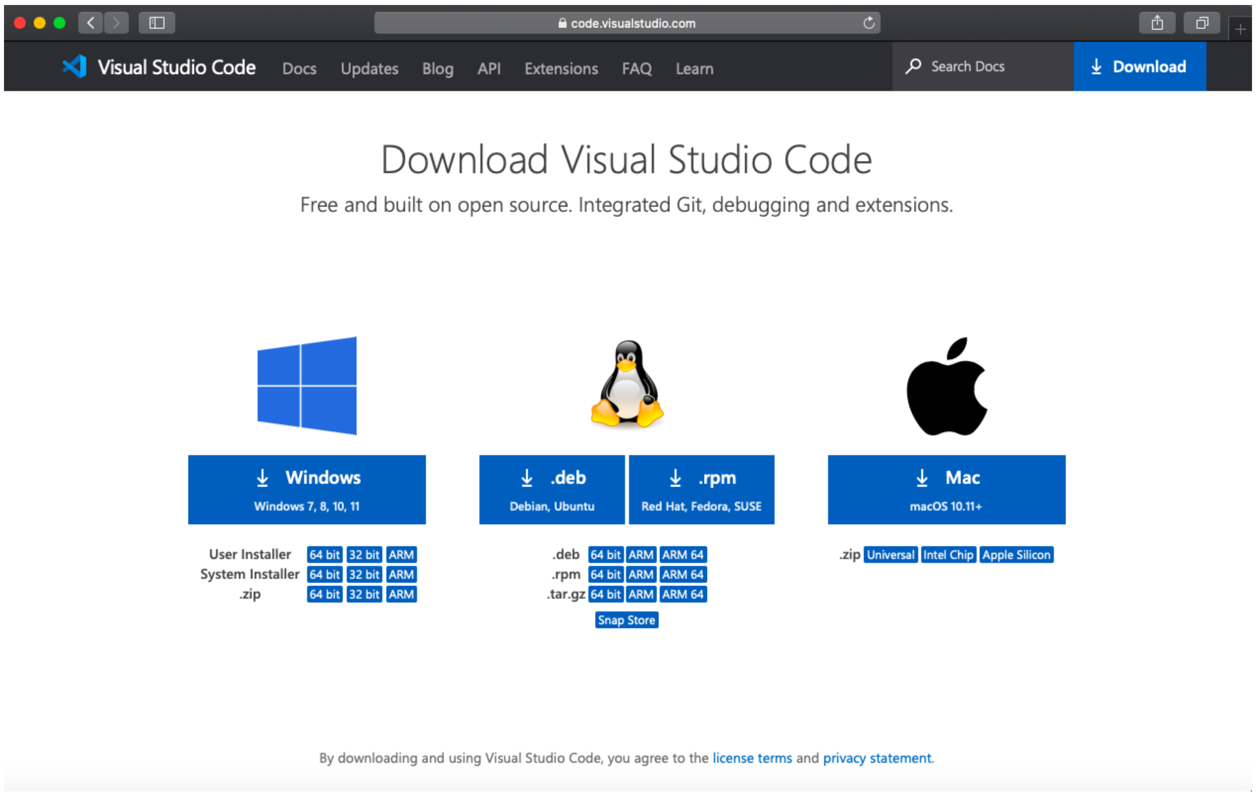
CSCI 1300 CS1: Starting Computing
Godley/Hoefer - Spring 2023
Visual Studio Code - MacOS

