Getting Your Computer Set Up

CSC 1300, Fall 2023

Department of Computer Science, College of Engineering, Tennessee Tech University

Basic Information

To **write** a program, you only need a simple text editor. For windows, this could be just ***Notepad***, which comes with the operating system. However, there are great text editors available that are free and that make coding easier. We will be using ***VS Code*** as our text editor.

To **run** a C++ program, you need a C++ compiler. There are many out there – TDM-GCC, XCode, MinGW, Cygwin, etc. The compiler you use will be based on your operating system.

The directions in this document with walk you step-by-step in installing the text editor and compiler. Please make sure you read through it and follow it carefully so you do not miss a step.

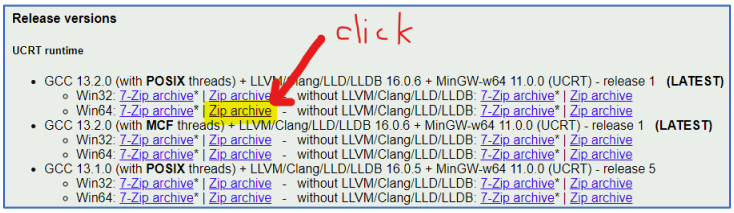
Table of Contents:

A. Visual Studio Code on Windows

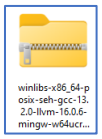
You will use Visual Studio Code (**VS Code**) on Windows as your text editor, which will allow you to write your programs. You will use the **Command Prompt** to use the **MinGW** compiler to compile & run your programs. After configuring VS Code, you will compile a simple C++ program using the Command Prompt.

1. Install MINGW on Windows

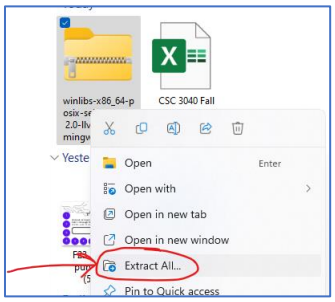
1) Install Mingw from **https://winlibs.com/**
   You will have to scroll down until you see the **Release versions** and then **UCRT runtime**. Under **GCC 13.2.0 (with MCF threads)**, click on the Zip archive link beside the Win64 bullet. (See screen capture below)



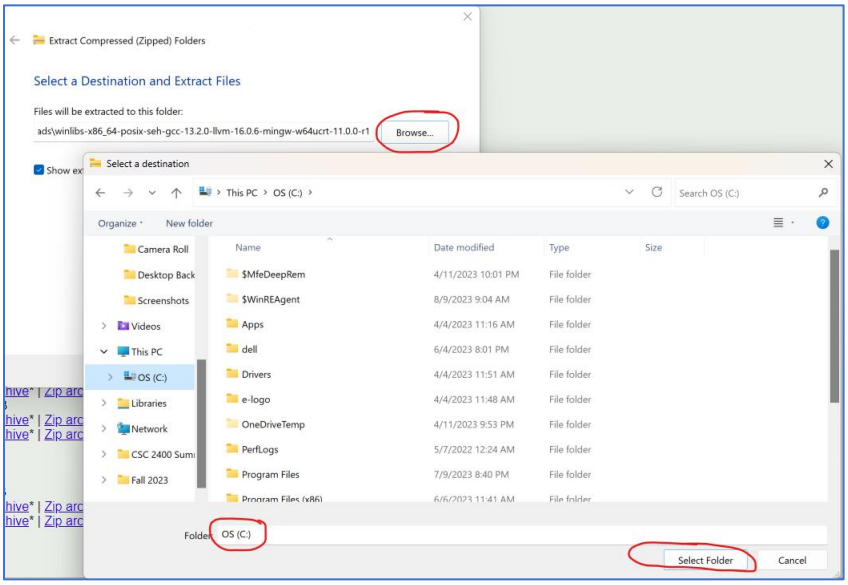
2) The **winlibs-x86_64-posix-seh-gcc-13.2.0-llvm-16.0.6-mingw-w64ucrt-11.0.0-r1.zip** file will be downloaded.



3) Right-click and select **Extract All**.

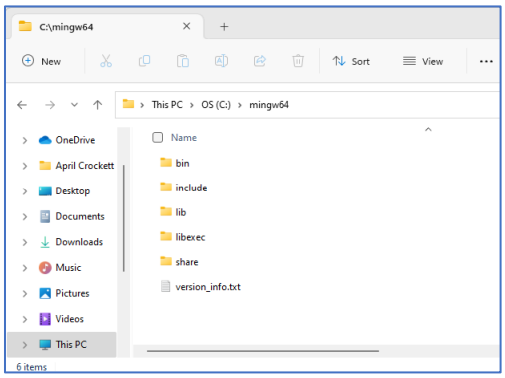


4) Click **Browse** to select your **C:** drive and then click the **Select Folder** button.



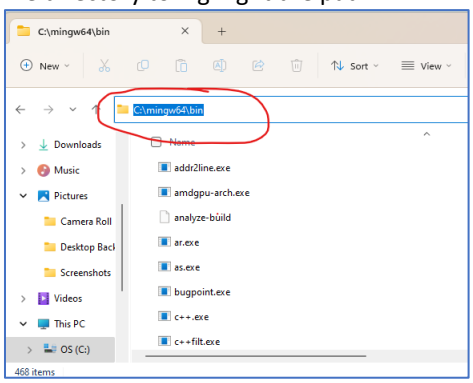
5) Click the **Extract** button. This will take several minutes because 11,474 items will be extracted.

6) After it has completed, you will have a MinGW folder in the C: drive that looks like the screen capture below.



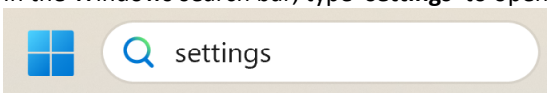
7) Below are the steps to add the path to your mingw64 **bin** folder to the Windows PATH environment variable.

a. Go to your File Explorer  and go to **This PC** and then **C:** drive and then **mingw64** and then **bin**. Then, click in the white space beside the file directory to highlight the path.

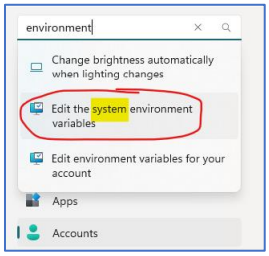


b. Hit **CTRL+C** to copy the path.

