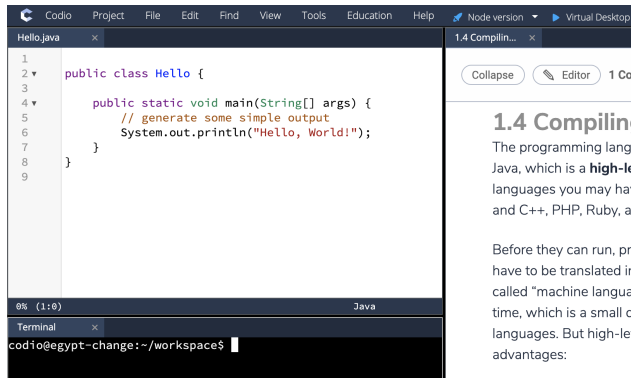# CS1111 Terminal And Compilation Tutorial for Fall 2020

Pablo Frank-Bolton

Terminal

The Terminal, or the Command Prompt is a text-based command-line interface that allows users to communicate and request actions from the computer. We will be using the Codio in-browser terminal but it is a good idea that you are also able to use your computer's terminal to compile and run programs.
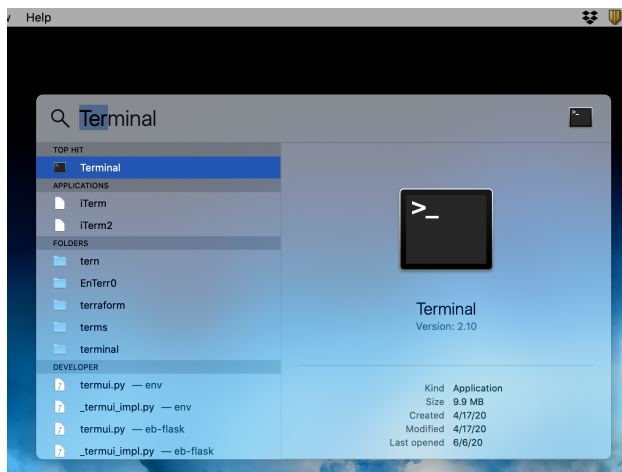
The Terminal In Codio

As you will soon learn, the Terminal can be opened in Codio or is predefined for some sections of the digital book. The following image shows its appearance for section 1.4 (the terminal is open in the bottom left pane):

If you are able to open the terminal in Codio, Mac, or Windows, it should work in a very similar manner.

Opening The Terminal In Mac

In Mac, you can open *Spotlight Search* by pressing `Command+Space` and searching for "Terminal".

You can also find it in `Applications→Utilities→Terminal`

Opening The Terminal In Windows

In Windows, the terminal is called the "Command Prompt". You can follow on of the steps in this tutorial to open it.

File Navigation

Mac, windows, and Unix (Linux) operating systems keep their files organized in different structures that usually resemmble hierarchical tree structure.
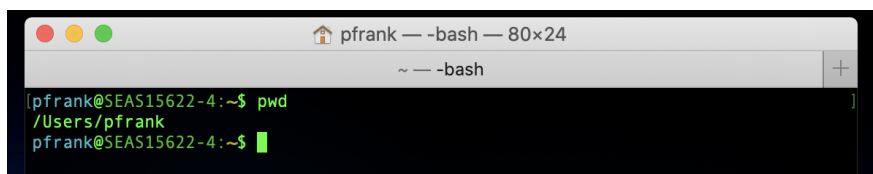
When you open a Terminal in any of these operating systems, it is like calling a company switch or operator. You can make a request or give a different extension so that your call is redirected.

When we open the Terminal window, the terminal opens into a *Current Directory* and it allows you to navigate to other directories.

