Setting up C++ Development Environment

Difficulty Level : Basic • Last Updated : 20 Mar, 2023

C++ is a general-purpose programming language and is widely used nowadays for competitive programming. It has imperative, object-oriented, and generic programming features.

C++ runs on lots of platforms like Windows, Linux, Unix, Mac, etc. Before we start programming with C++. We will need an environment to be set up on our local computer to compile and run our C++ programs successfully. If you do not want to set up a local environment you can also use online IDEs for compiling your program.

Using Online IDE

IDE stands for an integrated development environment. IDE is a software application that provides facilities to a computer programmer for developing software. There are many online IDEs available that you can use to compile and run your programs easily without setting up a local development environment. The

ide.geeksforgeeks.org is one such IDE provided by GeeksforGeeks.

You can click on the Run on IDE button to run the program.

C++

```
// Using online ide of C++
#include <iostream>
using namespace std;

int main()
{
    cout << "Learning C++ at GeekforGeeks";
    return 0;
}
```

Output

```
Learning C++ at GeekforGeeks
```

Time Complexity: O(1)
Auxiliary Space: O(1)

Setting up a Local Environment

For setting up a C++ Integrated Development Environment (IDE) on your local machine you need to install two important software:

1. C++ Compiler
2. Text Editor

1. C++ Compiler

Once you have installed the text editor and saved your program in a file with the '.cpp' extension, you will need a C++ compiler to compile this file. A compiler is a computer program that converts high-level language into machine-understandable low-level language. In other words, we can say that it converts the source code written in a programming language into another computer language that the computer understands. For compiling a C++ program we will need a C++ compiler that will convert the source code

written in C++ into machine codes. Below are the details about setting up compilers on different platforms.

Installing GNU GCC on Linux

We will install the GNU GCC compiler on Linux. To install and work with the GCC compiler on your Linux machine, proceed according to the below steps:

A. You have to first run the below two commands from your Linux terminal window:

```
sudo apt-get update
sudo apt-get install gcc
sudo apt-get install g++
```

B. This command will install the GCC compiler on your system. You may also run the below command:

```
sudo apt-get install build-essential
```

This command will install all the libraries which are required to compile and run a C++ program.

C. After completing the above step, you should check whether the GCC compiler is installed in your system correctly or not. To do this you have to run the below-given command from the Linux terminal:

```
g++ --version
```

D. If you have completed the above two steps without any errors, then your Linux environment is set up and ready to be used to compile C++ programs. In further steps, we will learn how to compile and run a C++ program on Linux using the GCC compiler.

E. Write your program in a text file and save it with any file name and.CPP extension. We have written a program to display “Hello World” and saved it in a file with the filename “helloworld.cpp” on the desktop.

F. Now you have to open the Linux terminal and move to the directory where you have saved your file. Then you have to run the below command to compile your file:

```
g++ filename.cpp -o any-name
```

G. *filename.cpp* is the name of your source code file. In our case, the name is “helloworld.cpp” and *any-name* can be any name of your choice. This name will be assigned

to the executable file which is created by the compiler after compilation. In our case, we choose *any-name* to be "hello".
We will run the above command as:

```
g++ helloworld.cpp -o hello
```

H. After executing the above command, you will see a new file is created automatically in the same directory where you have saved the source file and the name of this file is the name you chose as *any-name*.
Now to run your program you have to run the below command:

```
./hello
```

I. This command will run your program in the terminal windows.

2. Text Editor

Text Editors are the type of programs used to edit or write texts. We will use text editors to type our C++ programs. The normal extension of a text file is (.txt) but a text file containing a C++ program should be saved with a '.cpp' or '.c' extension. Files ending with the extension '.CPP' and '.C' are called source code files and they are supposed to contain source code written in C++ programming language. These extension helps the compiler to identify that the file contains a C++ program.
Before beginning programming with C++, one must have a text editor installed to write programs. Follow the below instructions to install popular code editors like VS Code and Code::Block on different Operating Systems like windows, Mac OS, etc.

1. Code::Blocks Installation
There are lots of IDE available that you can use to work easily with the C++ programming language. One of the popular IDE is **Code::Blocks**.

- To download Code::Blocks, select the setup package based on your OS from this link – Code::Blocks Setup Packages.
- Once you have downloaded the setup file of Code::Blocks from the given link open it and follow the instruction to install.
- After successfully installing Code::Blocks, go to *File* menu -> Select *New* and *create an Empty* file.
- Now write your C++ program in this empty file and save the file with a '.cpp' extension.
- After saving the file with the '.cpp' extension, go to the *Build* menu and choose the *Build and Run* option.

2. XCode Mac OS X Installation

If you are a Mac user, you can download Xcode as a code editor.