MacOS Installation Guide

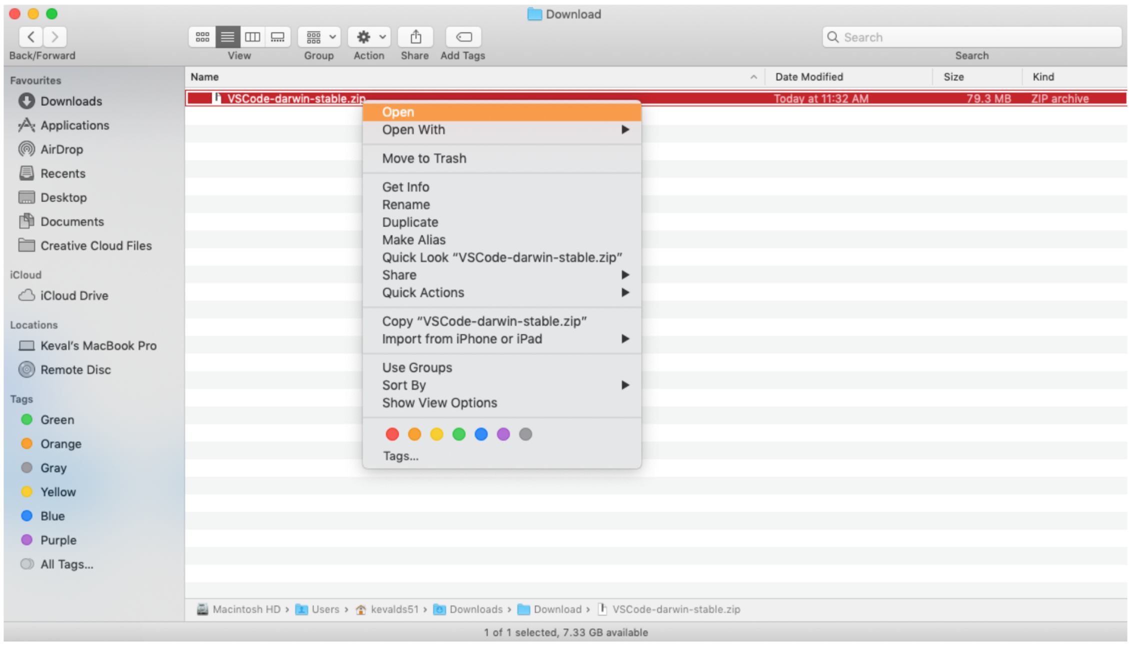
You will use Visual Studio Code (VS Code) to write and execute your programs locally.

Part 1- Install VS Code

Step 1: Go to VS code [download page](), and download for Mac.

