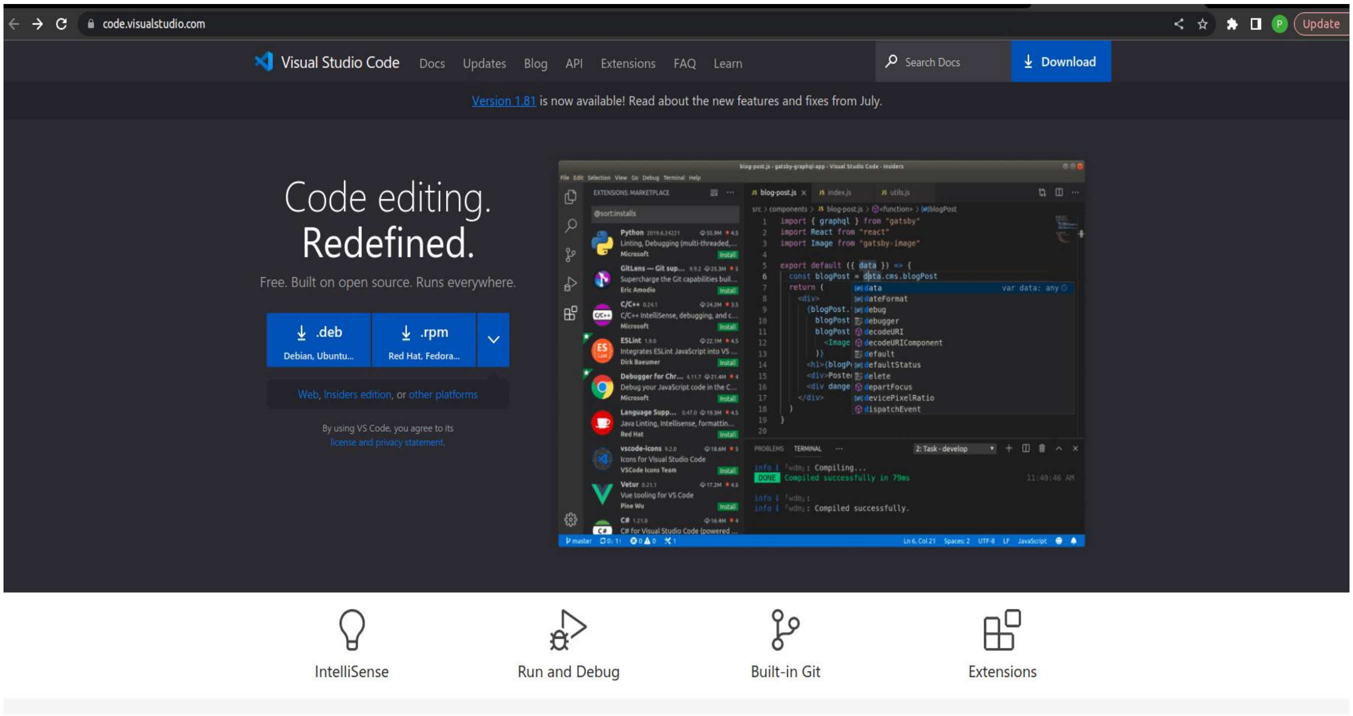
CSCI 1300 CS1: Starting Computing
Hoenigman/Naidu/Park/Ramesh - Fall 2023
Visual Studio Code - Linux

<u>Visual Studio Code installation guide for Linux</u>

You will use Visual Studio Code (VS Code) to write and execute your programs locally. For this installation 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, and download the version of VSCode corresponding to your Linux distribution.



Version 1.81 is now available! Read about the new features and fixes from July.

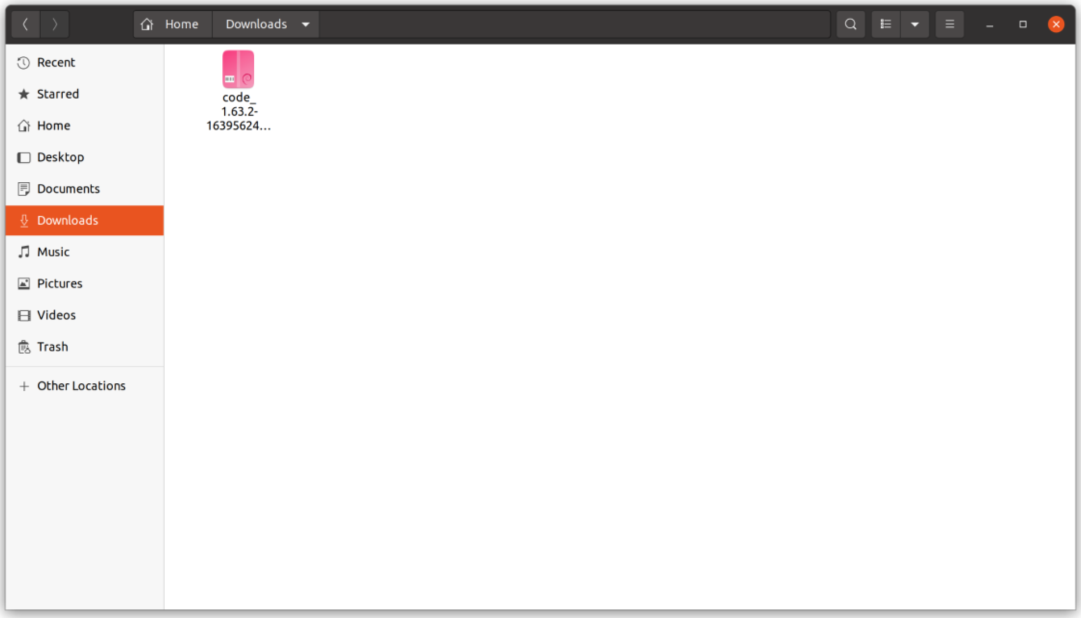
Free. Built on open source. Runs everywhere.

By using VS Code, you agree to its license and privacy statement.

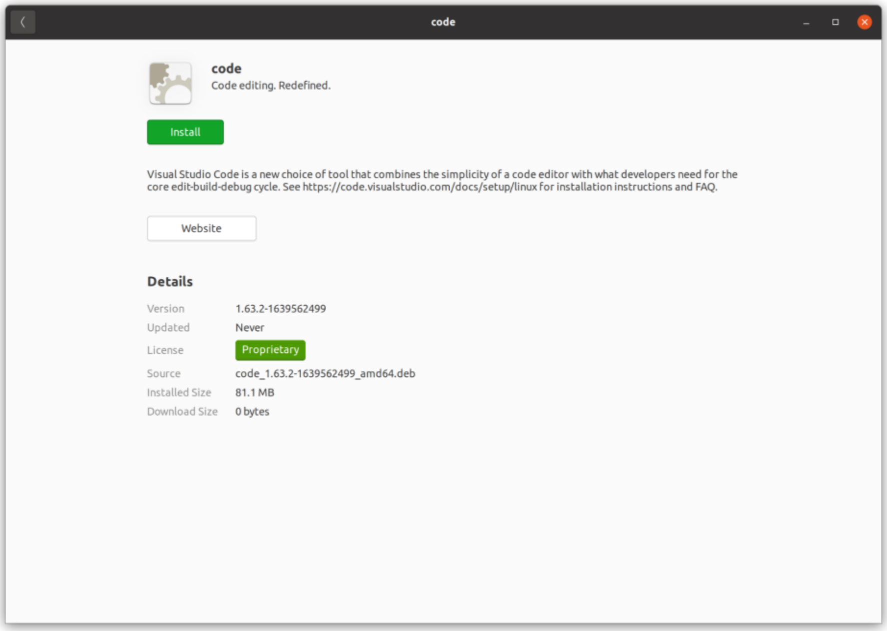
```
import { graphql } from "gatsby"
import React from "react"
import Image from "gatsby-image"

export default ({ data }) => {
  const blogPost = data.cms.blogPost
  return (
```

```
info ℹ ｢wdm｣: Compiling...
DONE Compiled successfully in 79ms                    11:40:46 AM

info ℹ ｢wdm｣:
info ℹ ｢wdm｣: Compiled successfully.
```

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 file should open the Ubuntu Software application, which should then look like this. Click on Install, enter your password into the prompt window tobegin the installation.



Visual Studio Code is a new choice of tool that combines the simplicity of a code editor with what developers need for the core edit-build-debug cycle. See https://code.visualstudio.com/docs/setup/linux for installation instructions and FAQ.

Details

Version 1.63.2-1639562499
Updated Never
License Proprietary
Source code_1.63.2-1639562499_amd64.deb
Installed Size 81.1 MB
Download Size 0 bytes

OR

Step 3: If the Ubuntu software application doesn't open up after double-clicking, you can use the terminal to install the VS code application.

Open the terminal by searching for it in 'Applications' or using the shortcut 'Ctrl+Alt+t'. Navigate to the folder where you downloaded the .deb file using the following command:

cd ~/Downloads

Install the downloaded package using the appropriate package manager command for your distribution.

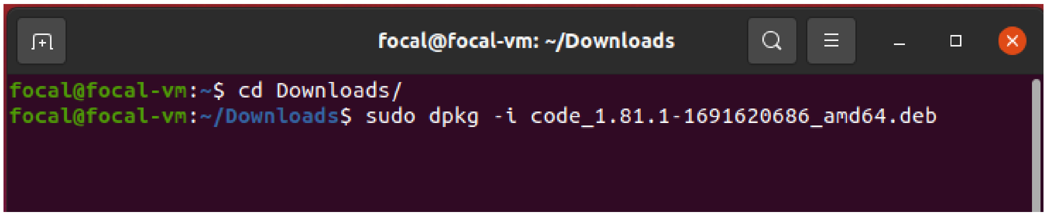
For Debian/Ubuntu-based systems (.deb package):

sudo dpkg -i <package-name>.deb

For Red Hat/Fedora-based systems (.rpm package):

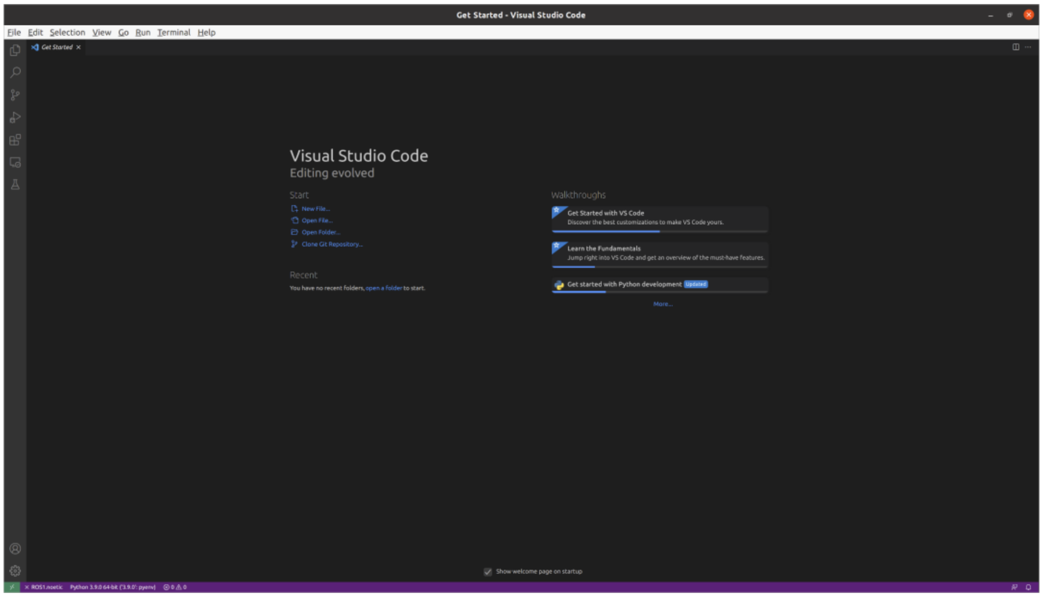
sudo rpm -i <package-name>.rpm

Replace *<package-name>* with the actual name of the downloaded package file.

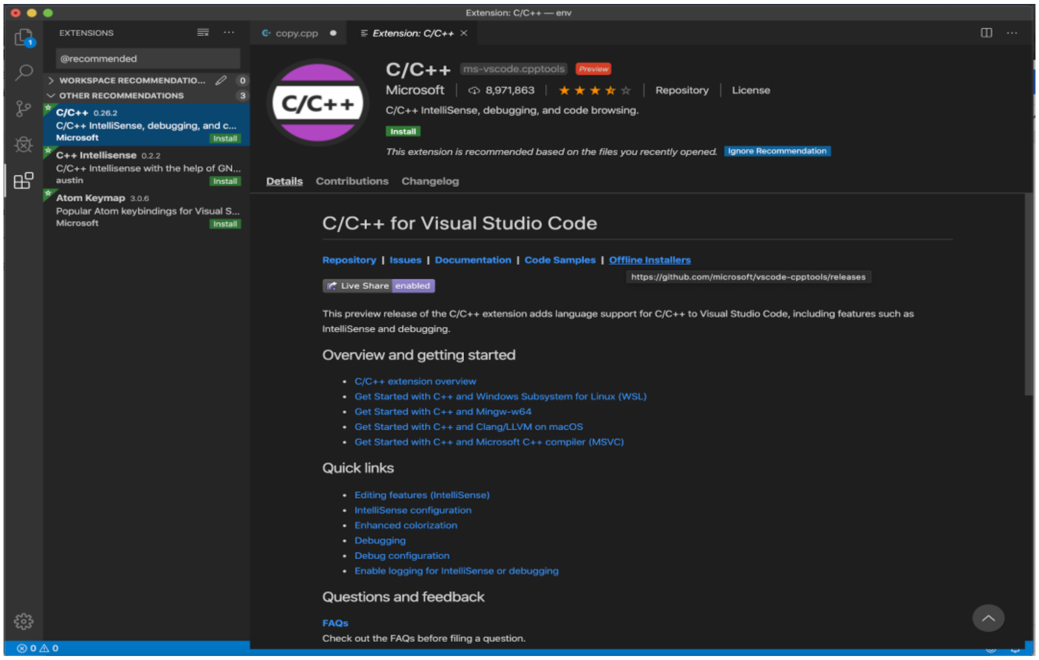


```
focal@focal-vm:~$ cd Downloads/
focal@focal-vm:~/Downloads$ sudo dpkg -i code_1.81.1-1691620686_amd64.deb
```

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). It should look something like this once it's open.


Get Started - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Get Started
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Recent
You have no recent folders, open a folder to start.
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.
Get started with Python development
Updated
More...
Show welcome page on startup
ROS1.noetic
Python 3.9.0 64-bit ('3.9.0': pyenv)
0 0

Step 5: Install the 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 on install.



C/C++ IntelliSense, debugging, and code browsing.

This extension is recommended based on the files you recently opened.

C/C++ for Visual Studio Code

Repository | Issues | Documentation | Code Samples | Offline Installers

This preview release of the C/C++ extension adds language support for C/C++ to Visual Studio Code, including features such as IntelliSense and debugging.

Overview and getting started

- C/C++ extension overview
- Get Started with C++ and Windows Subsystem for Linux (WSL)
- Get Started with C++ and Mingw-w64
- Get Started with C++ and Clang/LLVM on macOS
- Get Started with C++ and Microsoft C++ compiler (MSVC)

Quick links

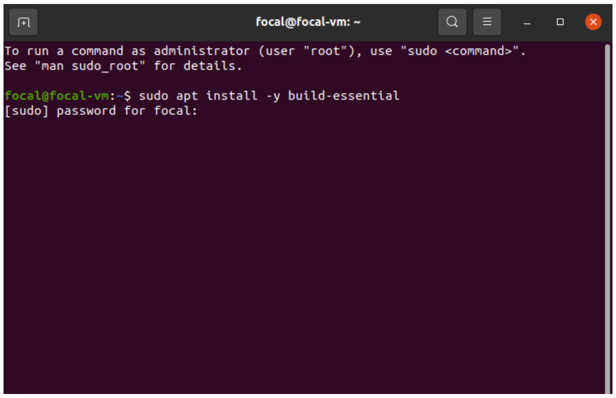
- Editing features (IntelliSense)
- IntelliSense configuration
- Enhanced colorization
- Debugging
- Debug configuration
- Enable logging for IntelliSense or debugging

Questions and feedback

FAQs
Check out the FAQs before filing a question.