In Codio, the navigation is identical to the one in the Mac version.
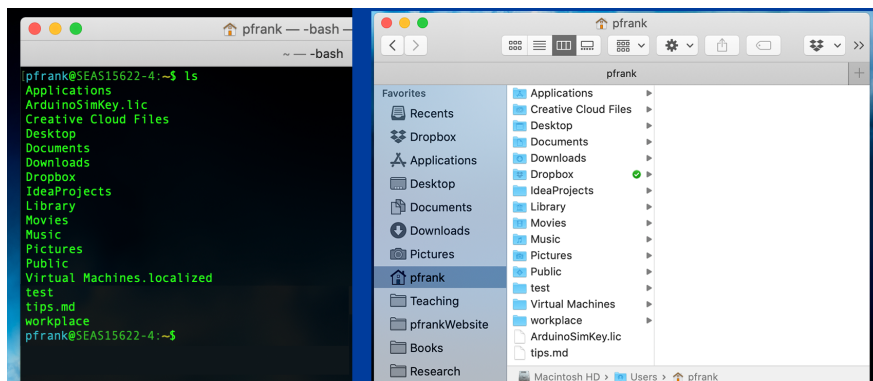
Basic Navigation Commands

Display the Current Directory : You can write `pwd+return` (print working directory) and the *Current Directory* will be printed in the Terminal.

This indicates that the *Current Directory* or location of the Terminal view is in `/Users/pfrank`. In Windows you'd see something like `\Users\pfrank`. In this example, my *username* is shown (pfrank). You will probablly see your own. Note the forward slashes in Mac or Linux, and the backward ones in Windows (they had to be special _(-_-)_/). In Codio, forward slashes are used.

List Directory Contens : You can write `ls+return` (list) and the contents of the *Current Directory* will be printed in the Terminal.

You'll see that the listed items correspond to the files and folders shown in the Finder (Mac) or Explorer (Windows) when you look in your home directory.

Changing Directory : You can write `cd Documents + return` (change directory to Documents) and the location of the *Current Directory* will now be "Documents". The general command is `cd <new directory> + return` where you can can replace the term `<new directory>` with any folder reachable from this location.

You can view this as going *down into* a lower folder. To go *up* to a *parent folder*, you can write the command: `cd .. + return`. Here, the two dots tell the Terminal that you want to go up.
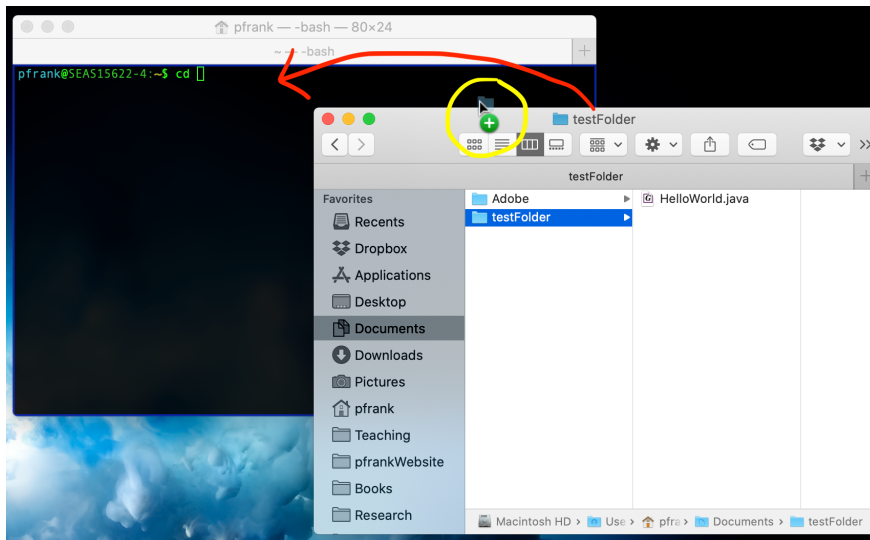
In general, the current directory is referred to with one dot. This means that, if you write `cd . + return`, you will remain in the same directory. All of these changes are done using *relative paths*, that is *paths or directions that start at the current location.*

Another way to navigate is using *absolute paths*, which are *paths or directions that start from the computer's root folder.* In Mac and (modern Windows) operating systems (OS), the root is indicated using a single slash. e.g. `cd /` would take you to the *root* in Mac.
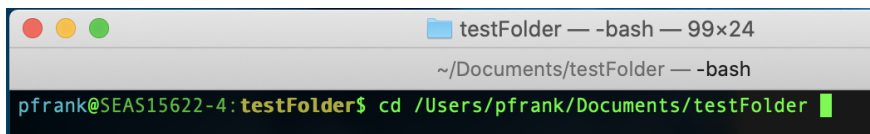
Opening the Terminal in the Correct Location