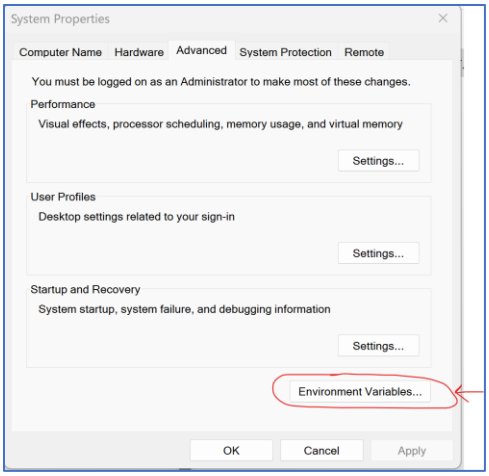
c. In the Windows search bar, type **'settings'** to open your Windows Settings.



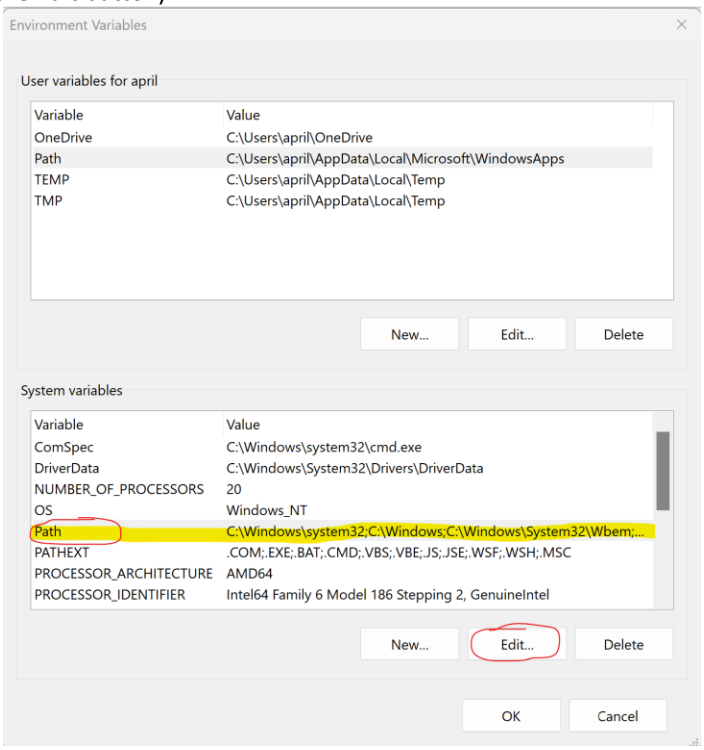
d. Search for **Edit the system environment variables**. Make sure not to select "Edit …for your account" or you will not be able to change the path.



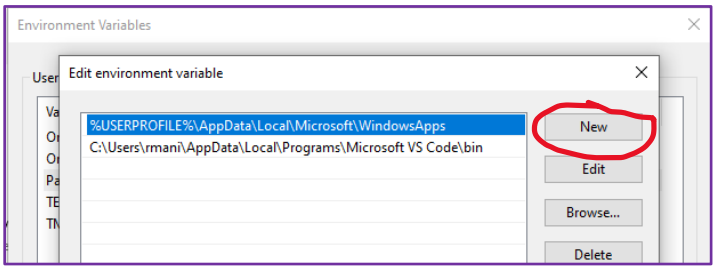
e. Click the Environment Variables button.



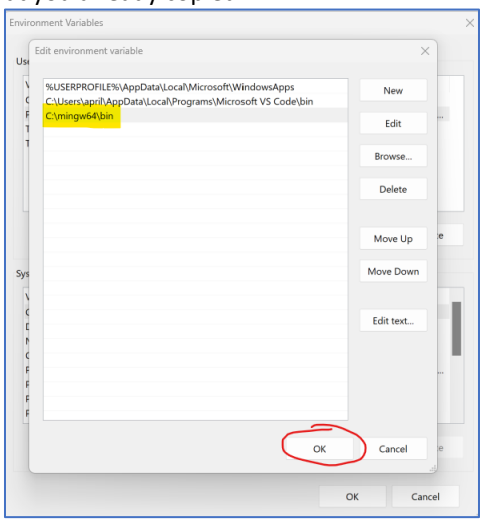
f. In the bottom half of the window under **System variables**, choose the **Path** variable (either double click or single click and select the Edit button).



g. Select **New**.



h. Paste (**CTRL+V**) the path that you already copied.

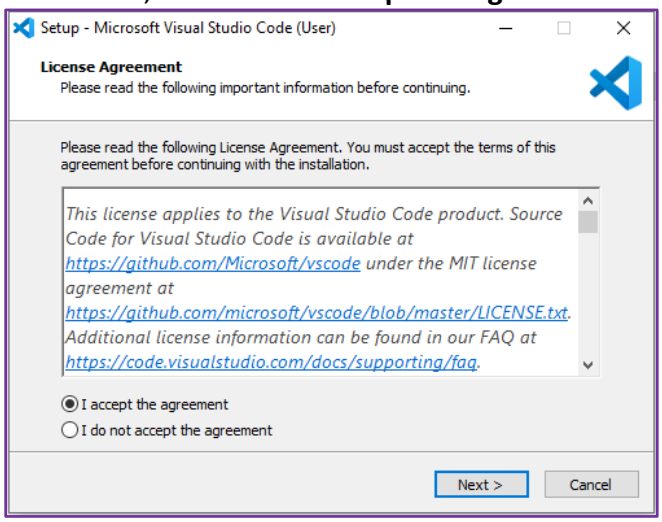


i. Click **OK** to save the Path update. You will need to reopen the Command Prompt window for the new PATH location to be available for the g++ command to work.

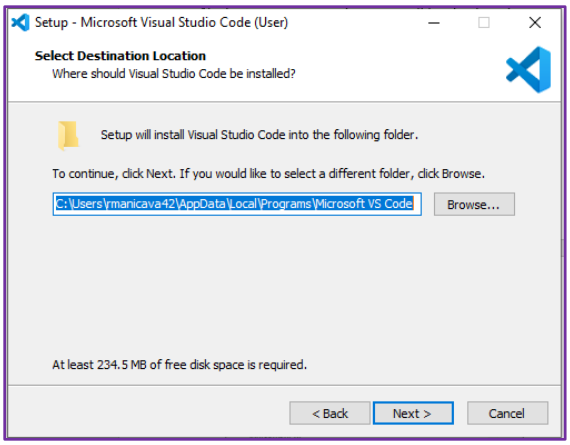
2. Installation of VS Code

1. Download either the 32-bit or 64-bit Windows **User** Installer version of Visual Studio Code (VS Code) from the link https://code.visualstudio.com/download . To know if you have 32- or 64-bit,
   - Right click on This PC in Windows File Explorer and go to Properties.
   - On the "System" type line, it should say either 64-bit Operating System or 32-bit Operating System.
   - Once you click on the correct download link, it will take you to the VS Code "Getting Started" page, and a download of the executable file should automatically begin.