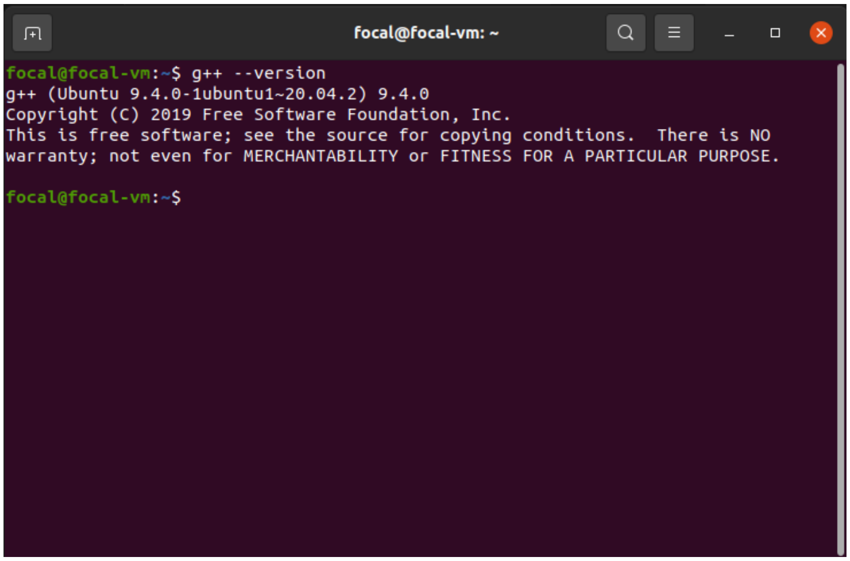
Part 2 - Install g++

1. Open a Terminal window by pressing “Ctrl + Alt + t” or by searching for it.
2. In the Terminal window, type **“sudo apt install -y 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:



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

focal@focal-vm:~$ sudo apt install -y build-essential
[sudo] password for focal:
```

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:

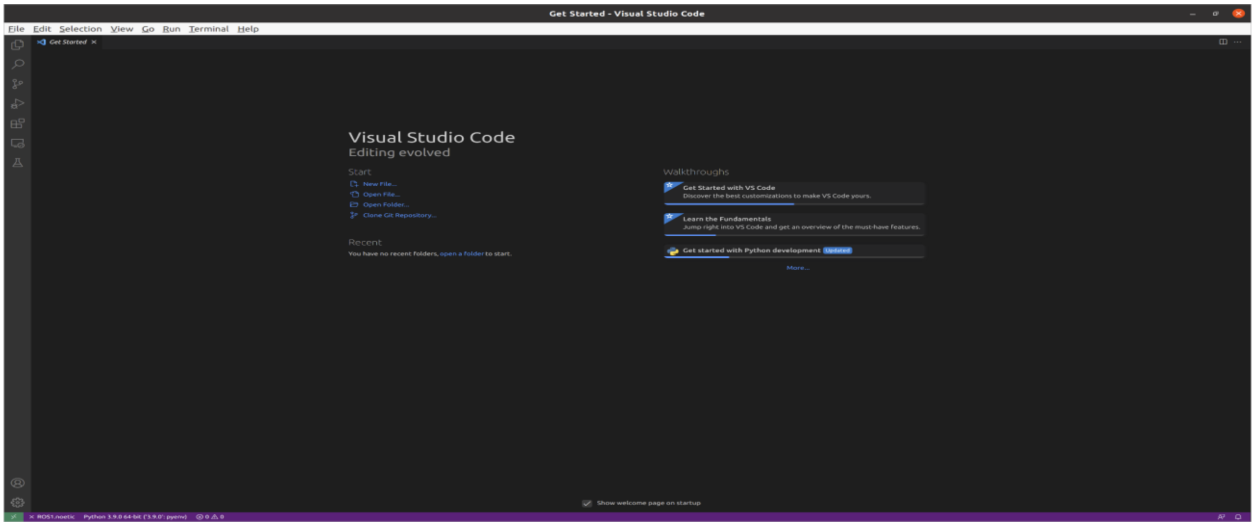


```
focal@focal-vm:~$ g++ --version
g++ (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