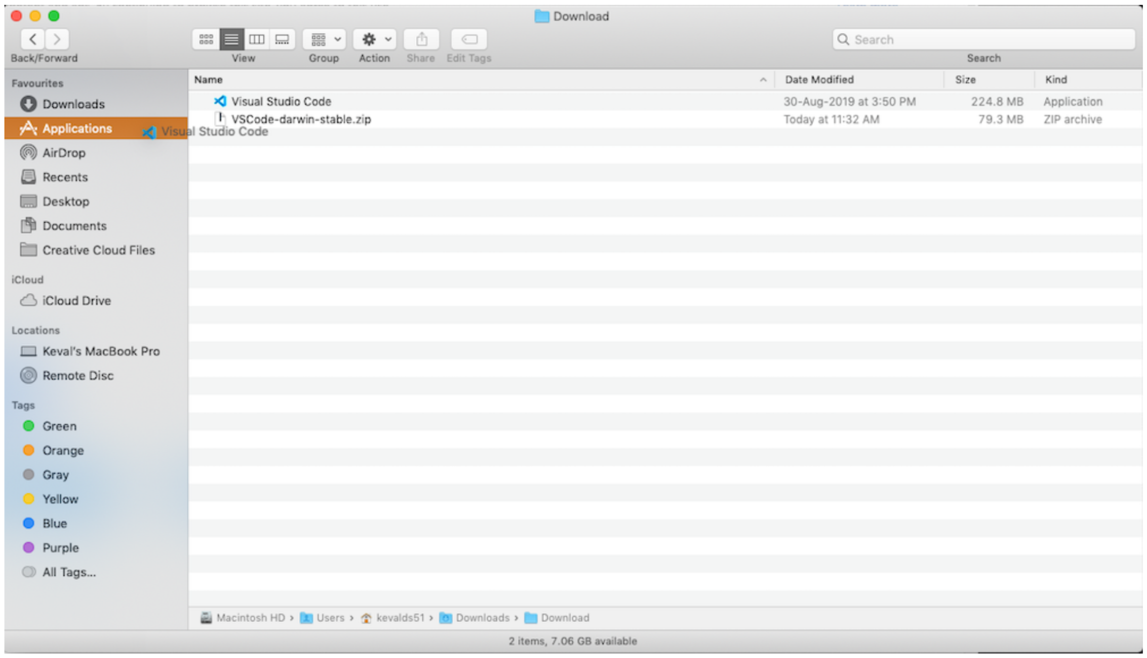

code.visualstudio.com
Visual Studio Code
Docs
Updates
Blog
API
Extensions
FAQ
Learn
Search Docs
Download
Download Visual Studio Code
Free and built on open source. Integrated Git, debugging and extensions.
Windows
Windows 7, 8, 10, 11
.deb
Debian, Ubuntu
.rpm
Red Hat, Fedora, SUSE
Mac
macOS 10.11+
User Installer 64 bit 32 bit ARM
System Installer 64 bit 32 bit ARM
.zip 64 bit 32 bit ARM
.deb 64 bit ARM ARM 64
.rpm 64 bit ARM ARM 64
.tar.gz 64 bit ARM ARM 64
Snap Store
.zip Universal Intel Chip Apple Silicon
By downloading and using Visual Studio Code, you agree to the license terms and privacy statement.

Step 2: After the download has finished, unzip the folder by double-clicking on it.



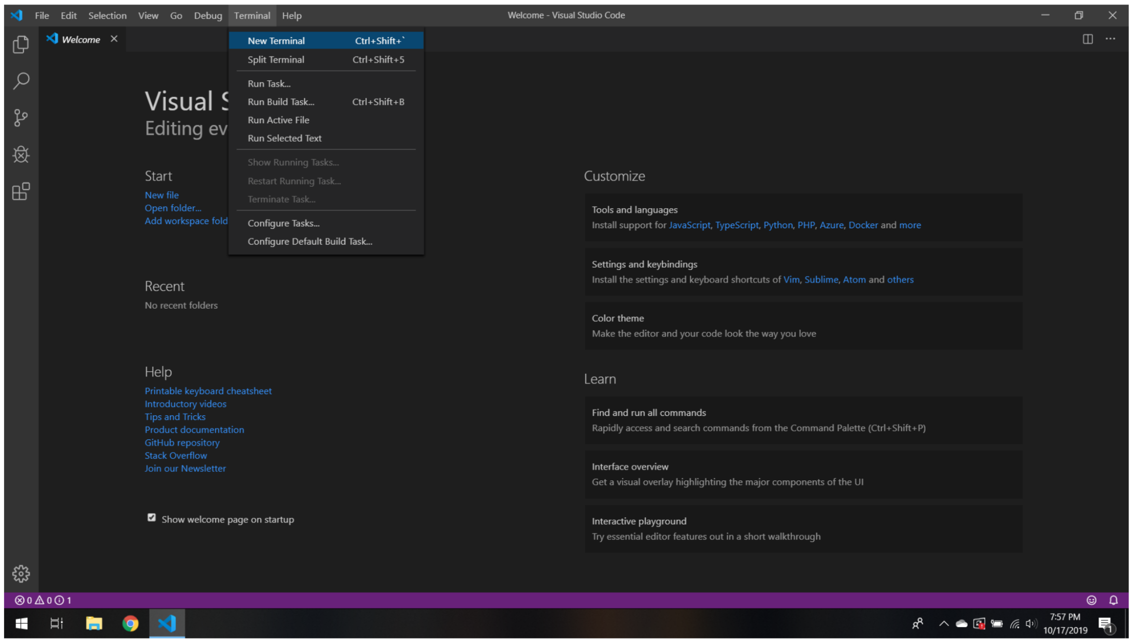
Name	Date Modified	Size	Kind
VSCode-darwin-stable.zip	Today at 11:32 AM	79.3 MB	ZIP archive

Step 3: Now you can see the "Visual Studio Code" application. Drag and drop this icon to the "Applications" folder of your computer.



Name	Date Modified	Size	Kind
Visual Studio Code	30-Aug-2019 at 3:50 PM	224.8 MB	Application
VSCode-darwin-stable.zip	Today at 11:32 AM	79.3 MB	ZIP archive

Step 4: Double click on the "Visual Studio Code" icon to launch the application. (You might need to right click and select "open" if you cannot launch the program). Next, select the new "Terminal option" to open the terminal window.