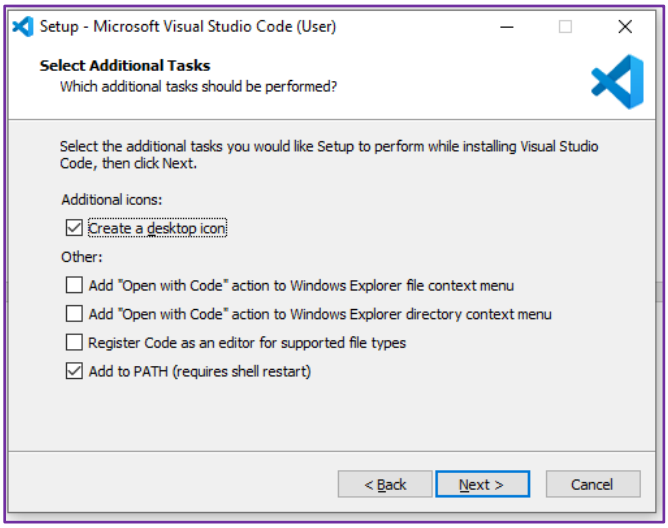
2. Run the executable to install VS Code, and check "**I accept the agreement**" and click **Next** button



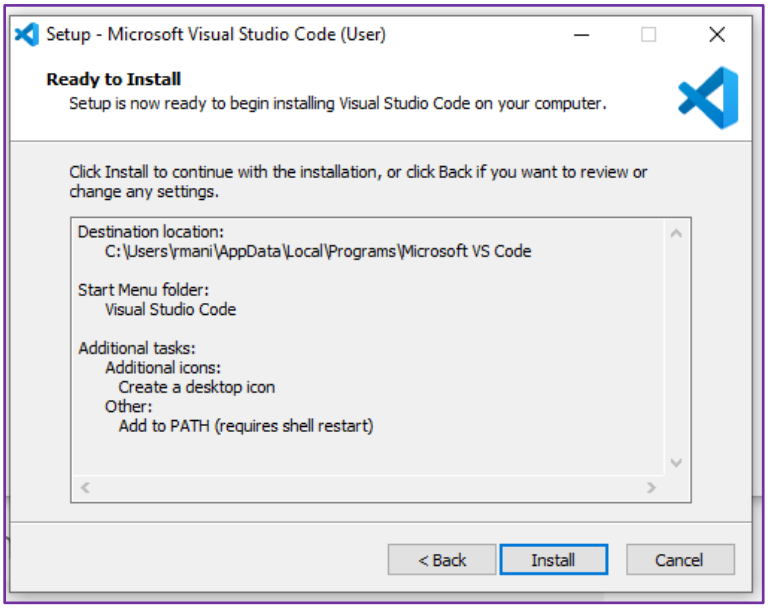
3. Select the location in your computer to install VS Code (*you can leave it to the default if you want*) and click **Next** button



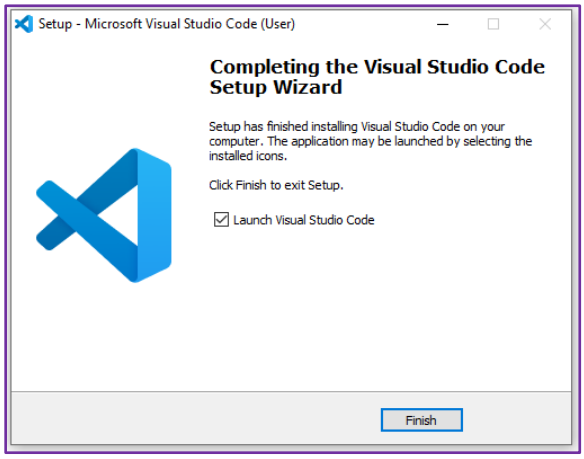
4. Check "**Create Desktop Icon**" and "**Add to PATH**", and click **Next** button



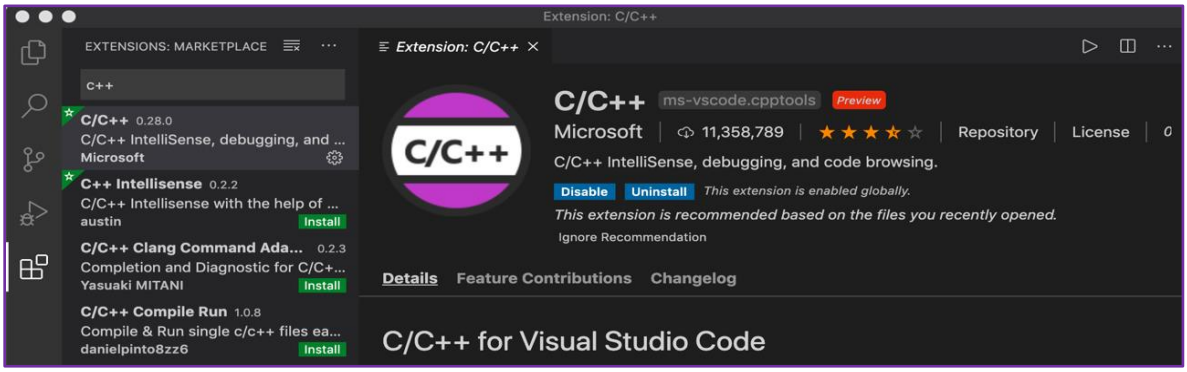
5. Click **Install** button to start installation.



6. After completion of setup, click **Finish** button to finish installation of VS Code.



7. If you select "**Launch Visual Studio Code**" in step 6 then it directly launches VS Code initially. Otherwise, you can launch VS Code by clicking the Desktop VS code icon.
8. To install the C/C++ extension type '**c++**' in the Extensions view (**Shift + ctrl +X**) and then click Install to install the C/C++ extension.