If you want to use the Terminal to compile a program, you need to open the Terminal into the location of that program. You can do it in two ways:

1. You can open a Terminal and navigate to the correct location using the instructions shown above, or
2. (Mac) You can write `cd` in the terminal (space after `cd`) and drag the folder location from the **top of the finder** to the terminal as show in the image below (dragging shown in yellow and destination in red):

That will result in the instruction `cd` followed by the full path of the file location (absolute path). If you hit `return` the Terminal will now be in that location. The next image shows the command before pressing "return":

Editing, Compiling, and Running a Program

We will show how to write a program in a text editor, compile it, and run it by using your native laptop capabilities.

Writing a Program in a normal text Editor

I use Sublime Text because it is pretty and it does nice syntax highlighting. You can use whatever you like.
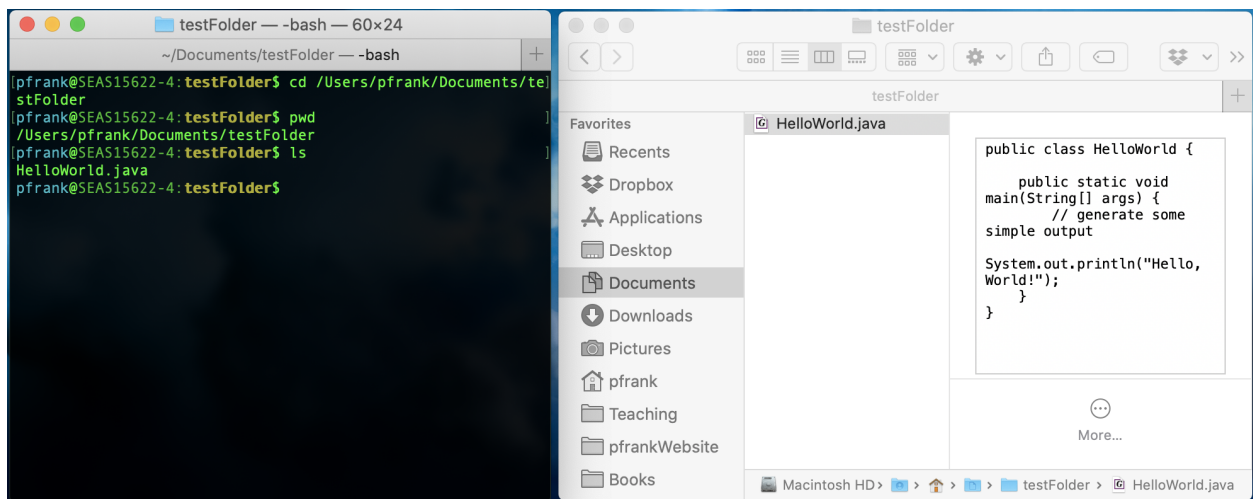
When you write code, make sure you save it with the correct file name and extension. In this case, the Program is encapsulated in a *Class* called *HelloWorld* so the file should be called *HelloWorld.java.* If your program has the wrong name or extension (watch out in Windows), it will not work.

```
1  public class HelloWorld {
2
3      public static void main(String[] args) {
4          // generate some simple output
5          System.out.println("Hello, World!");
6      }
7  }
```

Navigating to the Program Location

After you save it, You can open the *Terminal* and make sure you "Go To The Right Folder Location". I show the location with the file in the image below:

Once in the right folder (`/home/pfrank/Documents/testfolder/` in my example) you can now compile the program.

Compiling the Program

In Codio, Java comes pre-installed. In your machine, you might have to install it.

Installing Java in your machine. You can follow these instructions for Mac and these instructions for Windows.