Install support for JavaScript, TypeScript, Python, PHP, Azure, Docker and more

Install the settings and keyboard shortcuts of Vim, Sublime, Atom and others

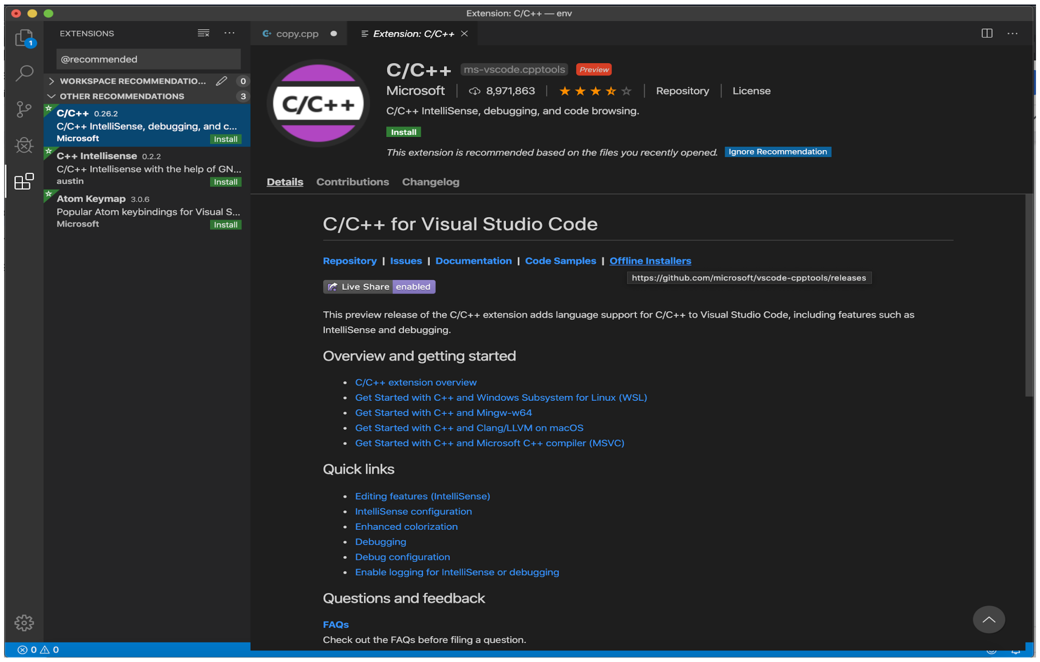
Make the editor and your code look the way you love

Rapidly access and search commands from the Command Palette (Ctrl+Shift+P)

Get a visual overlay highlighting the major components of the UI

Try essential editor features out in a short walkthrough

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.


Extension: C/C++ — env
EXTENSIONS
@recommended
WORKSPACE RECOMMENDATIO... 0
OTHER RECOMMENDATIONS 3
C/C++ 0.26.2
C/C++ IntelliSense, debugging, and c...
Microsoft Install
C++ Intellisense 0.2.2
C/C++ Intellisense with the help of GN...
austin Install
Atom Keymap 3.0.6
Popular Atom keybindings for Visual S...
Microsoft Install
copy.cpp
Extension: C/C++
C/C++
C/C++ ms-vscode.cpptools Preview
Microsoft | 8,971,863 | ★★★★☆ | Repository | License
Install
Ignore Recommendation
Details Contributions Changelog
Live Share enabled

