

CSCI 1300 CS1: Starting Computing  
Godley/Hoefer - Fall 2023  
Visual Studio Code - Windows

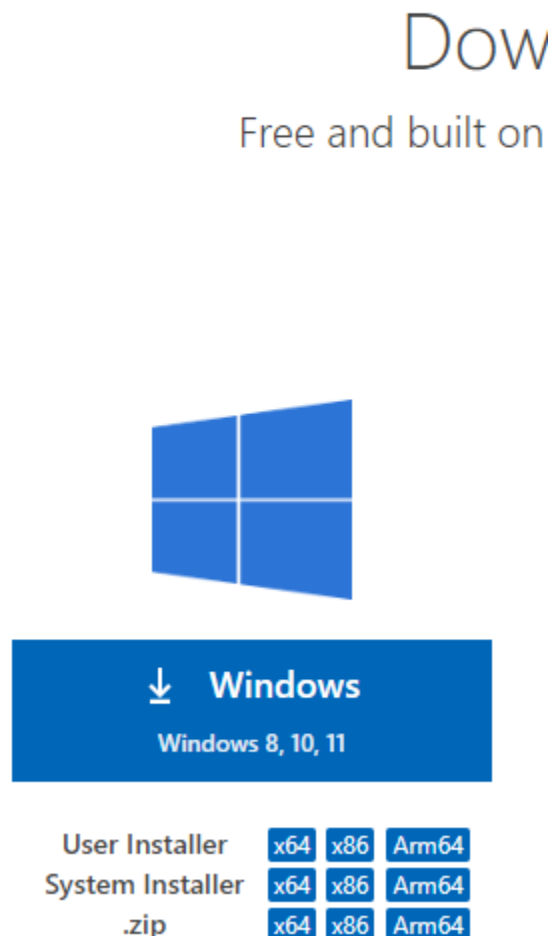
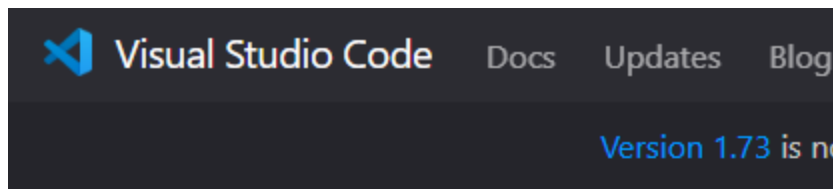
You will use Visual Studio Code (VS Code) to write and execute your programs locally.

**Important: Before proceeding with this document, make sure that you have run Windows Update within your Windows 10 or 11 environment. You must have the latest updates installed.**

## Windows Installation Guide (Part 1)

### Step 1:

- 1.1 Go to the VS code [download page](#), and download for Windows.



User Installer	<a href="#">x64</a>	<a href="#">x86</a>	<a href="#">Arm64</a>
System Installer	<a href="#">x64</a>	<a href="#">x86</a>	<a href="#">Arm64</a>
.zip	<a href="#">x64</a>	<a href="#">x86</a>	<a href="#">Arm64</a>

- **1.2** Run the installer and accept all of the default settings.
- **1.3** Click on Install and wait for Visual Studio Code to finish installing, then close the installer.

## Step 2: Installing MinGW

This section is based on this guide from Microsoft:

<https://code.visualstudio.com/docs/cpp/config-mingw>

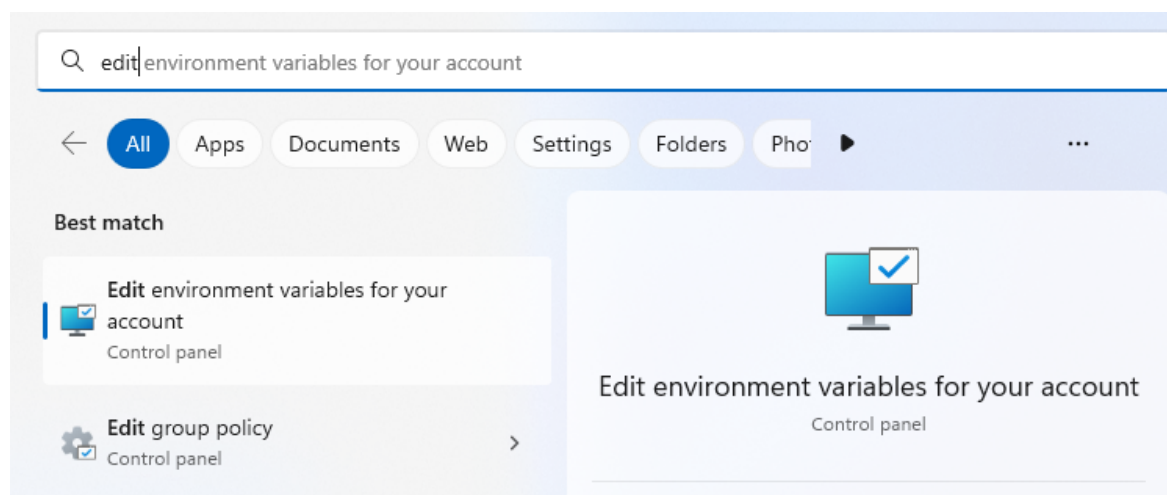
MinGW is a windows C/C++ compiler tool set that will allow us to compile our C/C++ code into a .exe file.

- **2.1** First Install MinGW from this link:  
[https://github.com/msys2/msys2-installer/releases/download/2022-06-03/msys2-x86\\_64-20220603.exe](https://github.com/msys2/msys2-installer/releases/download/2022-06-03/msys2-x86_64-20220603.exe)
- **2.2** Open the installer and choose the Defaults for all settings.
- **2.3** At the end of the installation run msys2 and then run the following command (Shift + Insert is the paste shortcut in MSYS2's terminal)

```
pacman -S --needed base-devel mingw-w64-x86_64-toolchain
```

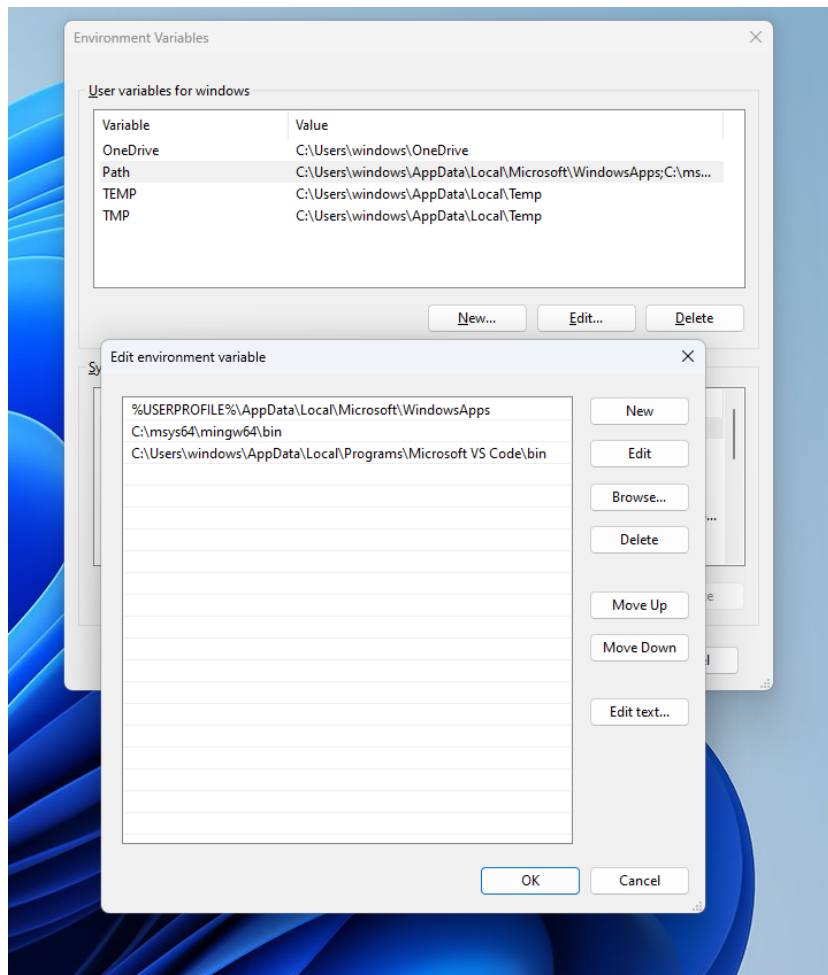
- **2.4** Press enter when prompted to install all of the default packages, then press Y to confirm the install. This will take 1 to 5 minutes to finish. Once the install completes you can close msys2.
- **2.5** Now we need to add msys2 to window's PATH variable.

Press the Windows key and begin typing "Edit environment variables for your account" until you see this option.



Now Select the "PATH" variable and click edit, in the window that opens click "New" and enter the following path for the default installation location of Msys2.

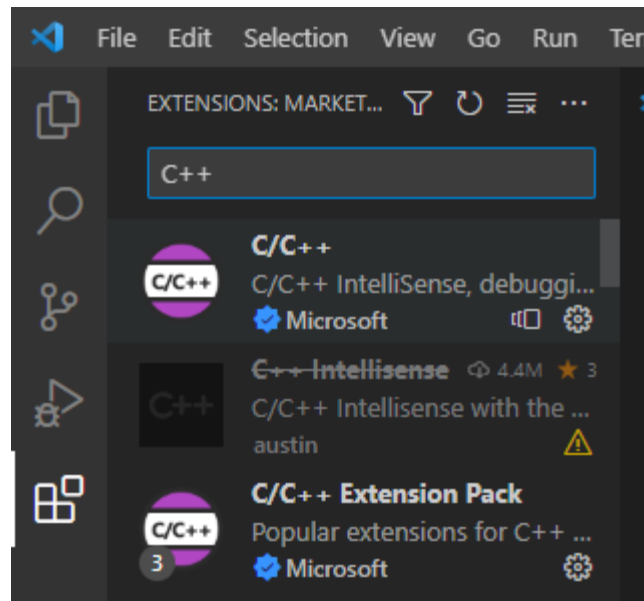
```
C:\msys64\mingw64\bin
```



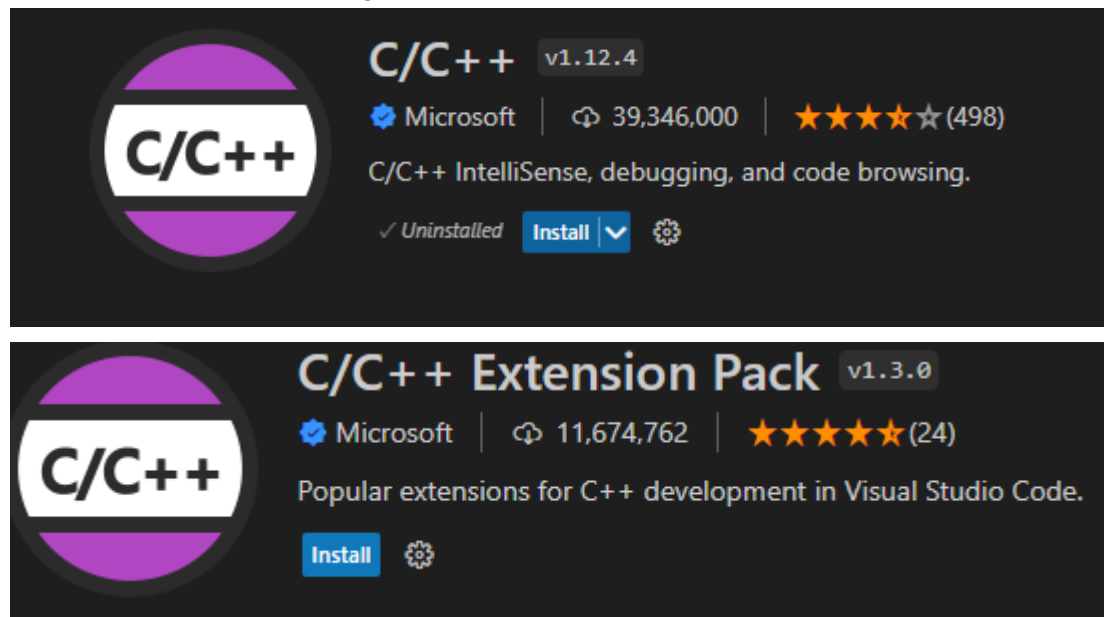
- **2.5** Press Ok to save these changes. Reboot is recommended at this point before continuing.

### Step 3: Adding VS code extensions

- **3.1** After you Reboot open VScode and select the extensions tab. (5th from the top), and search for "**C++**". We need to install the "**C/C++**" & "**C/C++ Extension Pack**" both from Microsoft.



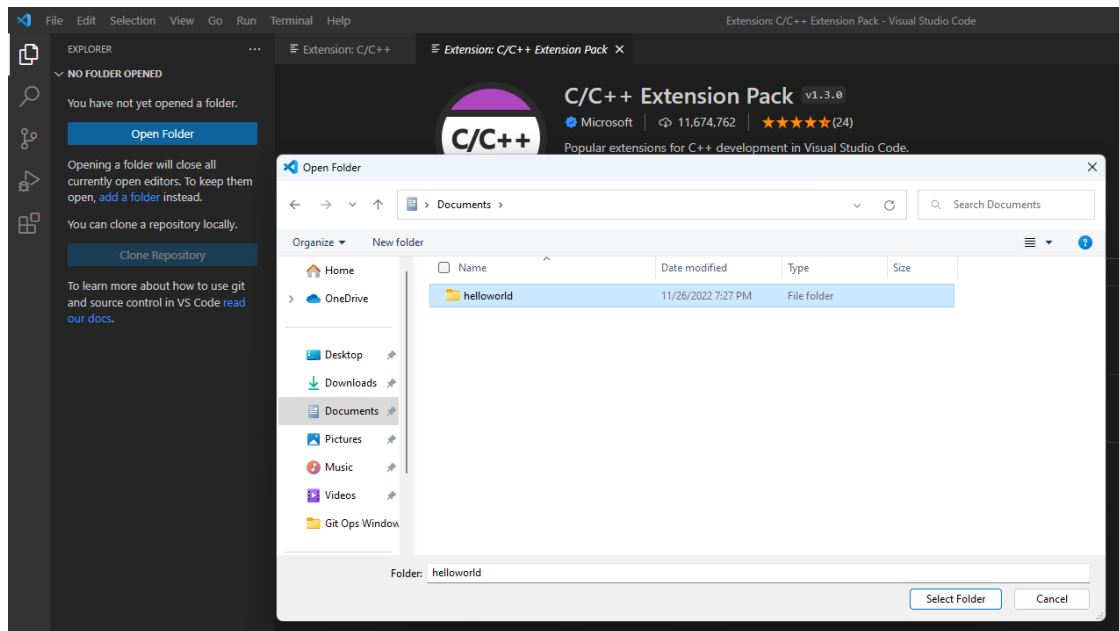
- **3.2** Select the extension then click on install, these will provide Syntax Highlighting and other useful tools when working in C++



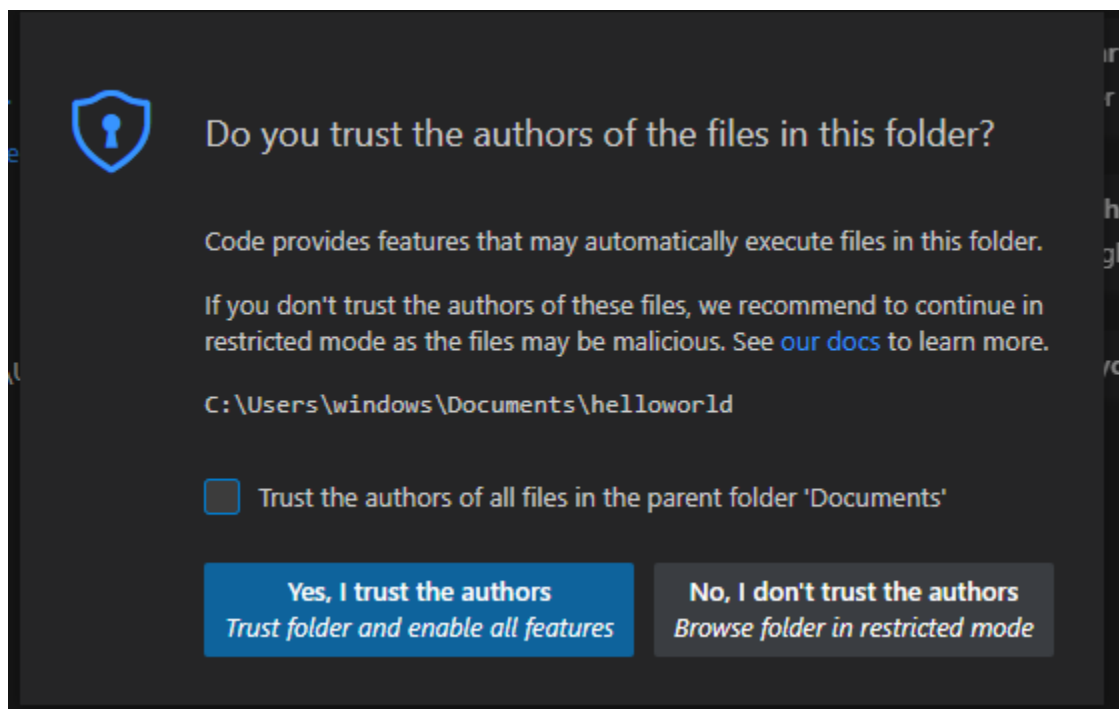
#### Step 4: Testing

Now we can test what we have installed.

- **4.1** Make a new folder in a convenient folder like Documents and call the folder "helloworld"  
Open that folder in VS code.



- **4.2** Trust the folder to allow feature like syntax highlight and debugging to work properly



- **4.3** Now create a new file called **helloworld.cpp** and paste the following code into it.

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

int main()
{
    vector<string> msg {"Hello", "C++", "World", "from", "VS Code", "and the
C++ extension!"};
```

```

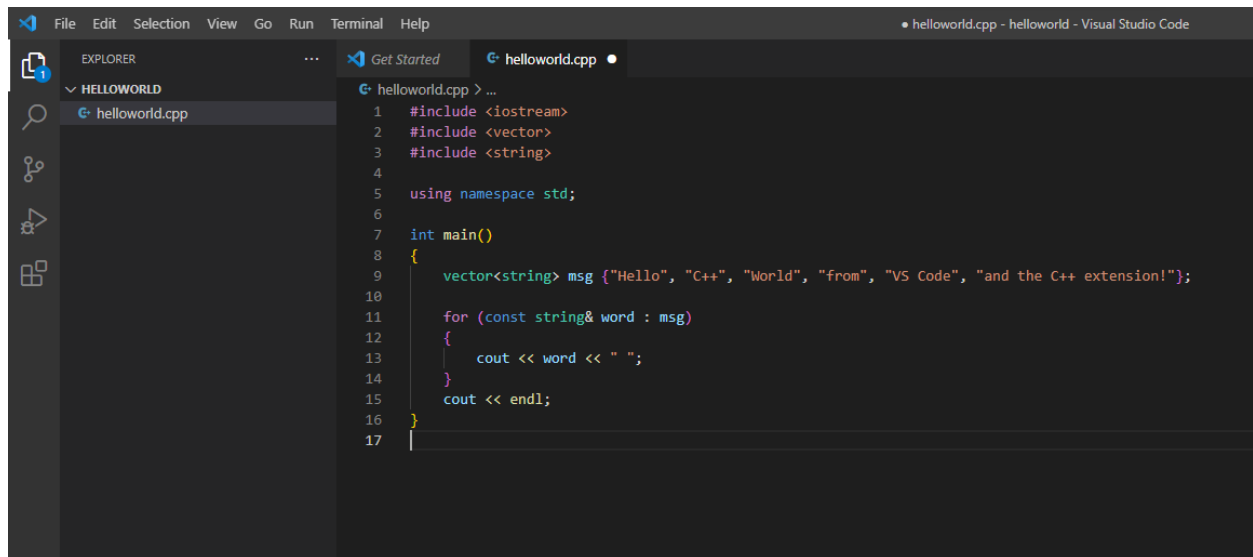
    for (const string& word : msg)
    {
        cout << word << " ";
    }
    cout << endl;
}

```

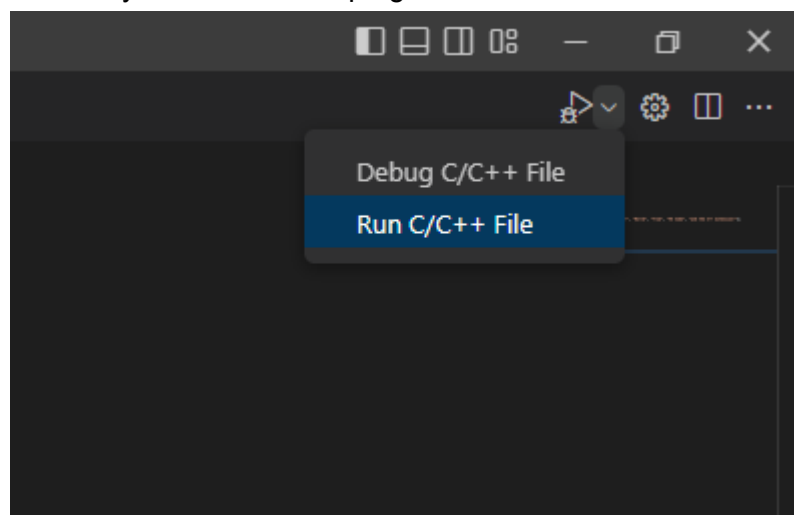
Your VS Code window should look like this.

If you have formatting issues when pasting, this same code is available from the Microsoft article here:

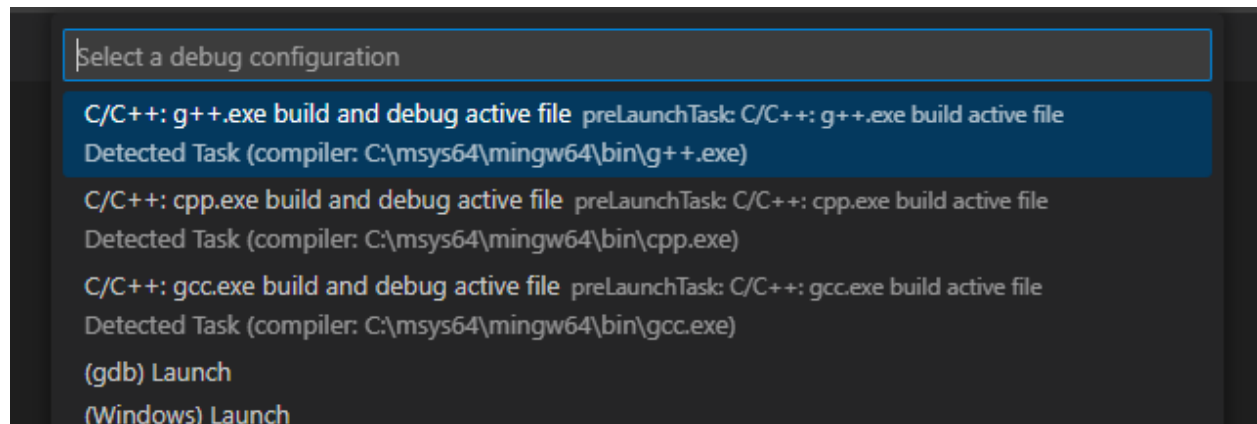
[Get Started with C++ and Mingw-w64 in Visual Studio Code](#)



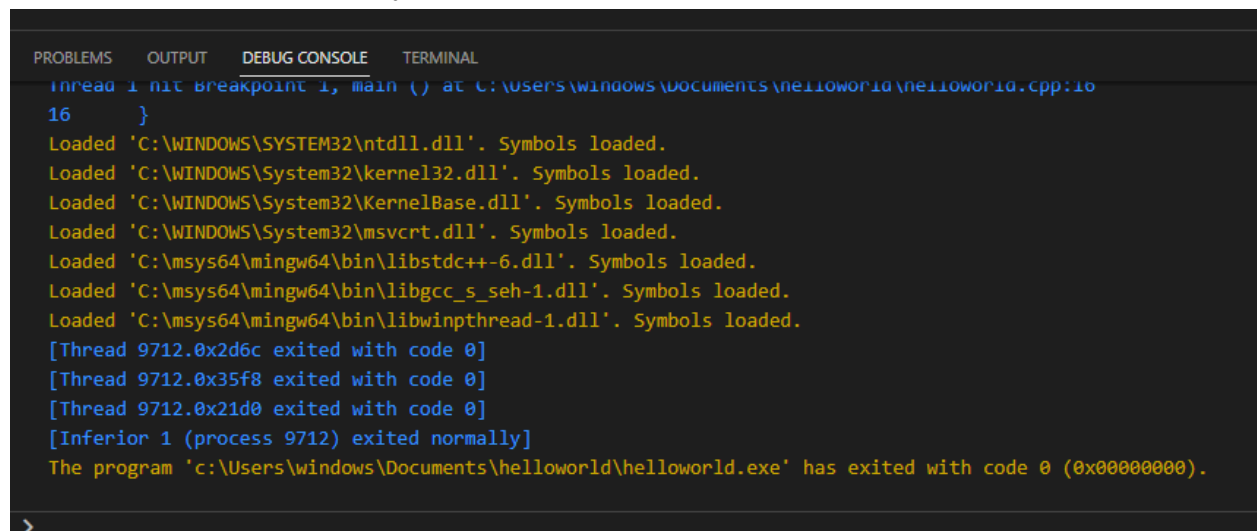
- **4.4** Save this file.
- **4.5** Then click on the Play button on the top right of the window to run this code.



Select the first option in the prompt at the top, this will use g++.exe to compile your code.

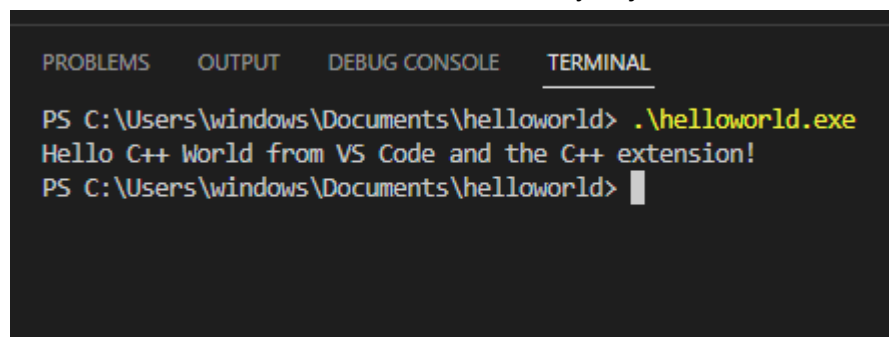


This should take less than 30 seconds but may take more depending on your hardware. When the process completes you will see output similar to this in the console.



If you encounter any errors confirm that you followed all the step in this guide and that your code matches what is in the microsoft article linked here: [Get Started with C++ and Mingw-w64 in Visual Studio Code](#)