- To download Xcode you have to visit the apple website or you can search for it on the apple app store. You may follow the link – Xcode for MacOS to download Xcode. You will find all the necessary installation instructions there.
- After successfully installing Xcode, open the Xcode application.
- To create a new project. Go to File menu -> select New -> select Project. This will create a new project for you.
- Now in the next window, you have to choose a template for your project. To choose a C++ template choose the *Application* option which is under the *OS X* section on the left sidebar. Now choose *command-line tools* from available options and hit the *Next* button.
- On the next window provide all the necessary details like 'name of organization', 'Product Name, etc. But make sure to choose the language as C++. After filling in the details hit the next button to proceed to further steps.
- Choose the location where you want to save your project. After this choose the *main.cpp* file from the directory list on the left sidebar.
- Now after opening the main.cpp file, you will see a pre-written c++ program or template provided. You may change this program as per your requirement. To run your C++ program you have to go to the *Product* menu and choose the *Run* option from the dropdown.

Another very easy-to-use and most popular IDE nowadays, is **VSC(Visual Studio Code),** for both Windows and Mac OS.

3. Installing VS Code on Windows

Start with installing Visual Studio Code as per your windows. Open the downloaded file and click Run -> (Accept the agreement) Next -> Next -> Next -> (check all the options) -> Next ->Install->Finish.

Now you'll be able to see the Visual Studio Code icon on your desktop.

- Download the MinGW from the link.
- After Install, "Continue". Check all the Packages (Right Click -> Mark for Installation). Now, Click on Installation (left corner) -> Apply Changes. (This may take time)
- Open This PC -> C Drive -> MinGW -> Bin. (Copy this path)
- Right, Click on "This PC" -> Properties -> Advanced System Setting -> Environment variables -> (Select PATH in System variables) -> Edit -> New -> Paste the path here and OK.
- Go to Visual Studio Code, and Install some useful extensions (from the right sidebar, last icon(probably))-
  - C/C++
  - Code Runner

- Now, Go to Setting -> Settings -> Search for Terminal -> Go to the end of this page -> Check [Code-runner: Run In Terminal]

Yay! You are good to go now. Open any folder, create new files and Save them with the extension ".cpp".

4. Installing VS Code on Mac OS

First of all, Install Visual Studio Code for Mac OS using this link. Then We'll install the compiler MinGW. For this, we first need to install Homebrew.

To install Homebrew, Open Terminal (cmd + space). Write Terminal and hit Enter. In cmd copy the given command

```
arch -x86_64 ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install)" <
/dev/null 2> /dev/null
```

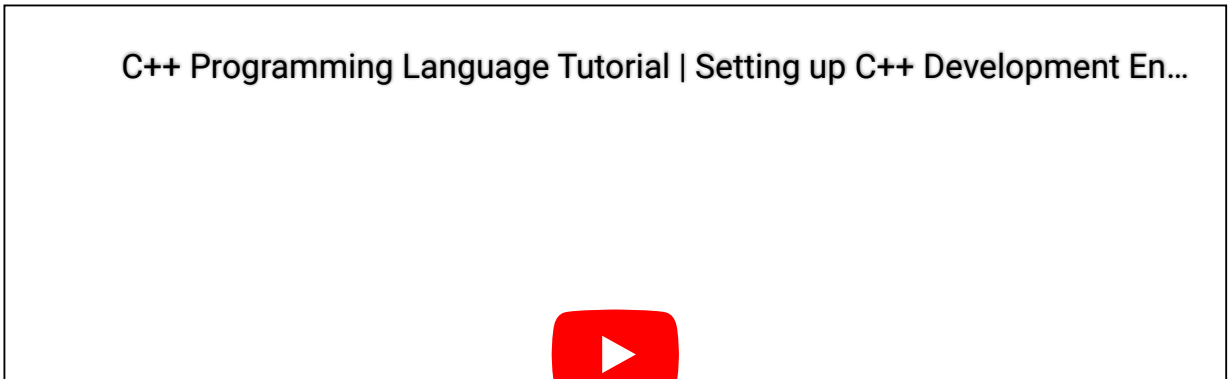
This will download and install HomeBrew on your Mac system. This process may take time.

Now We'll install the MinGW compiler on Mac OS. Paste the given command in the terminal and press Enter.

```
arch -x86_64 brew install MinGW-w64
```

- This is also time taking process so be patient!
- Go to Visual Studio Code, and Install some useful extensions (from the right sidebar, last icon(probably))-
  - C/C++
  - Code Runner
- Now, Go to Setting -> Settings -> Search for Terminal -> Go to the end of this page -> Check [Code-runner: Run In Terminal]

Yay! You are good to go now. Now open any folder, create new files, and Save them with the extension ".cpp".



This article is contributed by **Harsh Agarwal**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks. Please write comments if you find anything incorrect, or if you want to share more information about the topic discussed above.

914

Related Articles

1. Setting up a C++ Competitive Programming Environment
2. Setting up Sublime Text for C++ Competitive Programming Environment
3. Different Ways to Setting Up Environment For C++ Programming in Mac
4. Installing MinGW Tools for C/C++ and Changing Environment Variable
5. Top 10 Programming Languages for Blockchain Development
6. Setting Up Sublime Text For Competitive Coding in C++14 on Ubuntu
7. Setting up Sublime Text For Competitive Programming (C++) Using Fast Olympic Coding Plugin
8. C++ Error - Does not name a type
9. Execution Policy of STL Algorithms in Modern C++
10. C++ Program To Print Pyramid Patterns

Previous Next

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Basic