This will save several java language libraries as well as a program that *compiles Java programs* called **javac** (or **java c**ompiler).
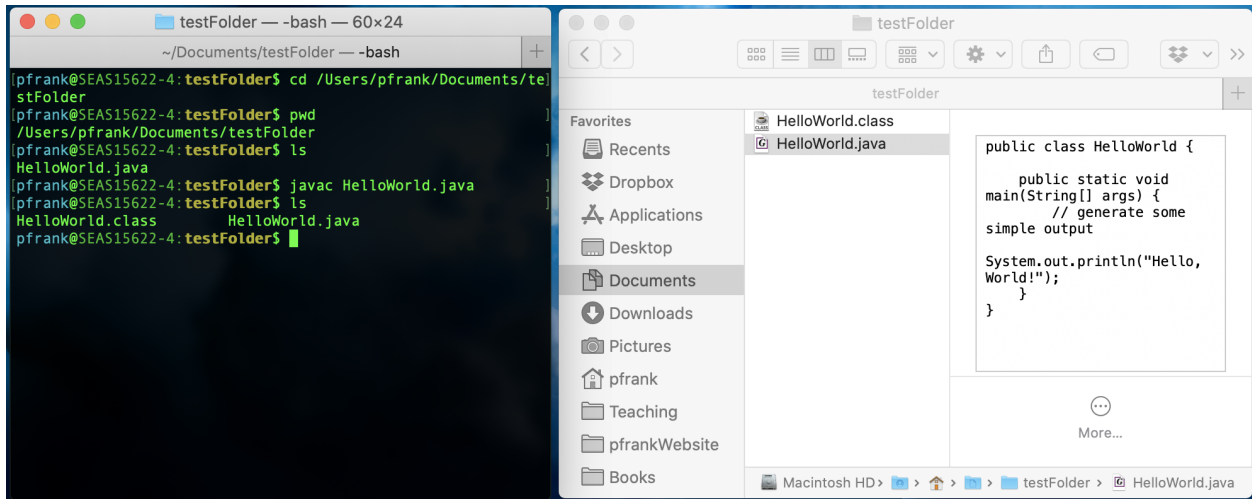
To compile your program you run the command:

```
javac <your program> + return
```

Where you would substitute the term `<your program>` with `HelloWorld.java` (in this case). The command you'll run is:

```
javac Helloworld.java + return
```

After this is completed, a new file has been created called `HelloWorld.class`. You can run `ls` in the current directory to display the contents. This whole process is shown in the next image:

This new file is the **executable** version of the program.
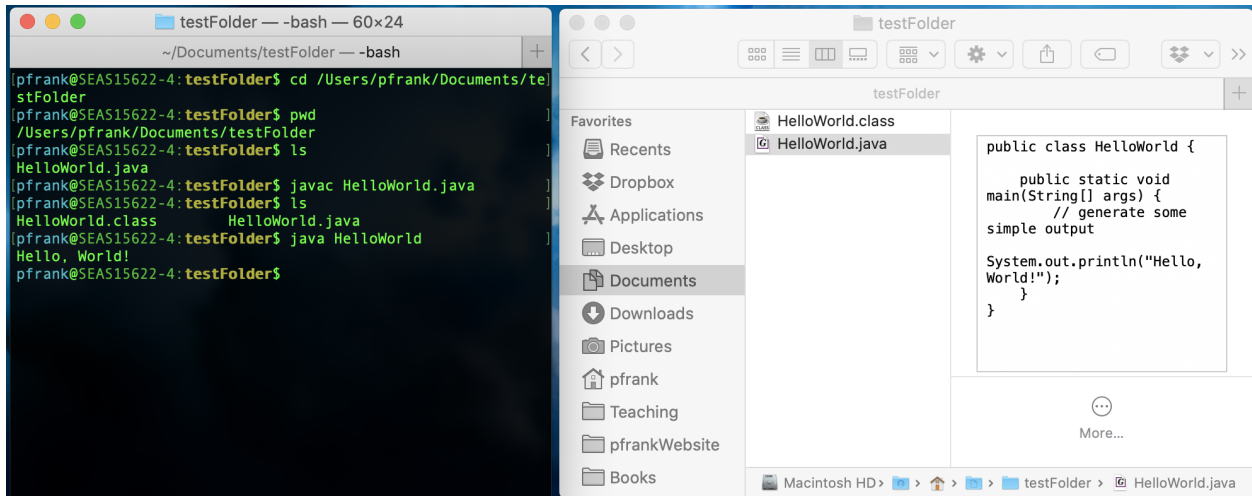
Running the Program

To run the program we will use the program **java** (no c! after "java"), you run the command:

```
java <your class file name> + return
```

Or

```
java Helloworld + return
```

This process is shown in the next image:

The program did what it was asked to do: It printed the sentence **"Hello, World!"** to the terminal.