

## Introduction

It can be challenging to set up a C++ development environment, especially if you're new to the language.

This guide will cover installing GCC, the GNU Compiler Collection, which is a compiler for C, C++, Objective-C, Fortran, Ada, and Go. More importantly, it's the compiler almost all competitive programming sites use to compile and run your code.

If you run into issues while using the document, or could use more detail, or have a favorite resource to contribute: please let us know / post on the Forums, and we'll get the document updated!

## Installing GCC

### MacOS

Note: MacOS ships with a `gcc` terminal command which is actually just an alias for a different compiler called `clang`; GCC is probably not installed. To check if you have actual GCC, run `gcc --version` in your terminal: if the output mentions `clang` (or if there is an error) then you do not yet have GCC and should follow the instructions here.

The most common way to install GCC on MacOS is through **Homebrew**, a package manager for MacOS and Linux.

1. Install Homebrew by going to <https://brew.sh/> and running the command in your terminal.

If the script asks you for your password, enter it. Note that the cursor will not move while typing your password, even as it is accepting keypresses.

2. Install GCC by running the following command in your terminal:

```
brew install gcc
```

3. Check that GCC is installed by running the following command in your terminal:

```
gcc-13 --version
```

4. Discover where `g++-13` got installed by running the following command in your terminal:

```
which g++-13
```

The answer will probably be either `/usr/local/bin/g++-13` or `/opt/homebrew/bin/g++-13`.

5. Next, we're going to symlink the new C++ compiler to the `g++` command. Run the following command in your terminal, replacing "G++PATH" with the result printed by the previous step:

```
sudo ln -s G++PATH /usr/local/bin/g++
```

Again, this may ask you for your password. If you get a "File exists" error, use `-sf` instead of `-s`.

6. Restart your terminal completely.

7. Check that `g++` is linked to `gcc-13` by running the following command:

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

It should output something along the lines of `g++-13 (Homebrew GCC 13.2.0) 13.2.0`. If it does, you're all set! If not, please ask for help on the Forums.

At least one of our students has found that reinstalling Xcode Command Line Tools got them unstuck.

## Windows

We've found that the most common way to install GCC on Windows is through **MinGW**, a port of GCC for Windows.

If you're using CLion, you don't need to do anything else, as CLion bundles MinGW.

If you're planning to use VSCode (or some other IDE), you'll need to install MinGW yourself. You can find detailed instructions on how to do so at <https://code.visualstudio.com/docs/cpp/config-mingw>. If you're using an IDE that isn't VSCode, you can use the guide above and skip the VSCode-specific steps.

## Linux

If you're using Linux, you probably already have GCC installed. If not, you can install it on debian-based distros (e.g. Ubuntu) by running the following command in your terminal:

```
sudo apt install gcc
```

If you're using another distro, you can find instructions on how to install GCC at <https://gcc.gnu.org/install/>, or even try following the Homebrew instructions for MacOS.

## Setting up your IDE

There are a few differences between CLion and VSCode.

VSCode is a free, open-source IDE that is lightweight and customizable.

CLion is a free-for-students IDE that is very easy to set up, but is a bit more heavyweight. You can view the CLion setup instructions at <https://www.jetbrains.com/help/clion/installation-guide.html>. You may need to apply for an educational license at <https://www.jetbrains.com/community/education/#students>.

There ultimately isn't a "better" option, so we recommend trying both and seeing which one you like more. It's far more important that you get comfortable with your IDE than it is to use a specific one.

## VSCode

1. Download VSCode by going to <https://code.visualstudio.com/>.
2. If you're on Windows, follow the instructions here to set up your C++ environment:  
<https://code.visualstudio.com/docs/cpp/config-mingw#create-hello-world>.
3. If you're on MacOS or Linux, you can follow the instructions below:
4. Click the "Extensions" button on the left sidebar, or press **Ctrl+Shift+X** (Windows/Linux) or **Cmd+Shift+X** (MacOS).
5. Search for "C/C++" and click "Install", and follow the instructions from there.

## CLion

1. You may need to apply for an educational license at  
<https://www.jetbrains.com/community/education/#students>.
2. Download CLion by following the directions at  
<https://www.jetbrains.com/help/clion/installation-guide.html>.

3. Open CLion and click “New Project”. Select “C++ Executable”. Feel free to change the folder, but make sure to remember where you put it. Click “Create”.
4. If you have MacOS, make sure your compiler is set to GCC by going to CLion > Preferences > Build, Execution, Deployment > Toolchains and typing g++-13 as your C++ compiler.
5. A lot of competitive programmers use the C++17 standard, so we recommend setting that as your default. You can do so by going to CLion > Preferences > Build, Execution, Deployment > C++ Compiler and adding `-std=c++17` to the “Additional command line options” field. Make sure to click “Apply”.
6. Lastly, if you’re looking for specific plugins, weAutoCP plugin, which can be found at [weAutoCP](#). You can find other plugins by going to CLion > Preferences > Plugins.

## IDE Plugins

As you use your IDE to write and debug code, you may find that some tasks or features are not supported or optimized by default. Fortunately, you can extend the functionality by using plugins, which are add-ons that provide extra tools or capabilities. However, keep in mind that not all plugins are equal in terms of quality, compatibility, or security. You should always read the documentation and reviews of a plugin before installing it, and be cautious of plugins from untrusted sources.

Here are some popular plugins that some students have found useful for their workflow:

### VSCode

- Competitive Programming Helper

### CLion

- AutoCP

Thanks to our sponsors:

