

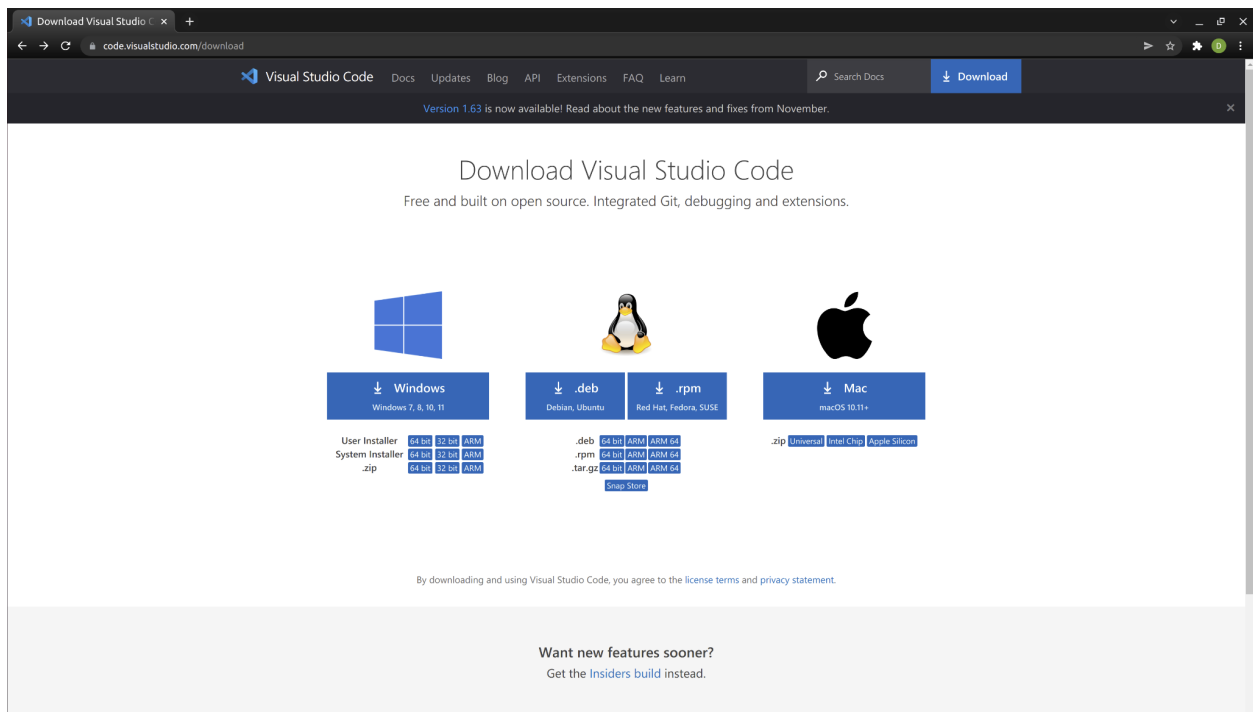
CSCI 1300 CS1: Starting Computing
Naidu/Yeh - Spring 2022
Visual Studio Code - Linux

Linux Installation Guide

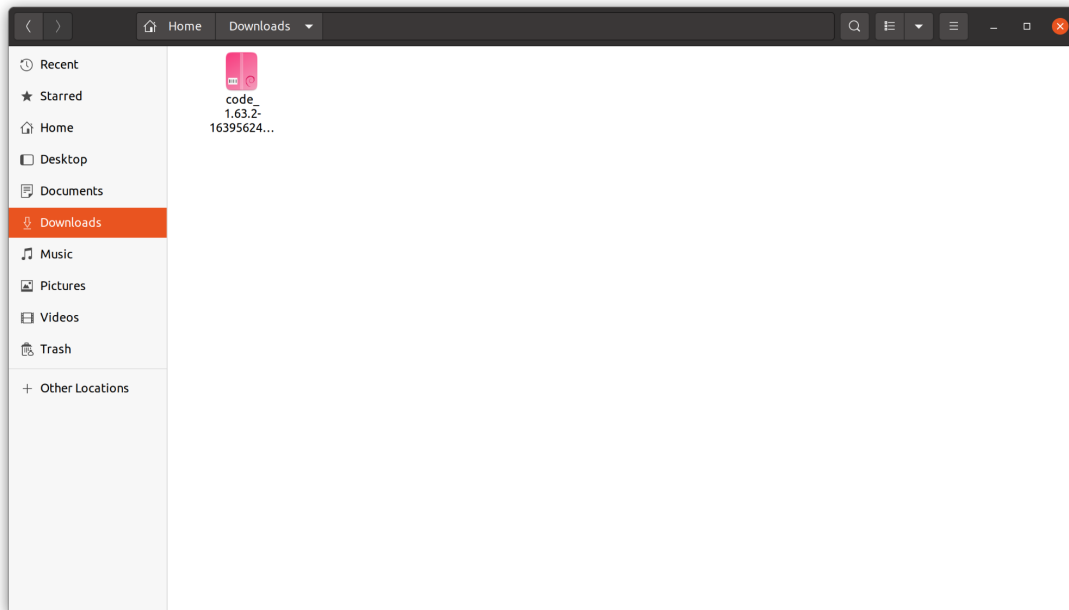
You will use Visual Studio Code (VS Code) to write and execute your programs locally. For this install guide, we'll be demonstrating the steps using Ubuntu 20.04.

Part 1- Install VS Code

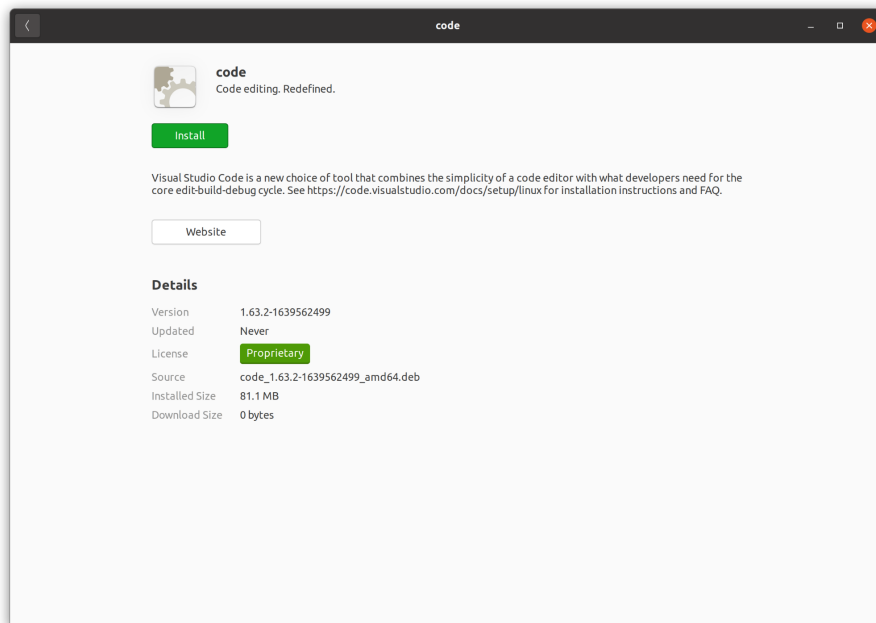
Step 1: Go to the VS code [download page](https://code.visualstudio.com/download), and download the version of VSCode for your Linux distribution.



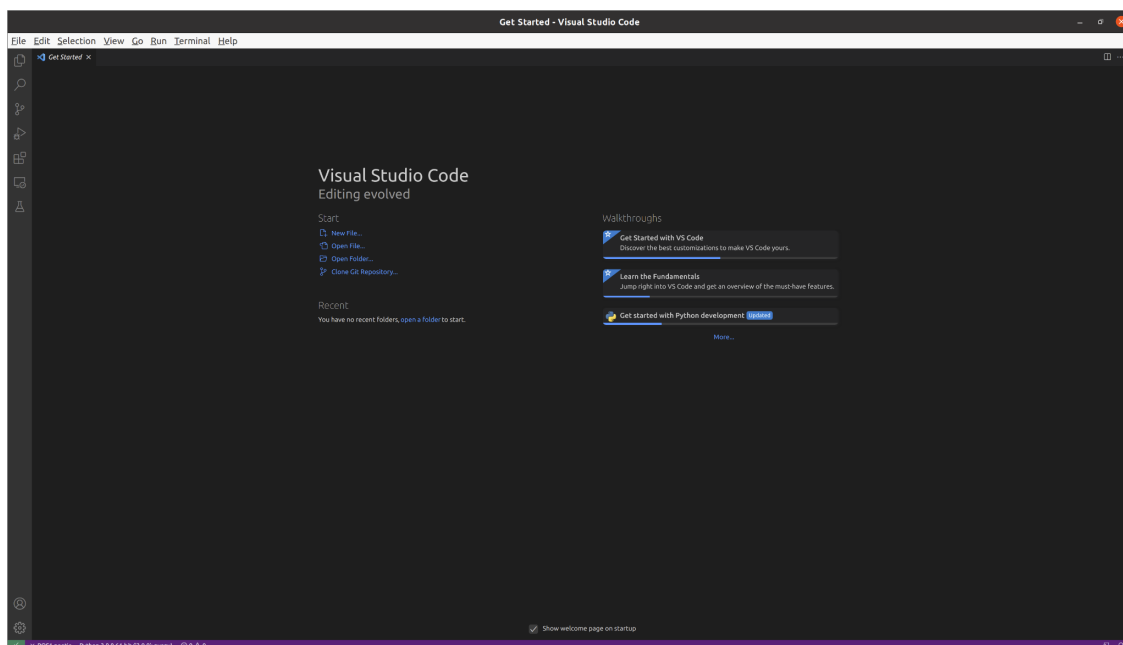
Step 2: Once downloaded, the application should look something like this in your Downloads folder. Proceed to double-click it.



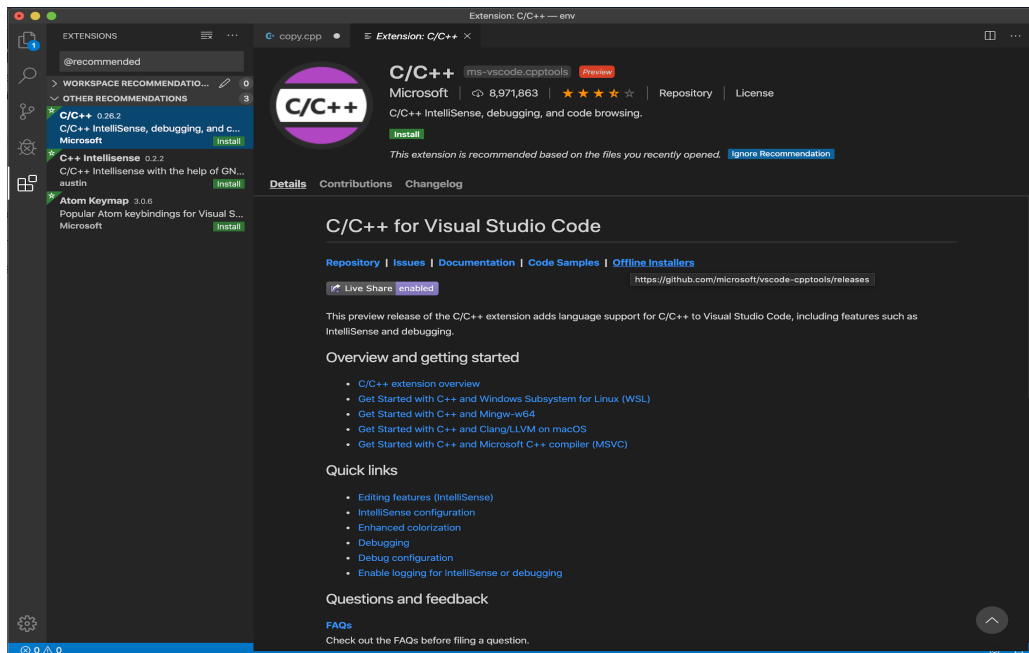
Step 3: Double-clicking the download should open the Ubuntu Software application which should then look like this. Click Install, enter your password into the prompt window and begin the installation.



Step 4: Open VSCode (you might have to search for it on your computer - you can save it to your “Favorite” apps to avoid having to do this in the future.) and it should look something like this once it’s open.

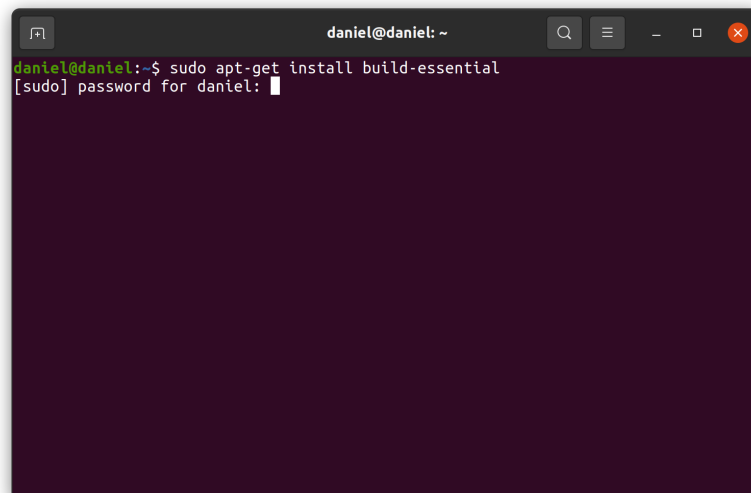


Step 5: Install C/C++ extension. In the toolbar on the left hand side of the screen click on the bottom icon for Extensions. Search for C/C++ and click install.

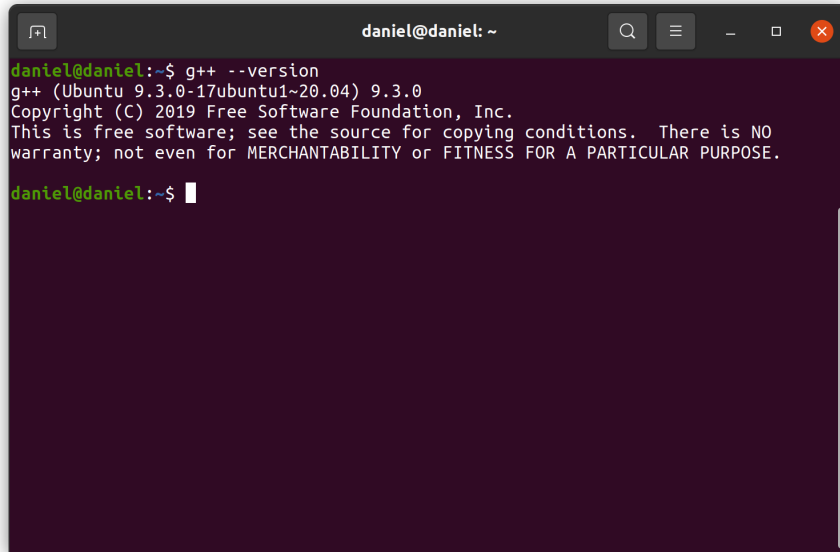


Part 2 - Install g++

1. Open a Terminal window by pressing “Ctrl + Alt + t” or by searching for it.
2. In the Terminal window, type “**sudo apt-get install build-essential**” and press Enter to install the g++ compiler along with the supporting tools. Enter your password when prompted - Linux by default won't make it visible as it's being typed in the terminal. Don't worry though, just type your password in, hit Enter and it should work! The terminal window should look something like this:



3. We'll quickly verify that g++ was installed correctly by running the “**g++ --version**” command in the same terminal. Running that command should print something that looks like this:

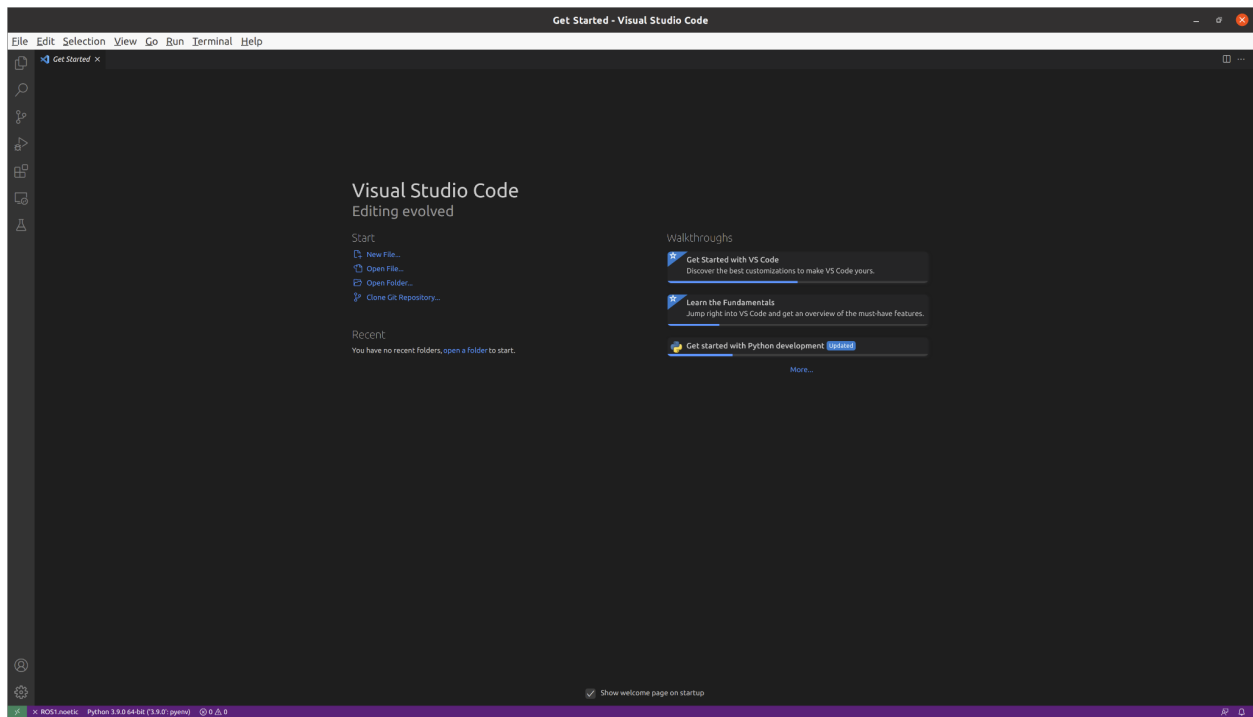
A terminal window with a dark purple background and a title bar that reads 'daniel@daniel: ~'. The terminal shows the command 'g++ --version' being executed. The output is: 'g++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0', 'Copyright (C) 2019 Free Software Foundation, Inc.', and 'This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.' The prompt 'daniel@daniel:~\$' is shown at the bottom with a cursor.

```
daniel@daniel:~$ g++ --version
g++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
daniel@daniel:~$
```

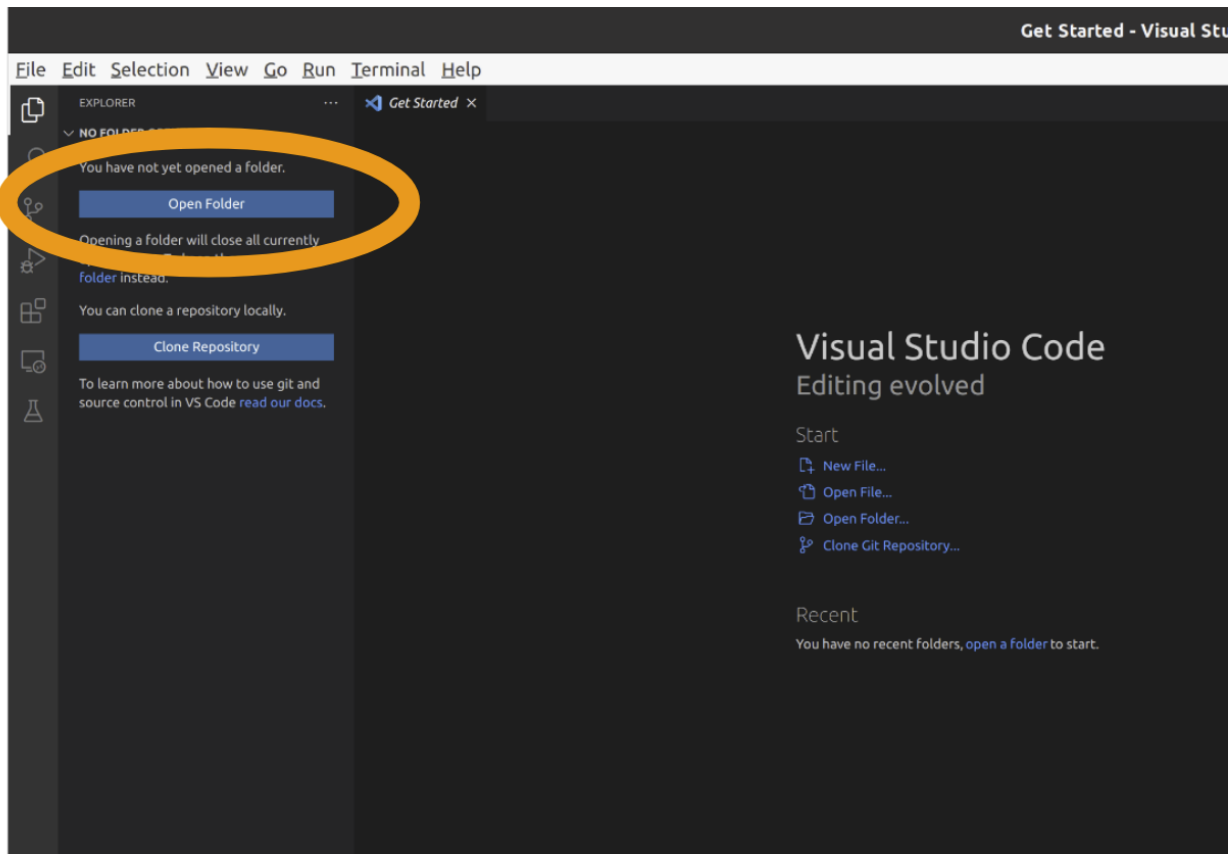
Well done! If you had any questions or issues during any of these steps please reach out to one of the TAs.

Compiling Code In Terminal

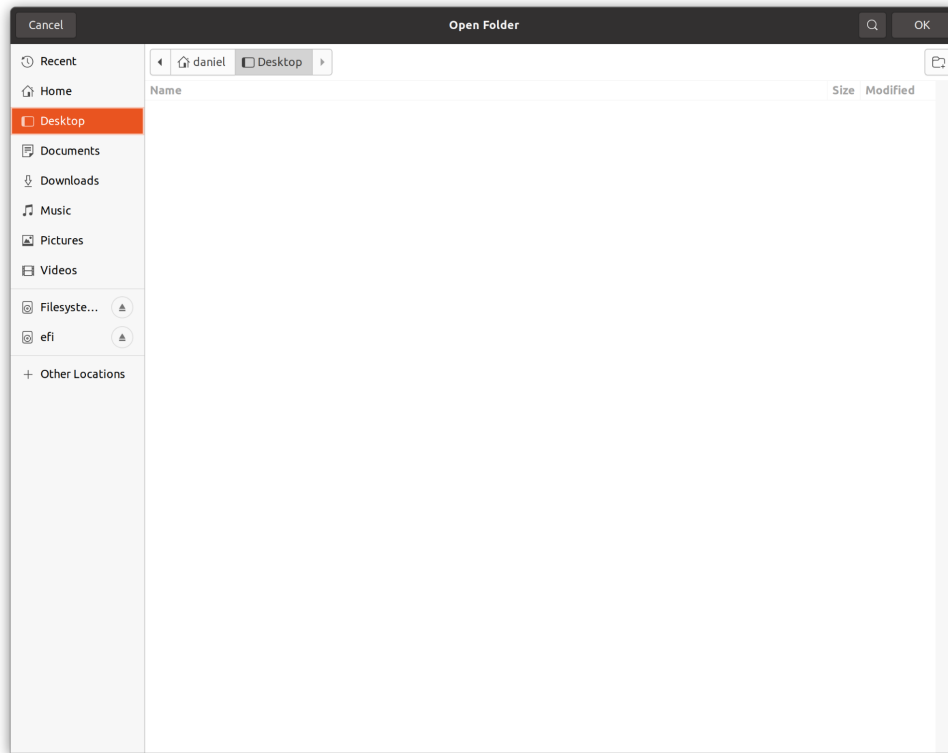
Now we'll go through a quick example on how to compile a .cpp file and verify that there were no issues with the installation. We'll start from the Welcome Page that shows up when VSCode is opened.



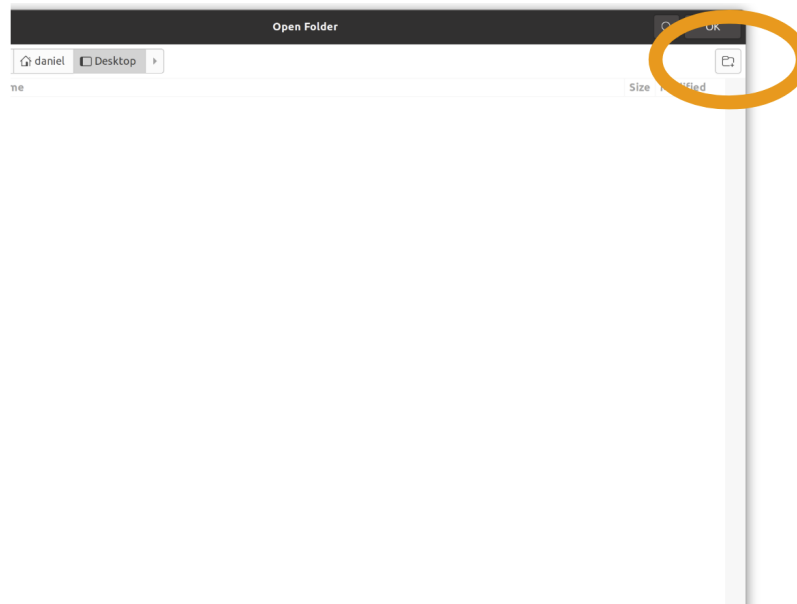
VSCode allows you to open and work out of folders where you have stored your files. Before we proceed to create a test file, we'll first create a directory on the Desktop where we'll be saving our work. First, go to the top left corner and select the Explorer icon (it looks like a folder and at the top) from the toolbar on the left. That should open the side menu which should look something like this.



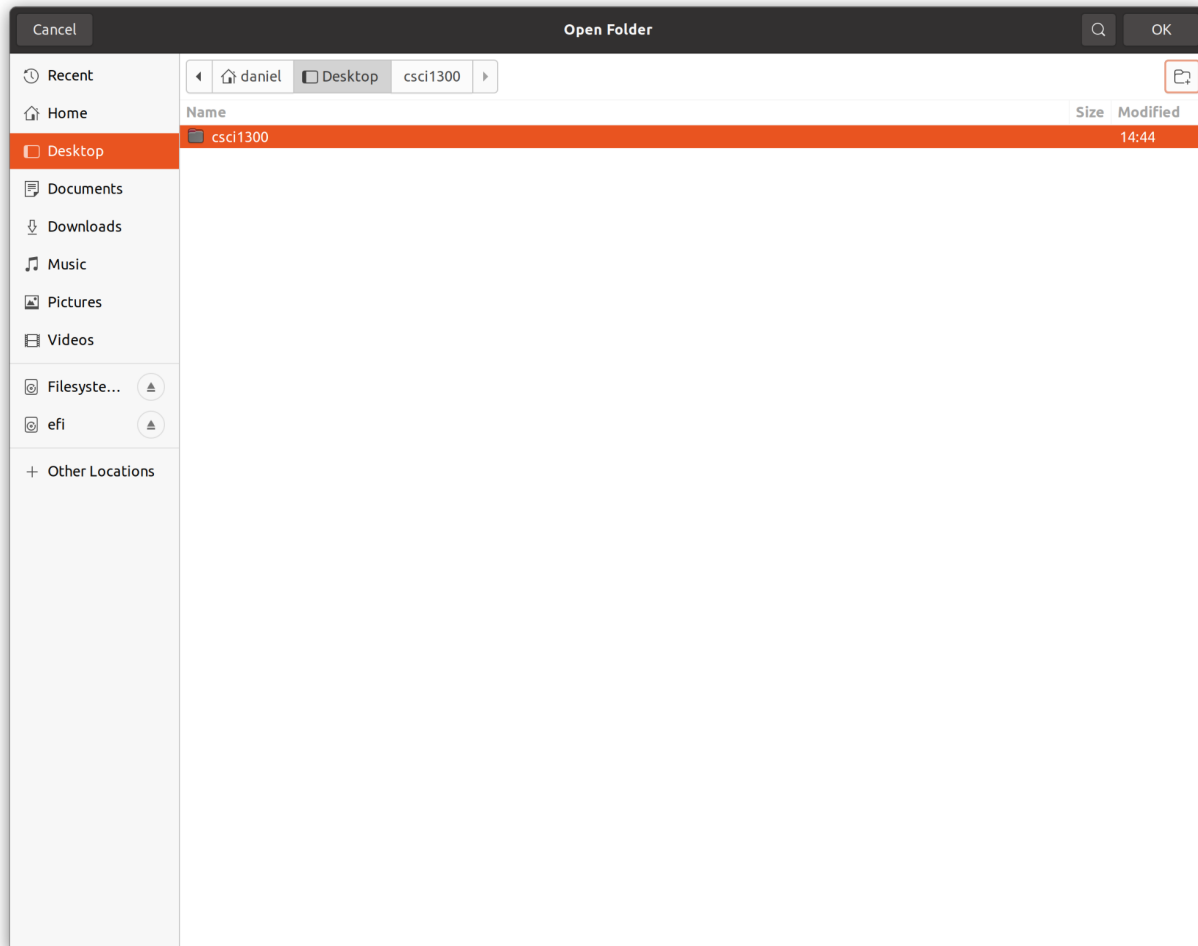
Here, select the Open Folder button which should open a File Explorer window. Navigate to the Desktop tab and select it - it should look something like the image below. In this case, my Desktop is empty though yours might not be.



Now we can create a directory in which we can store this test file. You can do so by navigating towards the top right of the image and selecting the folder icon with a plus sign. It should look something like this.

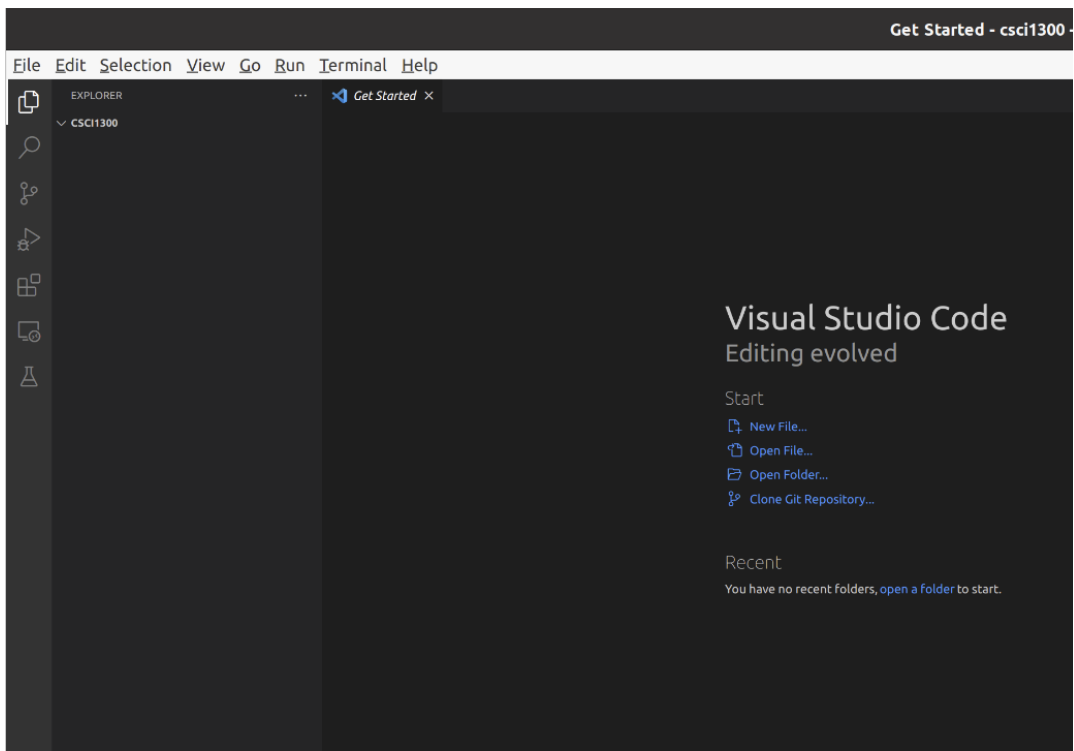


Selecting the create directory button should give you a prompt to enter the name of the directory you wish to create. I'll be naming it "csci1300". Once you've created the folder, your File Explorer should look something like this.

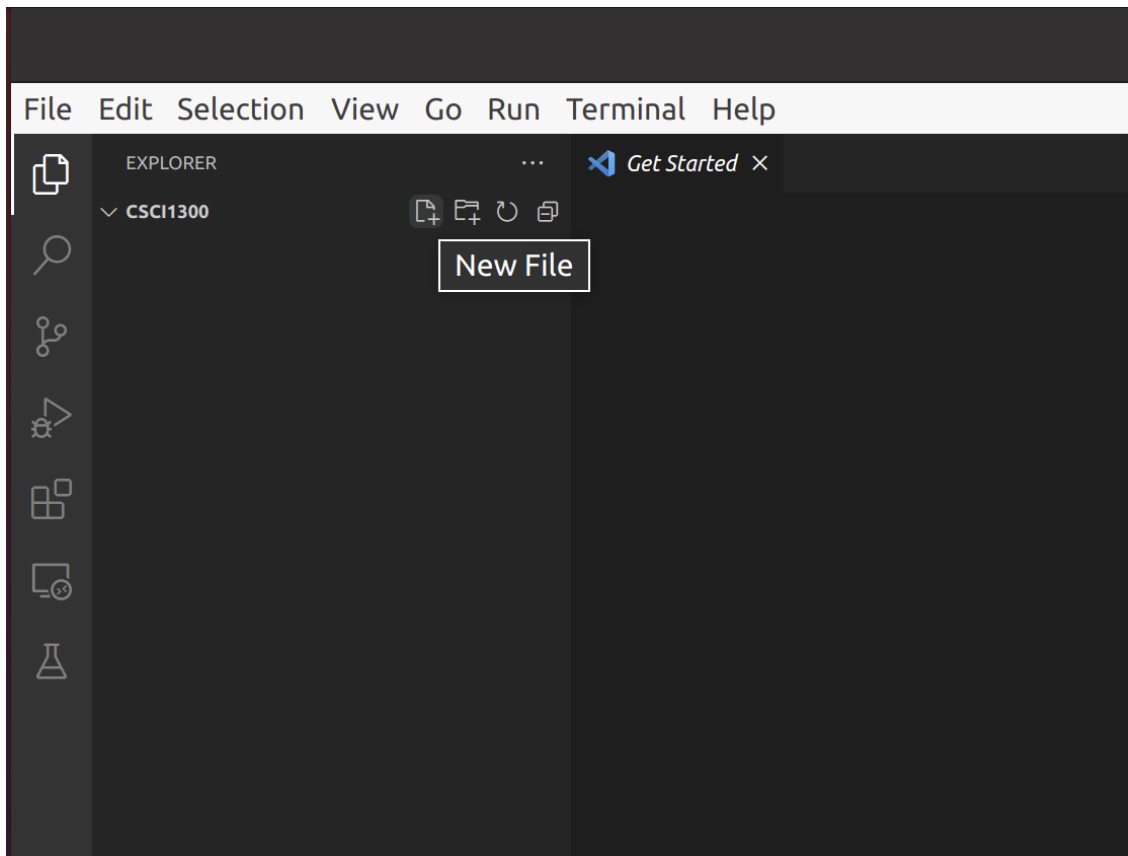


With the directory selected (highlighted in orange as shown above), click the OK button towards the top right of the window.

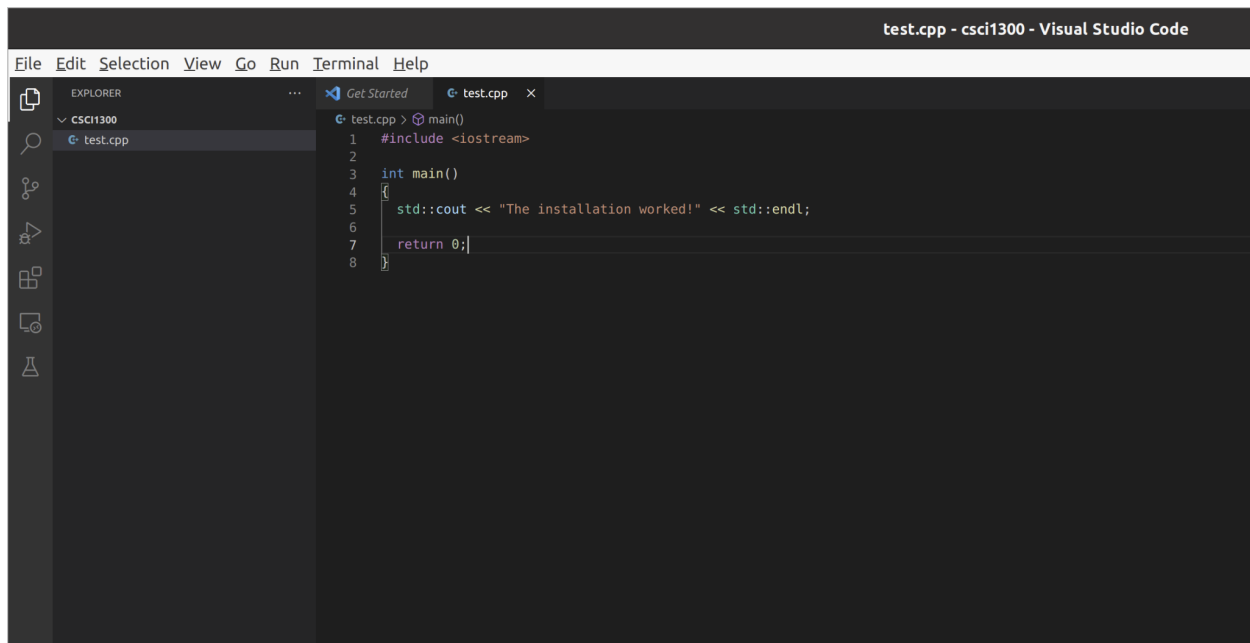
The Explorer tab of your VSCode window should now look like this.