focal@focal-vm:~$
```

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.



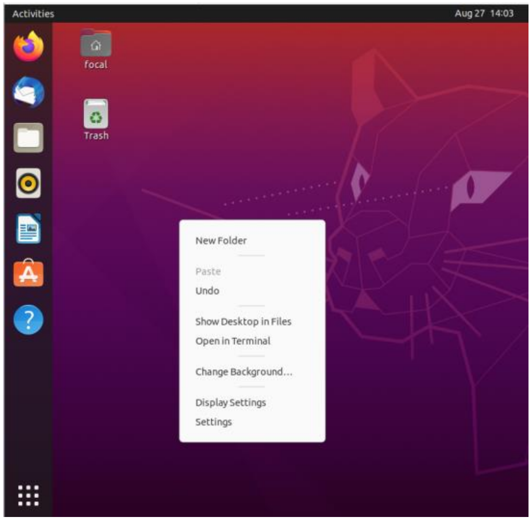
You have no recent folders, open a folder to start.

Discover the best customizations to make VS Code yours.

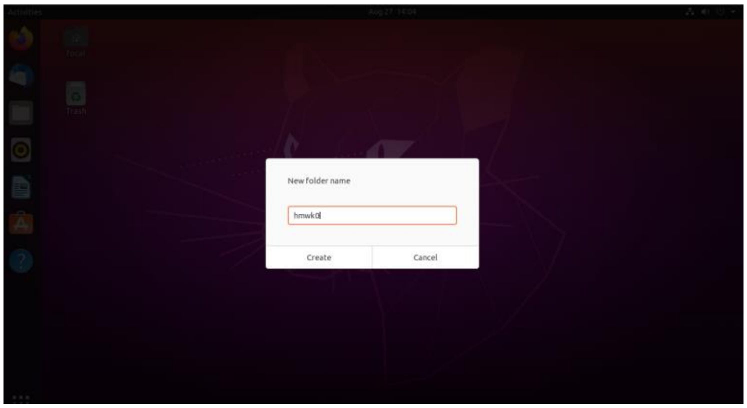
Jump right into VS Code and get an overview of the must-have features.

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. We will do this using the terminal.

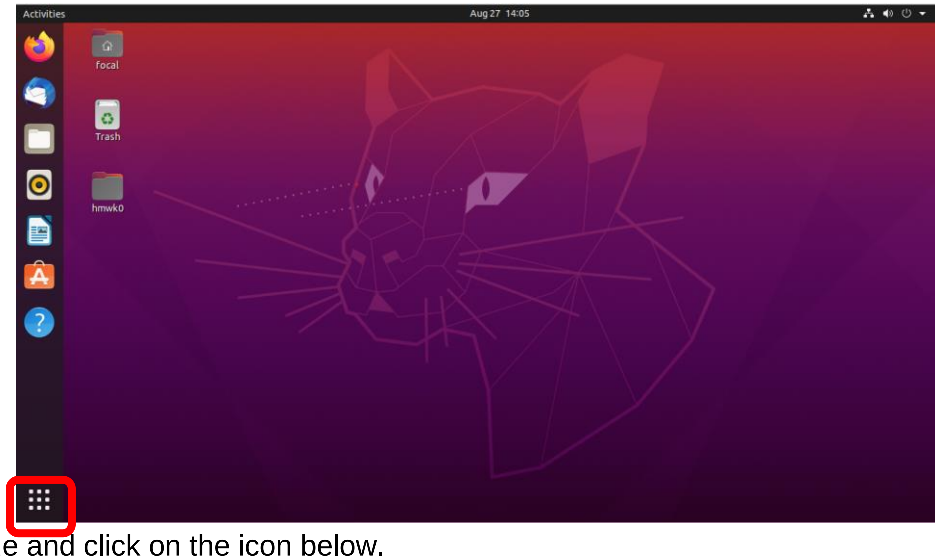
To create a directory on the desktop, right-click on desktop and select "**New Folder**"



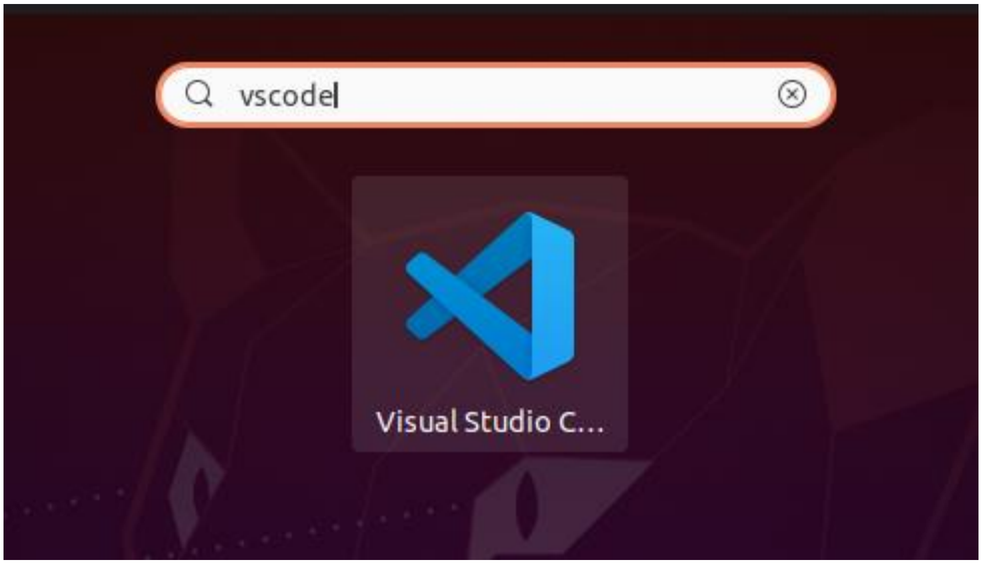
Name the folder ***hmwk0***



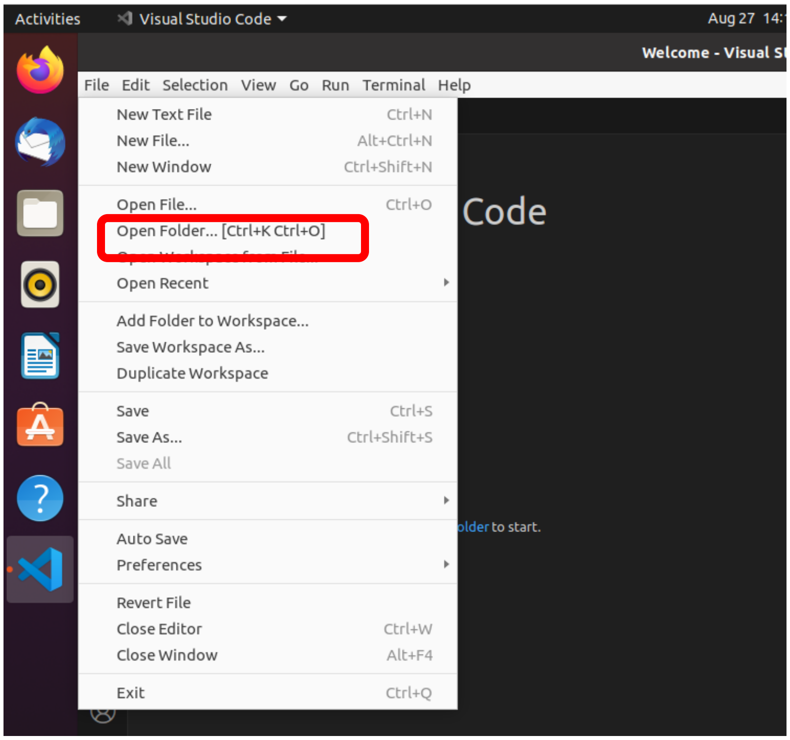