9. VS Code and the C/C++ extension does not include a C++ compiler or debugger.

3. Test Installation

3.1 Create workspace (helloworld):

You can directly create a workspace for your project by following the steps below:

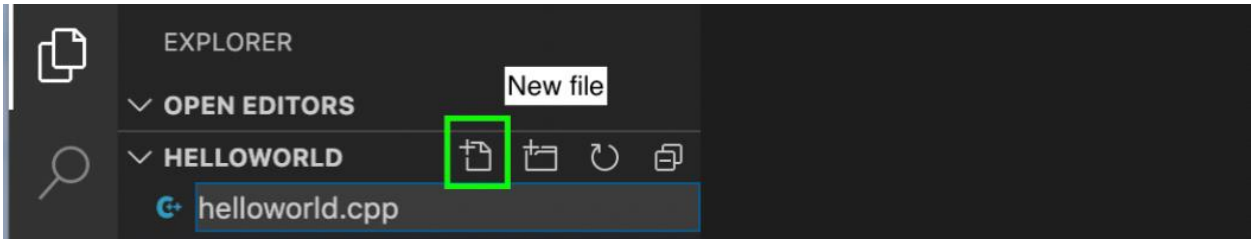
1. Create an empty folder called '**projects**' where you can store all your VS Code projects.
2. Click inside the '**projects**' folder and create a subfolder called '**helloworld**'.
3. Launch VS Code (i.e., clicking the VS Code Desktop icon**)**, and from the **start** menu select **open folder** to select the current working folder **projects > helloworld**.

Or, you can alternatively create a workspace and launch the VS Code from the Windows command prompt, by entering the following commands:

```
> mkdir projects
> cd projects
> mkdir helloworld
> cd helloworld
> code .
```

3.2 Add helloworld source code file:

In the VS Code File Explorer title bar, click the **New File** icon and name the file **helloworld.cpp**



TYPE the following lines in the **helloworld.cpp** file.

NOTE: Do NOT copy & paste or some hidden characters will be copied and your code will not work.

```
#include <iostream>
using namespace std;
int main()
{
   cout << "Hello World!";
   cout << endl;
   return 0;
}
```

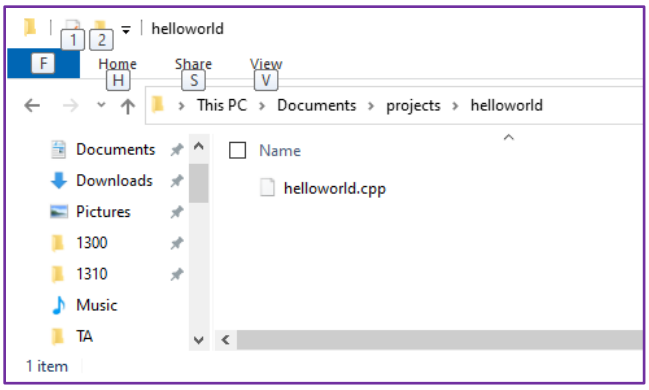
Now press **Ctrl + S** to save the file. Notice that your files are listed in the VS code **File Explorer** view (**Ctrl + Shift + E)** in the side bar of VS Code:

You can also enable Auto Save to automatically save your file changes, by checking **Auto Save** in the main **File** menu.

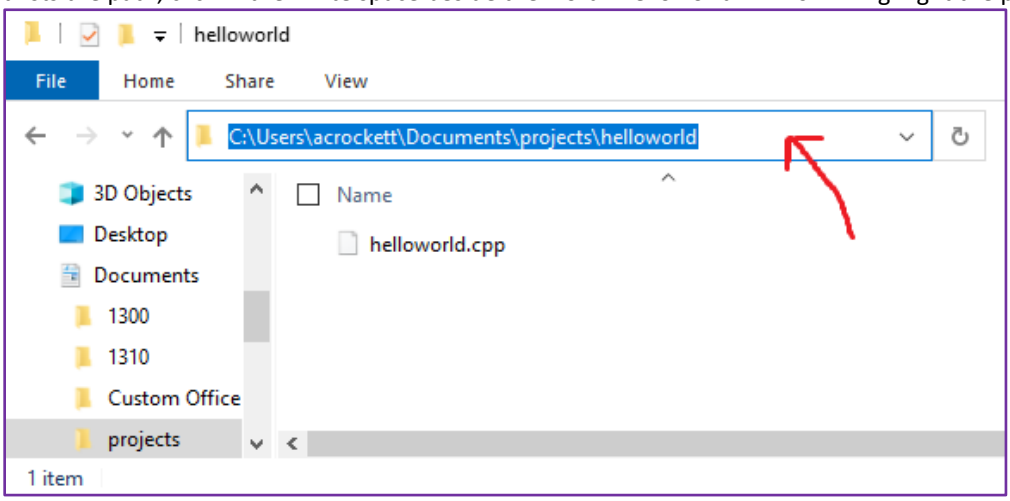
The Activity Bar on the edge of Visual Studio Code lets you open different views such as **Search**, **Source Control**, and **Run**. You can find out more about the other views in the VS Code at the link **https://code.visualstudio.com/docs/getstarted/userinterface**

3.3 Compile helloworld.cpp

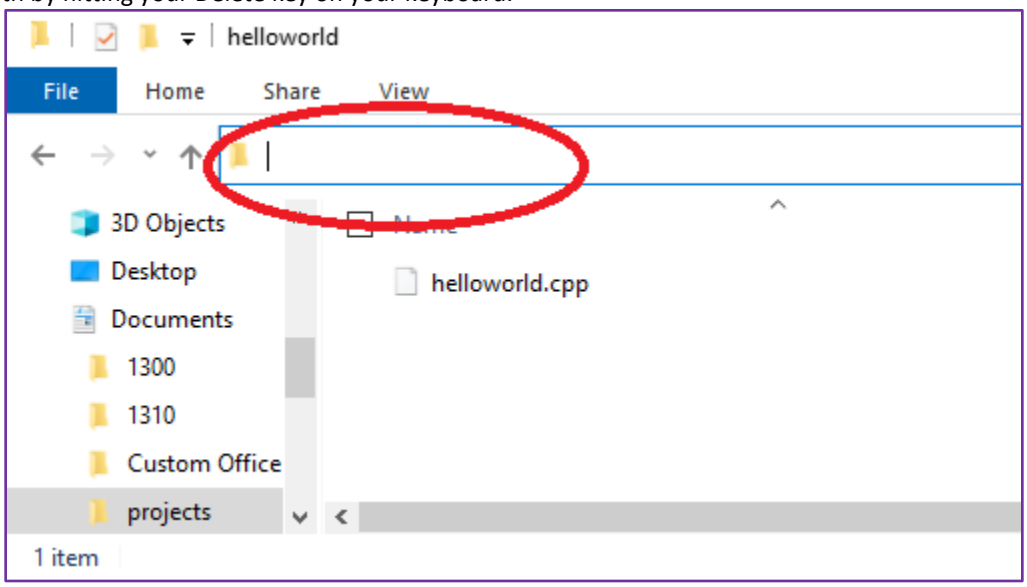
Open the Command Prompt. There are several ways to open. The best way is to open your File Explorer and navigate to where you have saved helloworld.cpp.



