML List

[Forum](#)[Donate](#)

## Learn to code — free 3,000-hour curriculum

[Git Checkout Remote Branch](#)[String to Char Array Java](#)[Insert Checkbox in Word](#)[JavaScript Append to Array](#)[Find and Replace in Word](#)[Add Page Numbers in Word](#)[C Programming Language](#)[JavaScript Projects](#)

## Our Nonprofit

[About](#) [Alumni Network](#) [Open Source](#) [Shop](#) [Support](#) [Sponsors](#) [Academic Honesty](#)[Code of Conduct](#) [Privacy Policy](#) [Terms of Service](#) [Copyright Policy](#)