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

https://github.com/microsoft/vscode-cpptools/releases

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

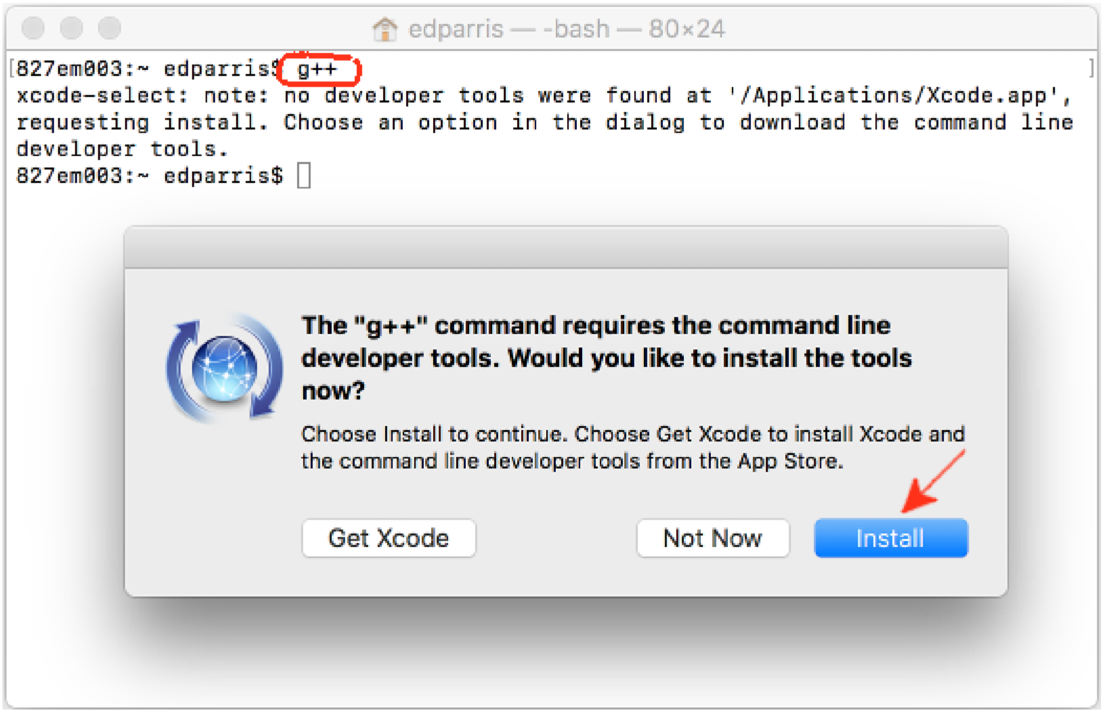
- Editing features (IntelliSense)
- IntelliSense configuration
- Enhanced colorization
- Debugging
- Debug configuration
- Enable logging for IntelliSense or debugging

Questions and feedback

FAQs
Check out the FAQs before filing a question.

Part 2 - Install g++

1. Open a Terminal window.
   (Press ⌘ Command+Space, type **Terminal** in the search field, and press the Return key.)
2. In the Terminal window type **g++** and press the Return key. We will see an alert box like this:



```
827em003:~ edparris$ g++
xcode-select: note: no developer tools were found at '/Applications/Xcode.app',
requesting install. Choose an option in the dialog to download the command line
developer tools.
827em003:~ edparris$ 
```

The "g++" command requires the command line developer tools. Would you like to install the tools now?

Choose Install to continue. Choose Get Xcode to install Xcode and the command line developer tools from the App Store.

3. Choose **Install** to get only the command line tools unless you want to learn Xcode. Xcode can be installed later from the App Store.