Open VS code by clicking on the 9-dot icon at the bottom left of the screen.



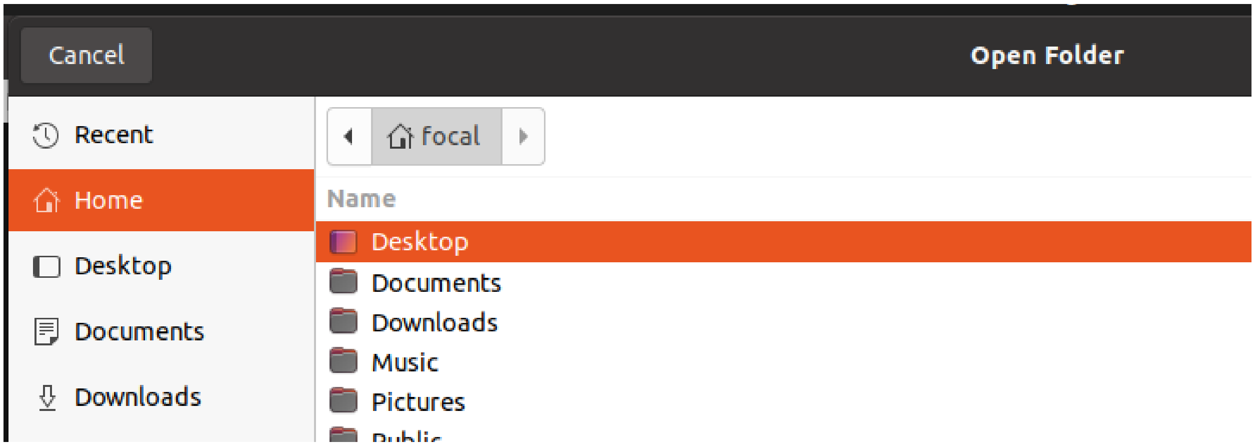
Type vscode and click on the icon below.



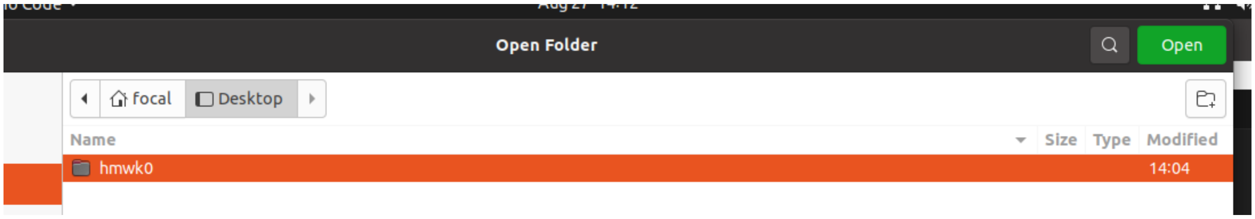
Once you open VS code, open the folder that you created via **File > Open Folder…**.



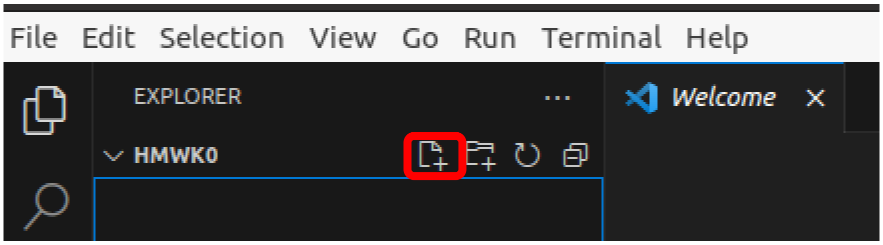
Double click on Desktop

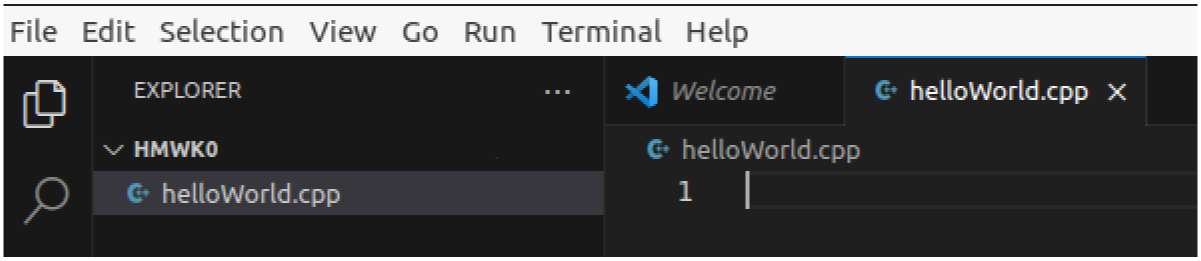


Select **hmwk0** and click **Open**



Now create a new file called helloWorld.cpp and paste the following code into it.

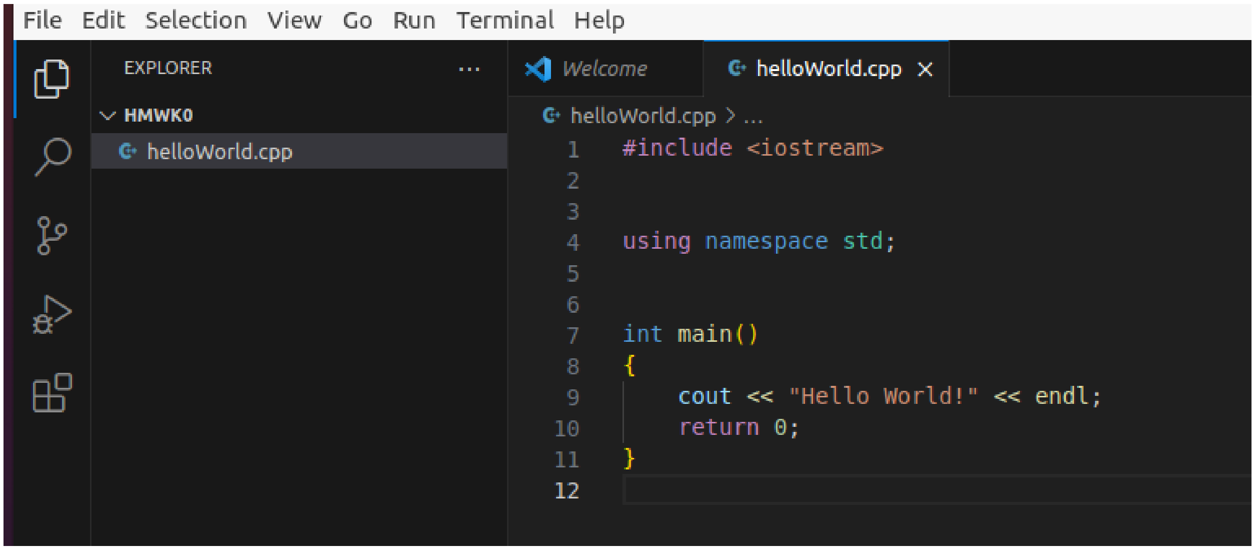




Code:

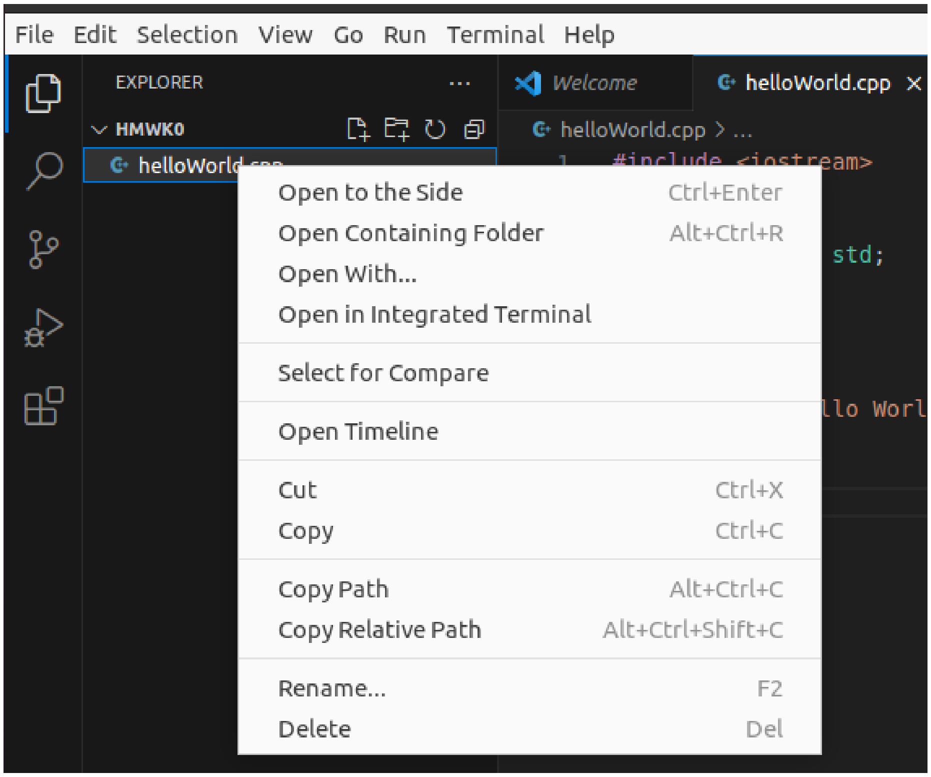
```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