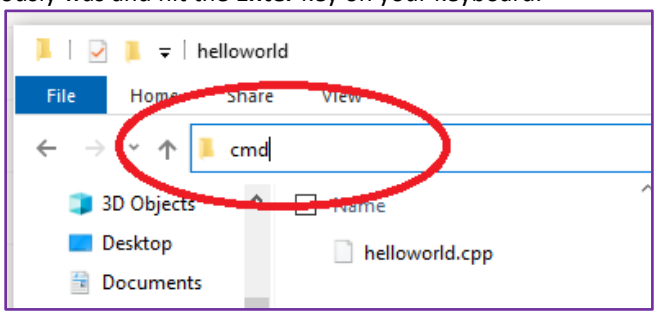
Then, in the box that lists the path, click in the white space beside the word “helloworld”. This will highlight the path.



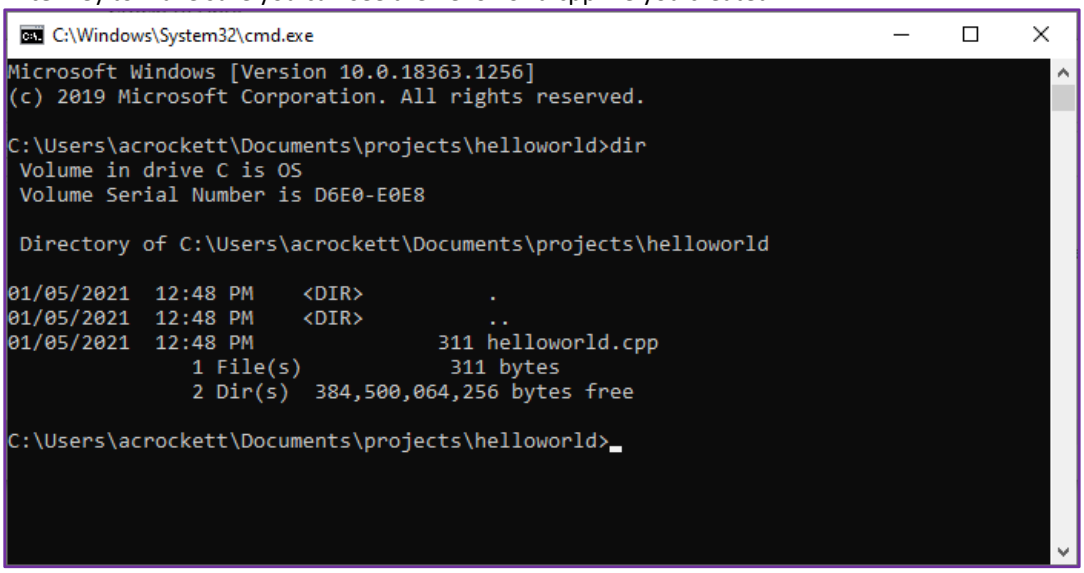
Now, delete the path by hitting your Delete key on your keyboard.



Last, type **cmd** where the path previously was and hit the **Enter** key on your keyboard.

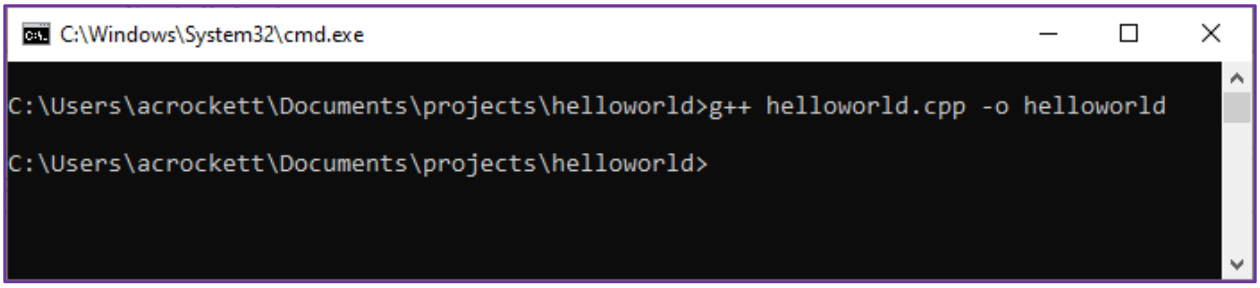


The Command Prompt should have opened on your computer to the directory where your source file is located. You can type **dir** and then hit the **Enter** key to make sure you can see the helloworld.cpp file you created.



Now, you compile the source file (.cpp file) by typing the following:

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

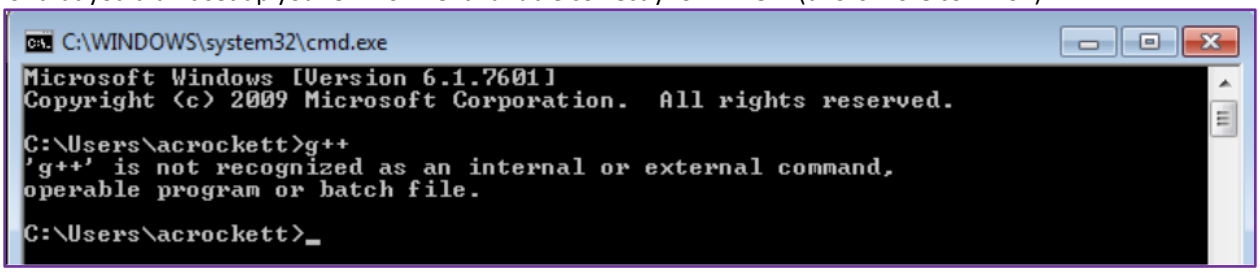


If nothing happens except the prompt appears again, this means you have no syntax errors in your code and so you are ready to run the program! You will now have an additional file in this folder named **helloworld.exe**, which is your runnable or executable file.