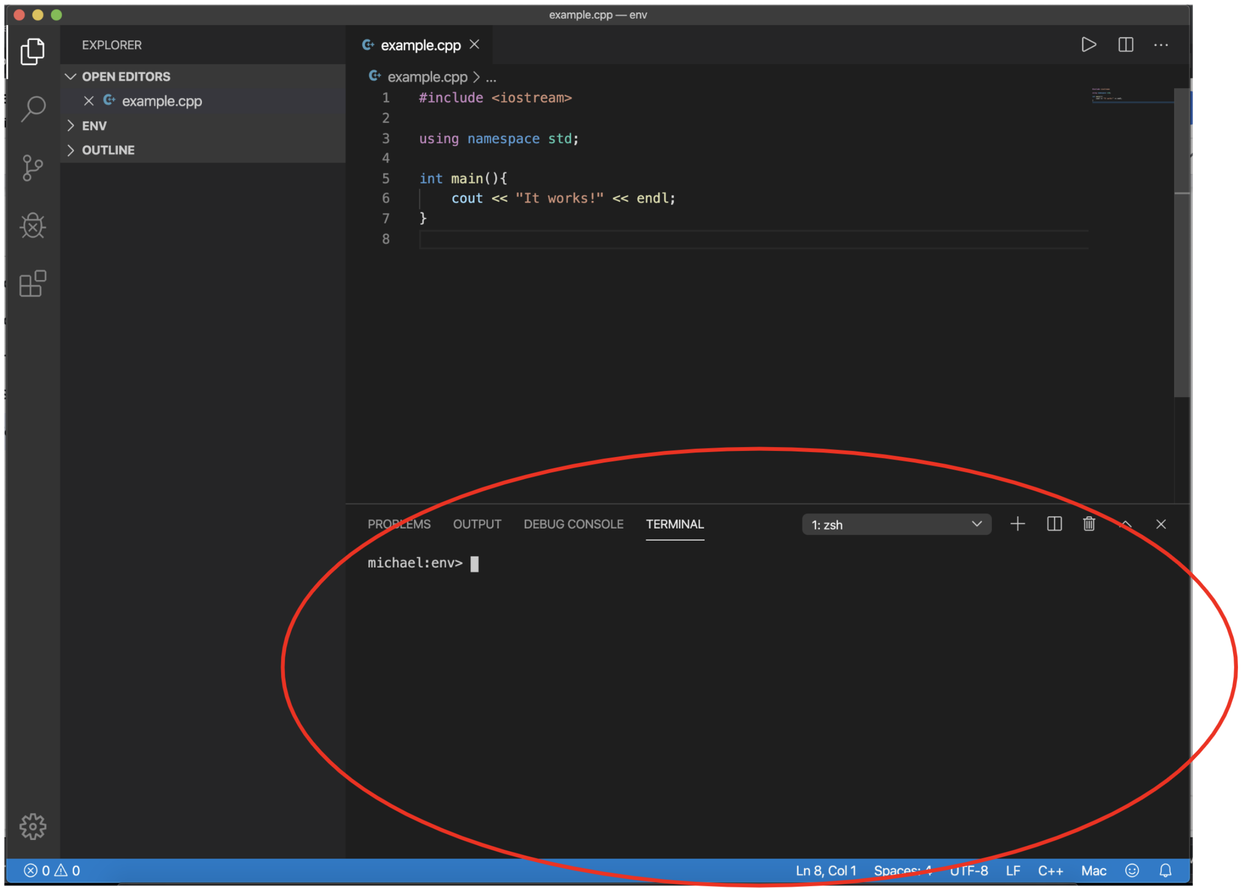
   After installation, type **g++** in the Terminal, press the Return key, and verify the terminal prints the message, "no input files".

   ```
   $ g++
   clang: error: no input files
   ```

Well done! Go to the section Compiling Code In Terminal to compile code on your computer.

Compiling Code In Terminal

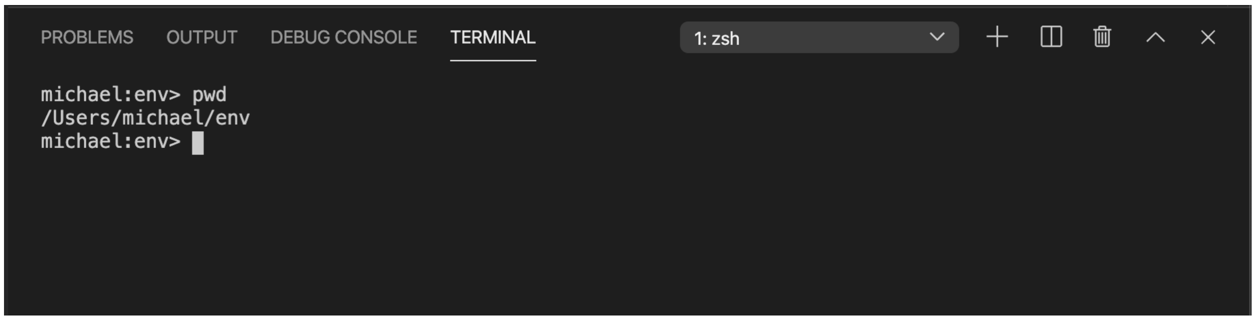
When you open VSCode at the bottom of the screen you should see a terminal.



```
#include <iostream>

using namespace std;

int main(){
    cout << "It works!" << endl;
}