- **4.6** Now select the terminal tab and run the .exe file you just made.



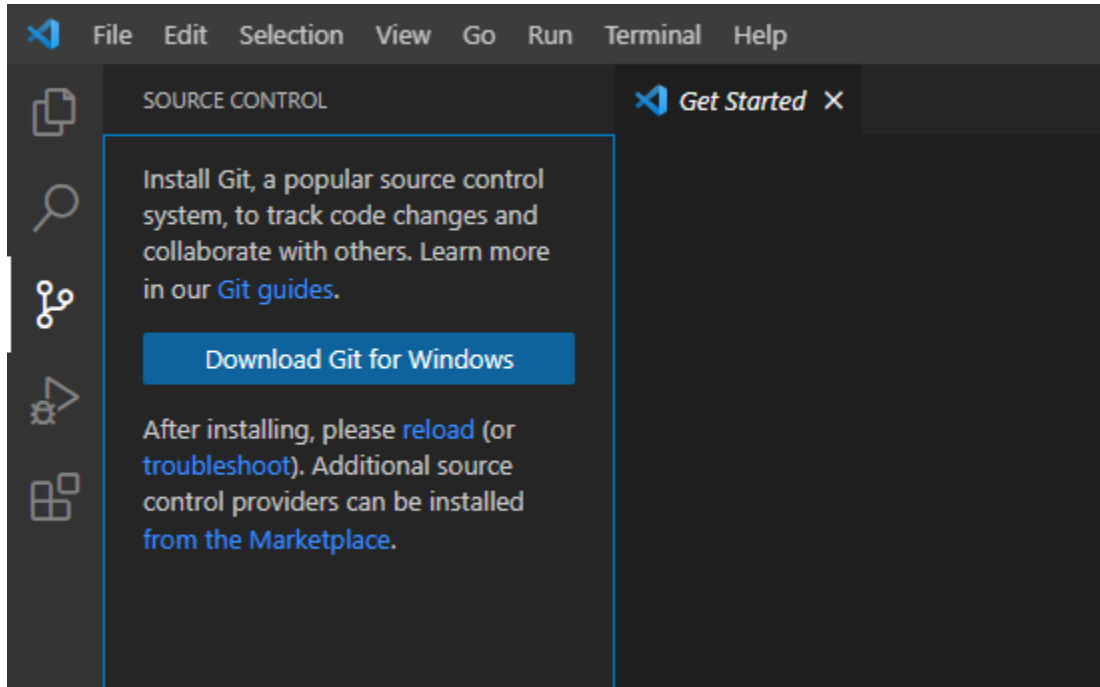
You should see the Hello World message from the code you pasted.

Now MinGW and VScode are setup for C++ development on Windows 10 or 11

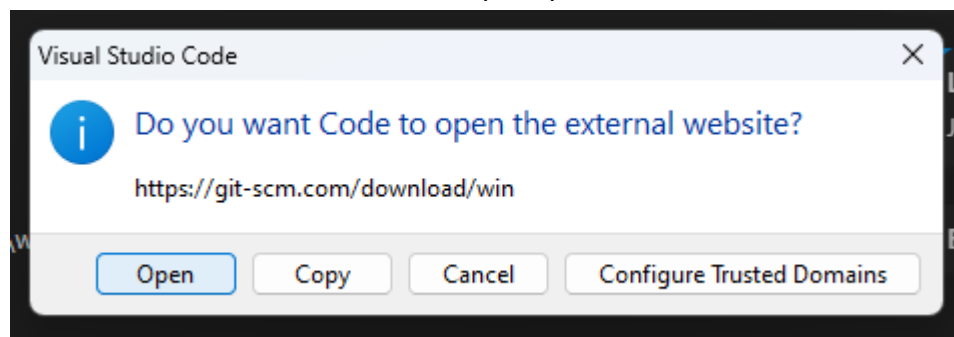
### Set 5 (Optional): Install Git

Git is a powerful and ubiquitous source control management (SCM) tool that allows for collaborative development, file and change history as well as many other features that make it an essential tool for any software developer. **While this class will not require the use of git** you may still find it useful, so this supplementary optional guide is provided should you choose to use it.

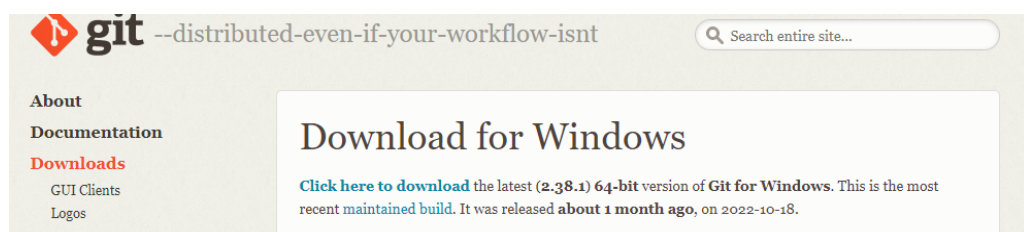
- **5.1** With VS code open select the Source control Tab (3rd icon from the top or bottom)