Troubleshooting Help

If you see the word "**error**" on the screen (like the example below), then you have at least one syntax error that must be fixed in VS Code (your text editor) before you can compile your program.

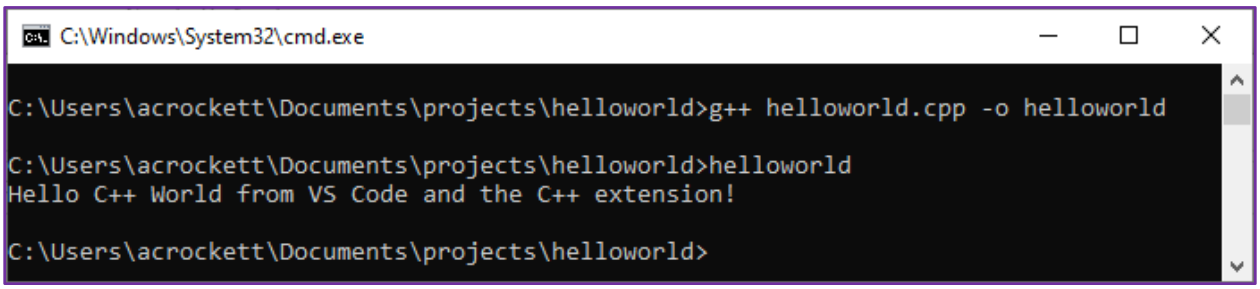
If you see an error that says that **g++ can't be recognized** (like the example below) then that means that either MinGW is not installed or that you didn't set up your environment variable correctly for MinGW (this is more common).



3.4 Run Your Program

To run the program after it successfully compiles, you type the following:

helloworld



B. Visual Studio Code on MacOS

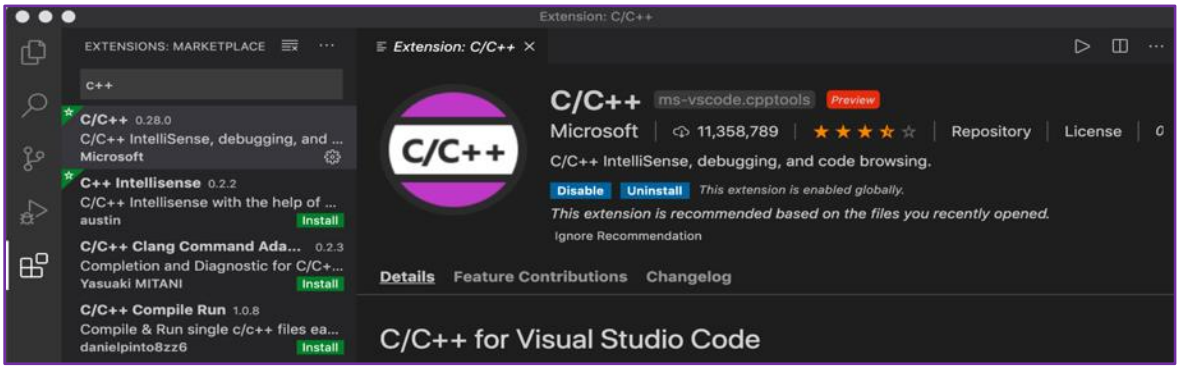
In this tutorial, you will learn to configure VS Code on macOS to use the **Clang** compiler. After configuring VS Code, you will compile a simple C++ program in VS Code.

1. Installation of VS Code & Clang

1. Download Visual Studio Code (VS Code) for macOS from the link **https://code.visualstudio.com/download**
2. Extract (Double click) the downloaded VS code file.
3. Drag **Visual Studio Code.app** to the '**Applications**' folder and make it available in the macOS Launchpad.
4. To launch VS code, press **Command + space bar** and type **Visual Studio Code.app** and click **VS code icon**. (You can add the VS Code to your Dock by **right-clicking** on the icon and choosing **Options**, **Keep in dock**. Later, you can launch the VS Code directly clicking the VS Code icon in Dock)
5. You can also launch VS Code from the macOS terminal, but path need to be added.

    - Launch VS Code (as in **step 4**)
    - Open the **Command Palette** (**Shift + command + P**) and type '**shell command**' and click **Shell Command: Install 'code' command in PATH**.

6. To install the C/C++ extension type '**c++**' in the Extensions view (**Shift + command +X**). And install C/C++ extension.



(*The C/C++ extension adds language support for C/C++ to VS Code, including features such as IntelliSense and debugging.*)

7. Clang could already be installed on your Mac. To verify the installation of Clang, open a macOS Terminal window and enter the following command:

    $ clang --version

    If **Clang** isn't installed, enter the following command:

    $ xcode-select --install

    (*Note: the "--version" and "--install" above is two hyphens before the word "version" and "install")*

2. Test Installation

2.1 Create workspace (helloworld):

You can directly create a workspace for your projects by following the steps below:

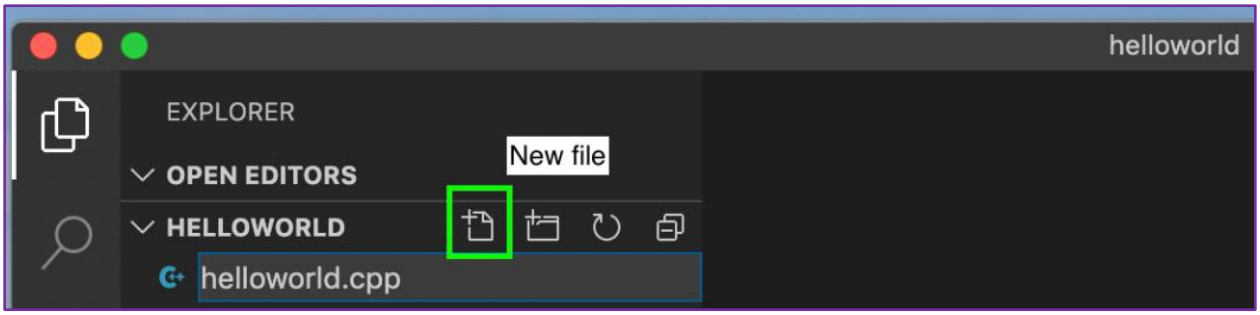
1. Create an empty folder called '**projects**' where you can store all your VS Code projects.
2. Click inside the '**projects**' folder and create a subfolder called '**helloworld**'.