Once that’s done, the top left of your VSCode window should look something like this.

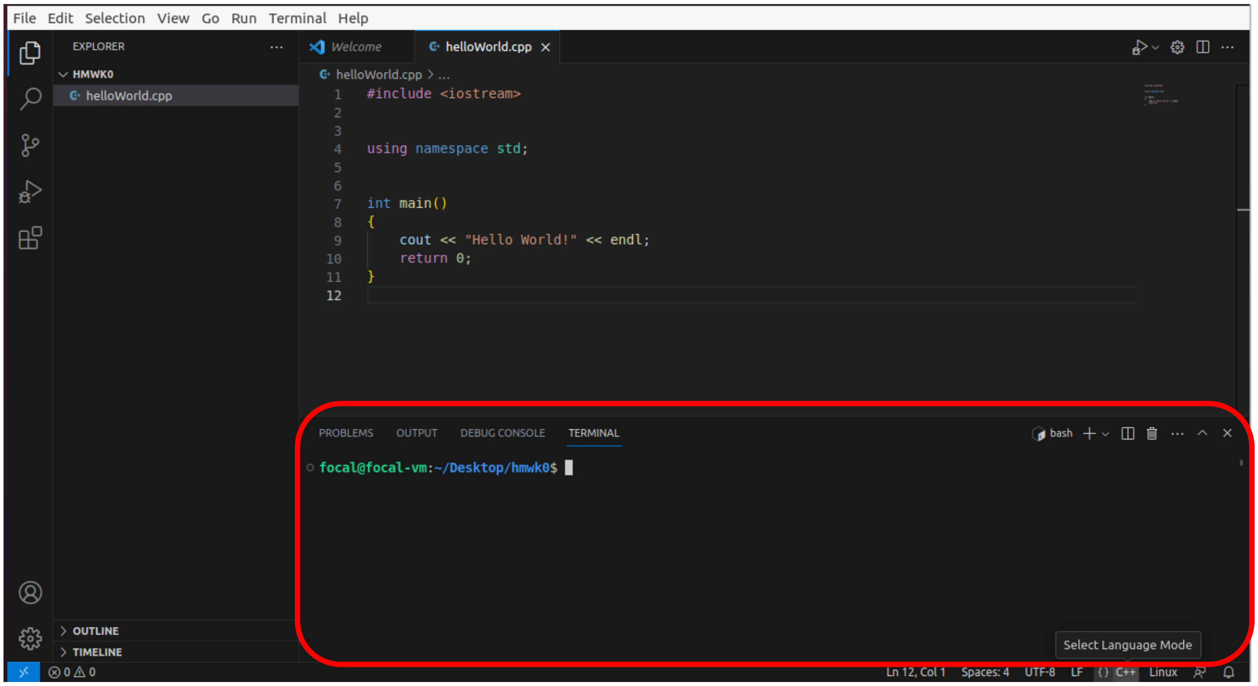


```cpp
#include <iostream>


using namespace std;


int main()
{
    cout << "Hello World!" << endl;
    return 0;
}