- **5.2** Click on the link then follow the link in the prompt. Afterwards close VS code.



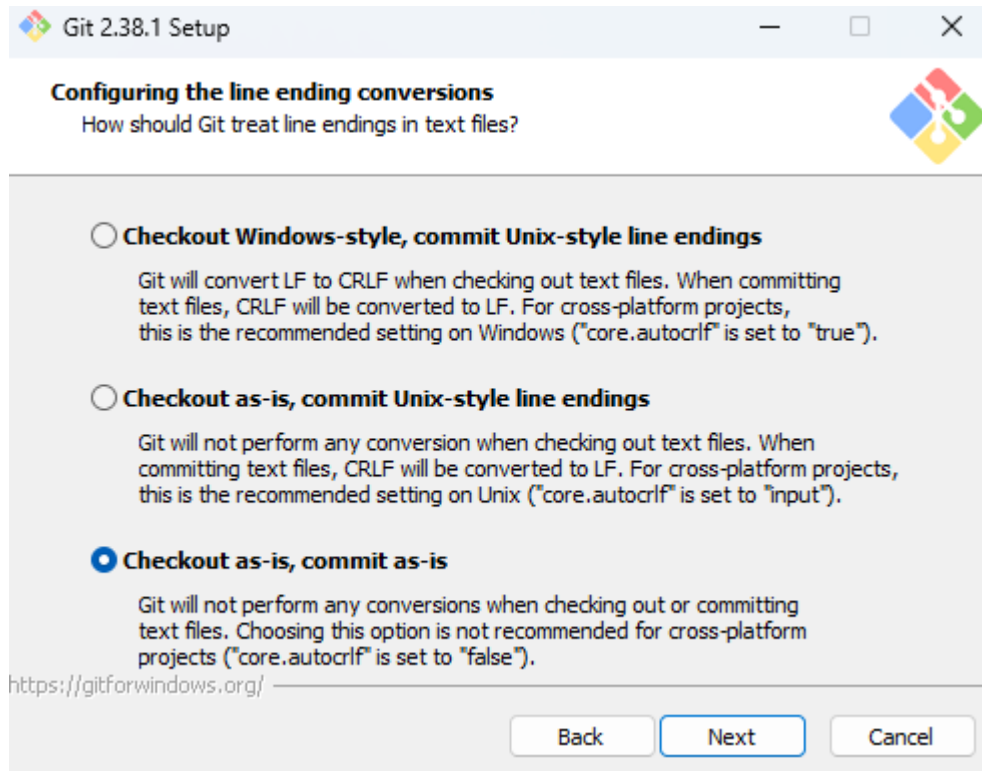
- **5.3** Next download git for windows this will be the 1st link on the page for the 64-bit version.



When Installing Git Select the default settings for all options except the following.



## - Line Endings



Choose the 3rd option for checkout & commit as-is, this will prevent issues when working with others who are on linux or macOS.

- **5.4** Git is now installed and you can use VScode's built-in git tools for working with repositories of your own or from others. This guide will not be covering authentication to remote repositories, however if you have a GitHub account you will be able to connect it to VScode directly which will resolve many of those issues.

LJ 11/26/22.