3. Lunch VS Code (as in **section 1 step 4**), and from the **start** menu select **open folder** to select the current working folder **projects > helloworld**.

Or, you can alternatively create a workspace from the macOS Terminal, by entering the following commands:

```
$ mkdir projects
$ cd projects
$ mkdir helloworld
$ cd helloworld
$ code .
```

2.2 Add helloworld source code file:

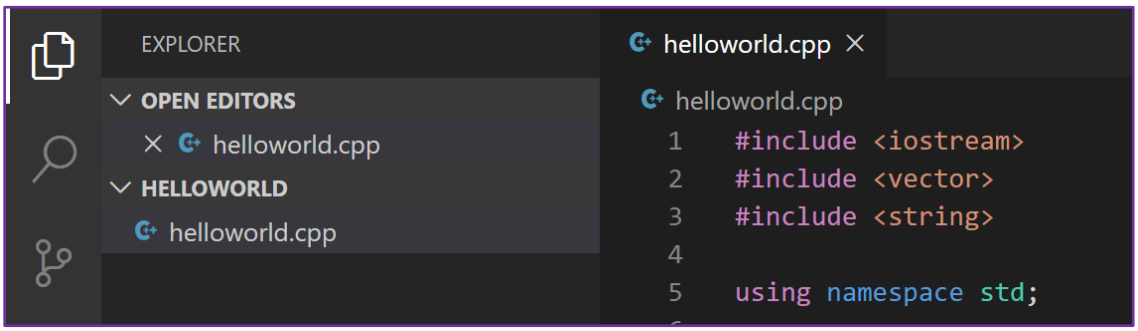


In the VS Code File Explorer title bar, click **New File icon** and name the file **helloworld.cpp**

Copy and Paste the following source code in **helloworld.cpp** file:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World!";
    cout << endl;
    return 0;
}
```

Now press **Command + S** to save the file. Notice that your files are listed in the VS Code File Explorer view (**Shift + command + E**) in the side bar of VS Code:



You can also enable Auto Save to automatically save your file changes, by checking **Auto Save** in the main **File** menu.

The Activity Bar on the edge of Visual Studio Code lets you open different views such as **Search**, **Source Control**, and **Run**. You can find out more about the other views in the VS Code in this link **https://code.visualstudio.com/docs/getstarted/userinterface**

2.3 Compile your code using Terminal

Open the Terminal application on your computer (you can do a program search for Terminal).

In Terminal, navigate to where the helloworld.cpp document is located. For example, if you created it under Documents > projects > helloworld then you will type the following in the Terminal:

```
cd Documents

cd projects

cd helloworld
```

You can type **ls** and hit the Enter key in order to see all files in the current directory. Make sure you can see helloworld.cpp.

Once you know you are at the correct directory, then type the following line to compile your source file:

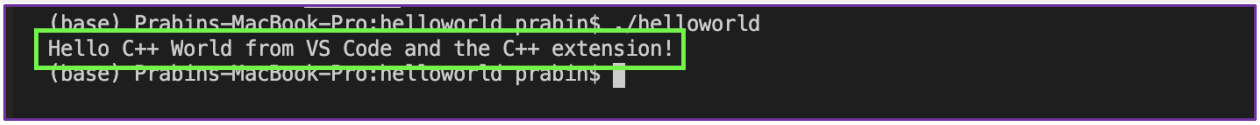
```
g++ helloworld.cpp -o helloworld
```

If the prompt reappears with no “error” messages, then your code compiled without syntax errors and you are ready to run.

2.3 Run your Code

To run your executable file for your program, you type the following command in the Terminal.

```
./helloworld
```



C. Visual Studio Code on Linux

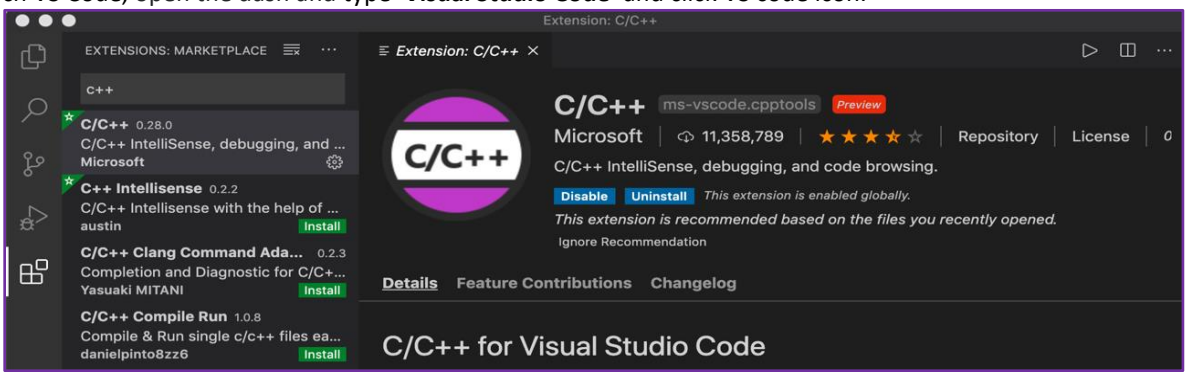
In this tutorial, you will learn to configure VS Code on Linux to use the **GCC C++** compiler. After configuring VS Code, you will compile a simple C++ program in VS Code.

4. 1. Installation of VS Code & GCC

1. Download Visual Studio Code (VS Code) of the same Linux platform that as you have in your machine via this link https://code.visualstudio.com/download . For this tutorial, we downloaded **.deb** as our machine uses Ubuntu Linux platform.

2. To install VS Code.
   - Open a terminal (**Ctrl + alt + T)**
   - Navigate to the folder where you have downloaded the VS Code file. For this tutorial, the installation file of VS Code downloaded to **Downloads** folder. So, we change the directory to it by following command:
     ```
     $ cd Downloads
     ```
   - Install VS Code by typing following command:
     ```
     $ sudo dpkg -i code_1.45.1-1589445302_amd64.deb
     ```
     Where '`code_1.45.1-1589445302_amd64.deb`' is the name of downloaded VS Code file.

3. To launch VS Code, open the dash and type **'Visual Studio Code'** and click VS code icon.



4. To install the C/C++ extension type '**c++**' in the Extensions view (**Shift + ctrl +X**). And install C/C++ extension.
5. **Ensure GCC is installed:**

   First, check to see whether GCC is already installed. To verify whether it is, open a Terminal window and enter the following command:

   ```
   $ gcc -v
   ```

   If GCC isn't installed, run the following command from the terminal window to update the Ubuntu package lists.

   ```
   $ sudo apt-get update
   ```

   Next install the GNU compiler tools and the GDB debugger with this command:

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

2. Test Installation

2.1 Create workspace (helloworld):

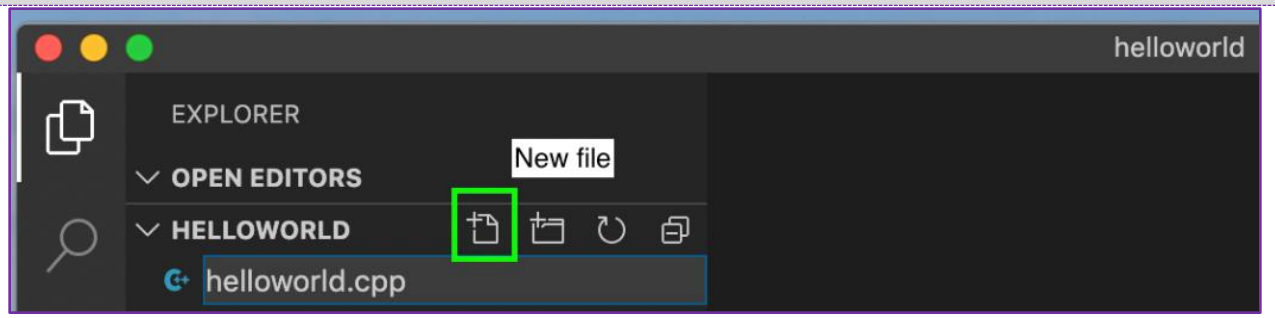
You can directly create a workspace for your projects by following the steps below:

1. Create an empty folder called '**projects**' where you can store all your VS Code projects.
2. Click inside the '**projects**' folder and create a subfolder called '**helloworld**'.
3. Lunch VS Code (as in **section 1 step 4**), and from the **start** menu select **open folder** to select the current working folder **projects > helloworld**.

Or, you can alternatively create a workspace from the macOS Terminal, by entering the following commands:

```
$ mkdir projects
$ cd projects
$ mkdir helloworld
$ cd helloworld
$ code .
```

2.2 Add helloworld source code file:

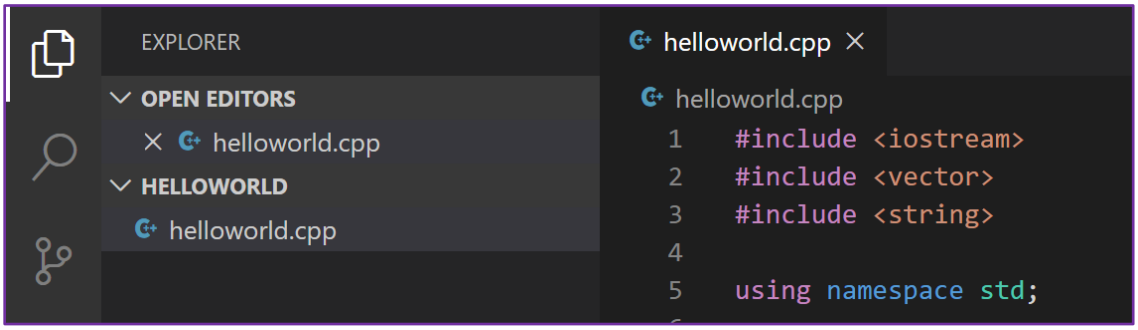


In the VS Code File Explorer title bar, click **New File icon** and name the file **helloworld.cpp**

Copy and Paste the following source code in **helloworld.cpp** file:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World!";
    cout << endl;
    return 0;
}
```

Now press **Command + S** to save the file. Notice that your files are listed in the VS Code File Explorer view (**Shift + command + E**) in the side bar of VS Code:



You can also enable Auto Save to automatically save your file changes, by checking **Auto Save** in the main **File** menu.
The Activity Bar on the edge of Visual Studio Code lets you open different views such as **Search**, **Source Control**, and **Run**. You can find out more about the other views in the VS Code in this link https://code.visualstudio.com/docs/getstarted/userinterface

2.3 Compile your code using Terminal

Open the Terminal application on your computer (you can do a program search for Terminal).
In Terminal, navigate to where the helloworld.cpp document is located. For example, if you created it under Documents > projects > helloworld then you will type the following in the Terminal:

```
cd Documents

cd projects

cd helloworld
```

You can type `ls` and hit the Enter key in order to see all files in the current directory. Make sure you can see helloworld.cpp.

Once you know you are at the correct directory, then type the following line to compile your source file:

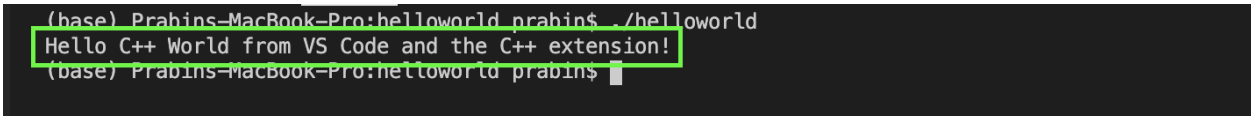
```
g++ helloworld.cpp -o helloworld
```

If the prompt reappears with no “error” messages, then your code compiled without syntax errors and you are ready to run.

2.4 Run your Code

To run your executable file for your program, you type the following command in the Terminal.

```
./helloworld
```



D. Explore IntelliSense

IntelliSense is an intelligent code-completion tool that includes a number of features such as list members, parameter information, quick information, and complete word. These features help you to learn more about the code you’re using, keep track of the parameters you’re typing, and add calls to properties and methods with just a few keystrokes.

For example, go to the **helloworld.cpp** file used in an earlier example. On line 8 after you have defined the vector named **msg**, type **msg.** and you will see a large, scrollable list of methods that can be used with this vector. You can use your up/down key to scroll through the list and hit the **tab** key to select one of the methods. For this example, select **at.** Then, type in a **(** and another popup window will appear that gives you information on this function and its expected arguments.

E. Color Themes

You can select a color theme for VS Code by opening the Color Theme picker with **File** > **Preferences** > **Color Theme**.

You can also use the keyboard shortcut **Ctrl+K Ctrl+T** to display the picker. Read more about Color Themes and download many more from the Extension Marketplace by visiting the Visual Studio Code website Themes page:
https://code.visualstudio.com/docs/getstarted/themes