```

```
michael:env>
```

In order to compile and run your code you first need to navigate to where your code is saved through the terminal. You do this using the **cd** (change directory) command (see the Homework 0 document on Canvas if you need an explanation on how to do this). If you have a folder open in VSCode by default the terminal will open to the location of that folder.
For instance on my computer I have a folder I made called **env** in VSCode which the terminal automatically opens to. You can check the current path your terminal is working at using the **pwd** command (print working directory) as shown below:

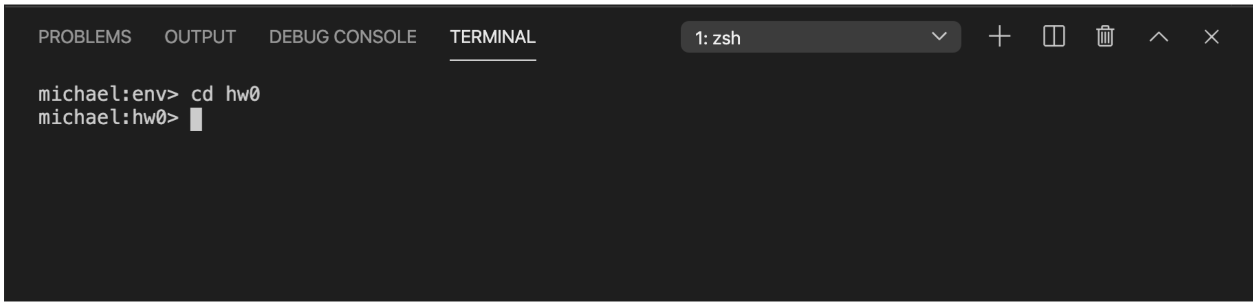


```
michael:env> pwd
/Users/michael/env
michael:env>
```

Note: Your terminal may look slightly different, as everyone's computer is different.
The first line in the screenshot above reads "michael:env> pwd", michael is the username of the person using the terminal, and env is the name of the current working directory. And pwd is the current command. Any command that users type will be after the > symbol. This symbol might be different depending on your computer, but that's okay.

Now the second line shows the output when the pwd command was run by pressing enter/return after the command which shows the current working directory. Don't worry about copying this directory path, yours can be different.

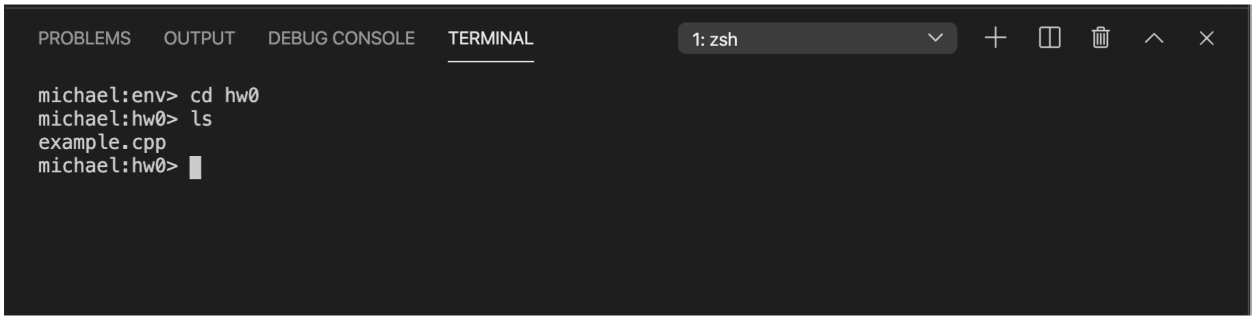
Now let's say the code I want to run is located in a subdirectory called **hw0** within the **env** directory, then I can navigate to hw0 using the **cd** command (change directory)**.** This command is of the form "cd <destination>". Where <destination> should be replaced by the path to the directory you want to switch to. Here if a folder/directory name is mentioned without a full path (as it is in the example below), it is considered a sub folder/directory of the current working directory.



```
michael:env> cd hw0
michael:hw0> 
```

Instead, if you wanted to provide a full path, the command would look something like:
cd /Users/michael/env/hw0
which corresponds to the same directory in the above example, albeit using a full path to the directory.