```

Now that we have created the **helloWorld.cpp** file, we need to open a terminal to compile **helloWorld.cpp** and verify that everything is working correctly.
On the left panel, right-click on the **helloWorld.cpp** file and select “**Open in Integrated Terminal**”.



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



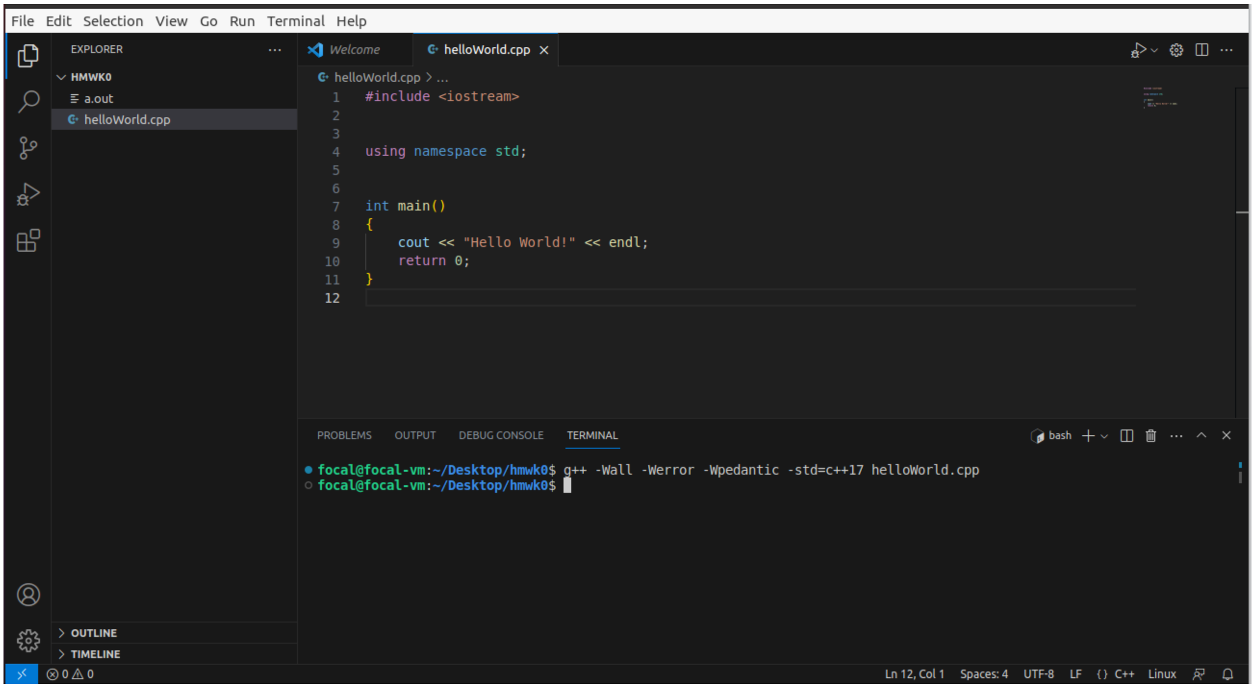
```cpp
#include <iostream>


using namespace std;


int main()
{
    cout << "Hello World!" << endl;
    return 0;
}

```

```
focal@focal-vm:~/Desktop/hmwk0$
```

In order to compile the file, type in the following command into the terminal and press Enter:
g++ -Wall -Werror -Wpedantic -std=c++17 helloWorld.cpp

g++ is the compiler program
-Wall -Werror -Wpedantic this will make sure that our code does not violate any standards
-std=c++17 specifies the version of C++ we want to use
helloWorld.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 **helloWorld.cpp**. 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).

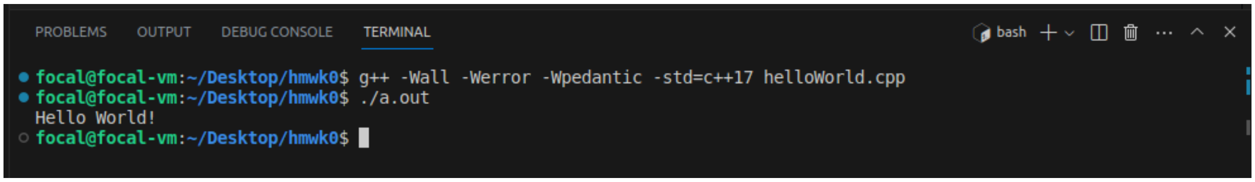


```
#include <iostream>


using namespace std;


int main()
{
    cout << "Hello World!" << endl;
    return 0;
}

```

```
focal@focal-vm:~/Desktop/hmwk0$ g++ -Wall -Werror -Wpedantic -std=c++17 helloWorld.cpp
focal@focal-vm:~/Desktop/hmwk0$ 
```

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

./a.out



```
focal@focal-vm:~/Desktop/hmwk0$ g++ -Wall -Werror -Wpedantic -std=c++17 helloWorld.cpp
focal@focal-vm:~/Desktop/hmwk0$ ./a.out
Hello World!
focal@focal-vm:~/Desktop/hmwk0$ 
```

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!