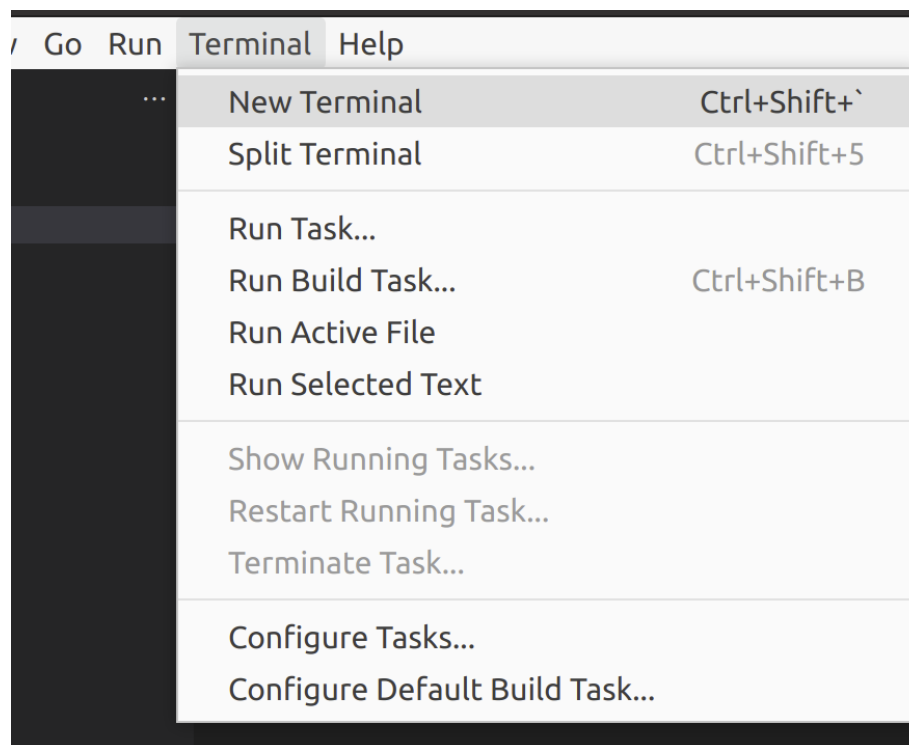
Now to create our test file, we'll navigate towards the top left and select the New File button. It should look like this when the mouse hovers over it.



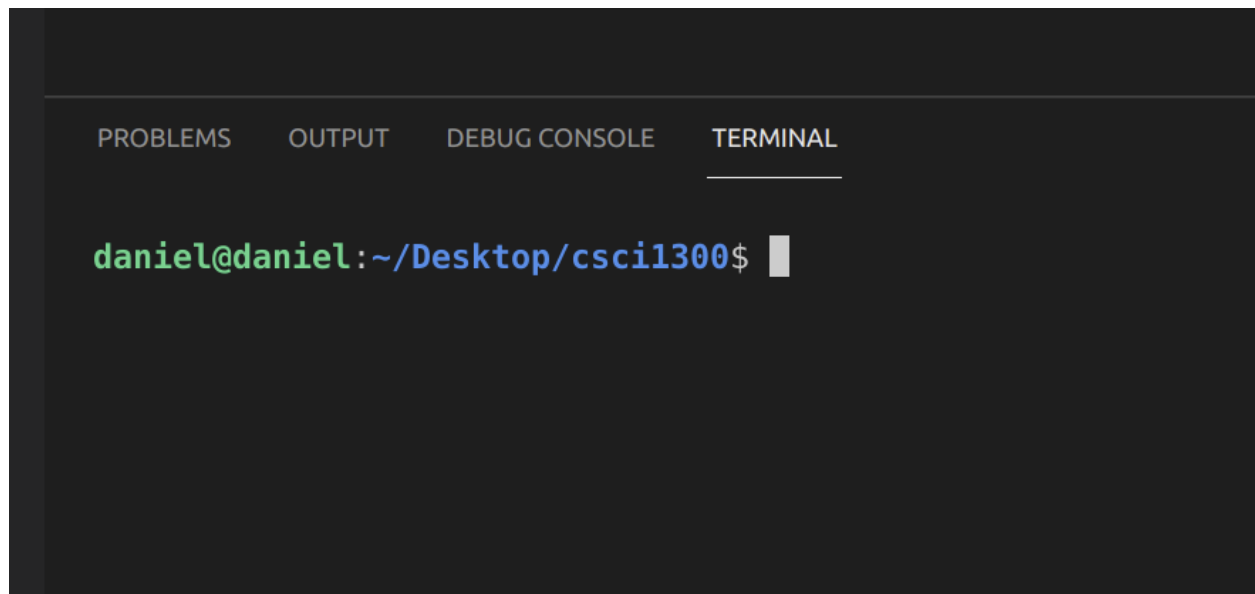
Select the New File button - this should create a prompt for you to enter the name of the test file. In this case, I'll be naming it **test.cpp**. We'll write a quick program that will print out a statement to the terminal. Once that's done, the top left of your VSCode window should look something like this.



Now that we have created the test.cpp file, we need to open a terminal to compile test.cpp and verify that everything is working correctly. Navigate towards the top of the VSCode window and go to the “Terminal” menu and select the first option “New Terminal”.



That should open a terminal at the bottom of your screen that looks like this. VSCode should by default open it to whatever folder it is that you’re working out of.



We can verify that **test.cpp** is in this directory by using the “ls” command. In order to compile the file, type in the following command into the terminal and press Enter:

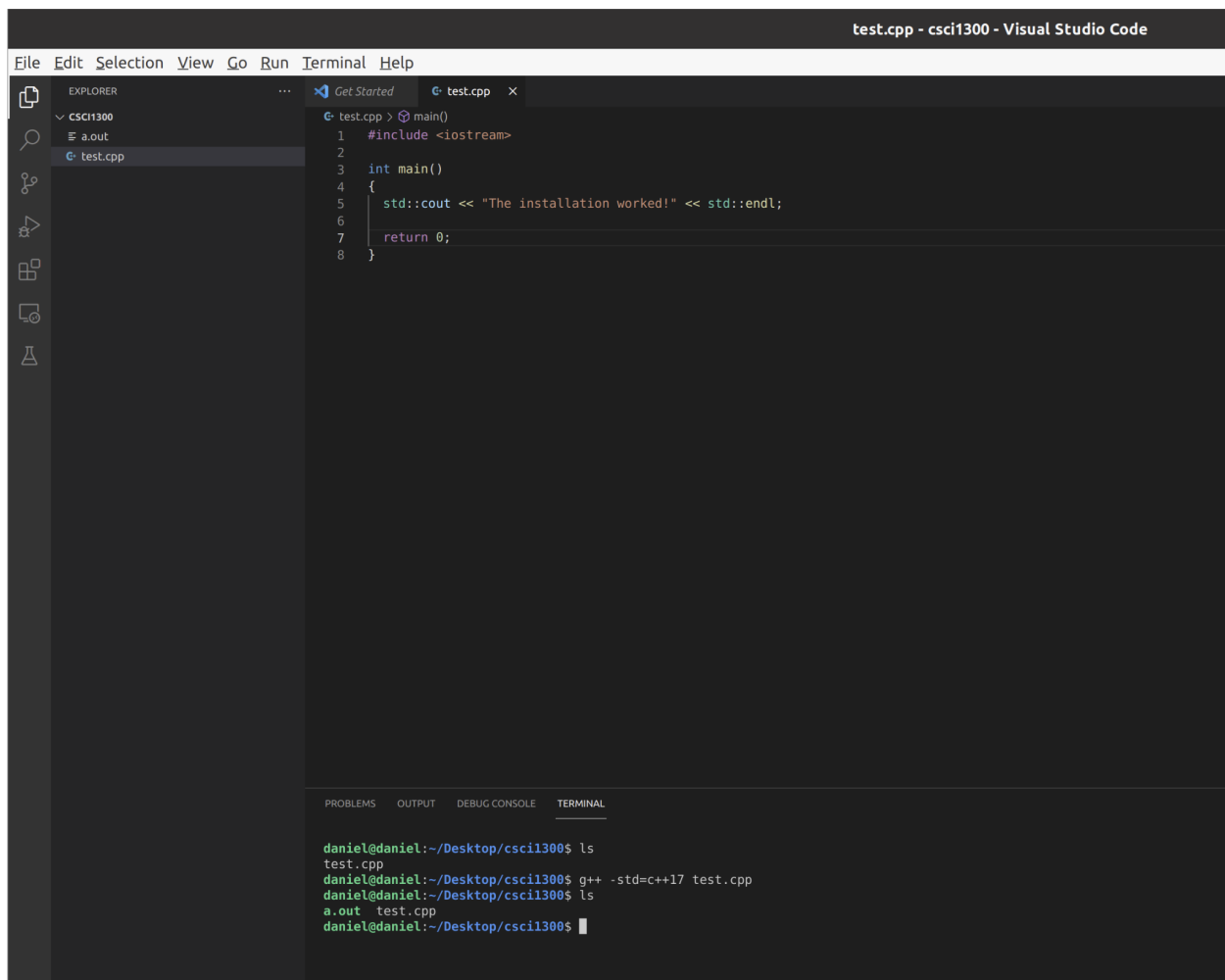
g++ -std=c++17 test.cpp

g++ is the compiler program

-std=c++17 specifies the version of C++ we want to use

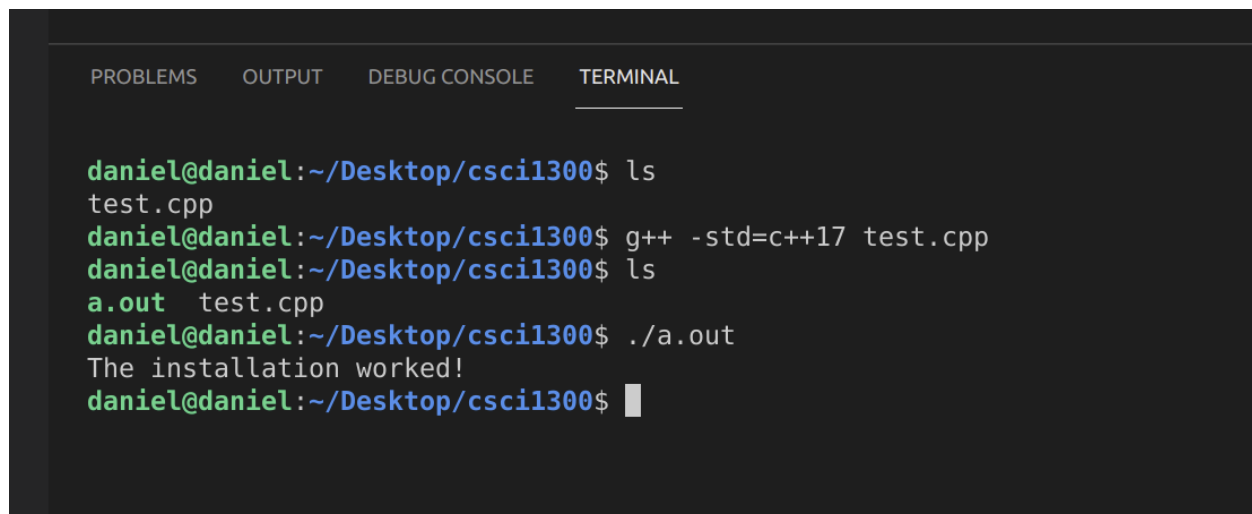
test.cpp is the file we want to compile

This command creates an executable file named **a.out** which will run the code we wrote in test.cpp. We'll once again use the “ls” command to verify that the a.out file was generated. After running the previous steps, your VSCode window and terminal should look something like this (as you can see, a.out also shows up on the left hand side in your directory).



To run the executable we created, type this into the terminal and you should get a result similar to the screenshot below:

./a.out



The image shows a terminal window with a dark background. At the top, there are four tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is selected and underlined. The terminal displays a series of commands and their outputs. The prompt is 'daniel@daniel:~/Desktop/csci1300\$'. The first command is 'ls', which outputs 'test.cpp'. The second command is 'g++ -std=c++17 test.cpp', which compiles the file. The third command is 'ls', which outputs 'a.out test.cpp'. The fourth command is './a.out', which outputs 'The installation worked!'. The prompt is then shown again with a cursor.

```
daniel@daniel:~/Desktop/csci1300$ ls
test.cpp
daniel@daniel:~/Desktop/csci1300$ g++ -std=c++17 test.cpp
daniel@daniel:~/Desktop/csci1300$ ls
a.out test.cpp
daniel@daniel:~/Desktop/csci1300$ ./a.out
The installation worked!
daniel@daniel:~/Desktop/csci1300$
```

If this worked for you, then you should be good to go and ready to start coding. If not, please reach out to one of the TAs and we will be happy to help!