Now let's double check the files we are looking for are actual there by using the **ls** (list) command.

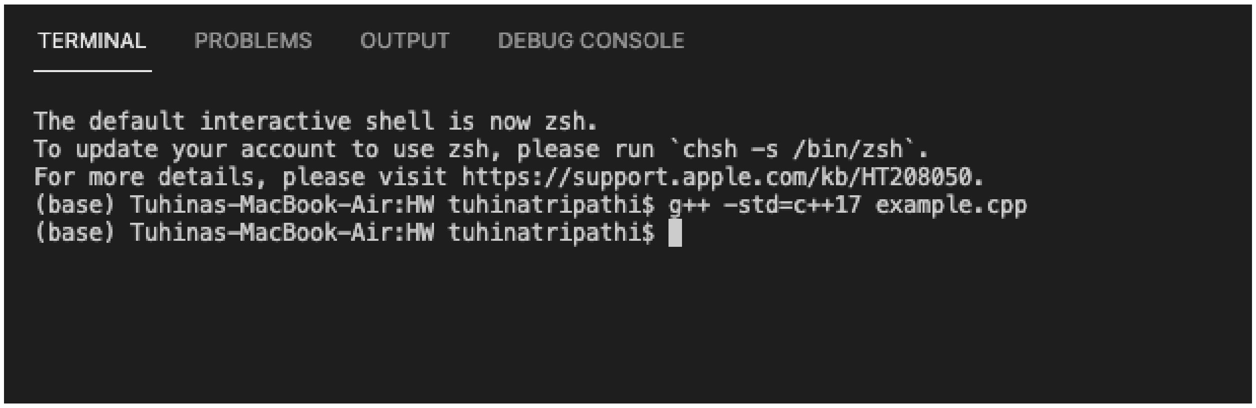


```
michael:env> cd hw0
michael:hw0> ls
example.cpp
michael:hw0> 
```

Now to compile our code we use the program g++ in the following command.

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

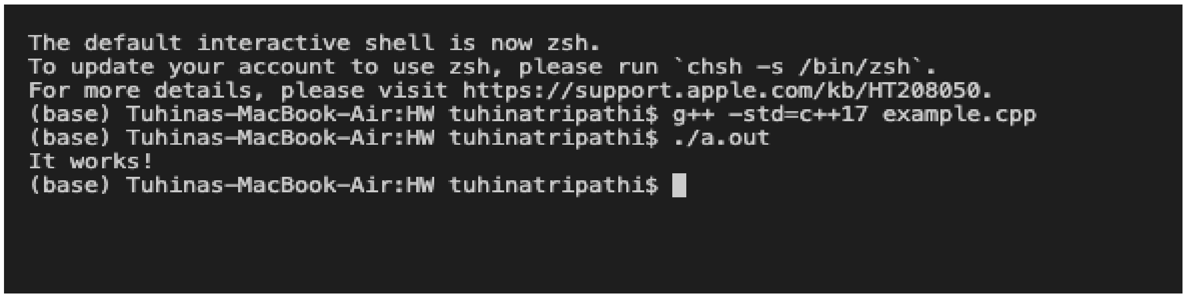
g**++** is the compiler program
-std=c++17 specifies the version of C++ we want to use
example.cpp is the file we want to compile



```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Tuhinas-MacBook-Air:HW tuhinatripathi$ g++ -std=c++17 example.cpp
(base) Tuhinas-MacBook-Air:HW tuhinatripathi$ 
```

This command creates a file named **a.out** which is the compiled version of the code in example.cpp. Now to execute our code we run a.out by typing the following and hitting enter.
./a.out

The same is shown in the screenshot below. (Note: the first line is from the previous command to compile)



```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Tuhinas-MacBook-Air:HW tuhinatripathi$ g++ -std=c++17 example.cpp
(base) Tuhinas-MacBook-Air:HW tuhinatripathi$ ./a.out
It works!
(base) Tuhinas-MacBook-Air:HW tuhinatripathi$ 
```

And that is how you compile and run a C++ program from a terminal.