VSCode and Git Basics

Sam McCauley
Data Structures, Spring 2026

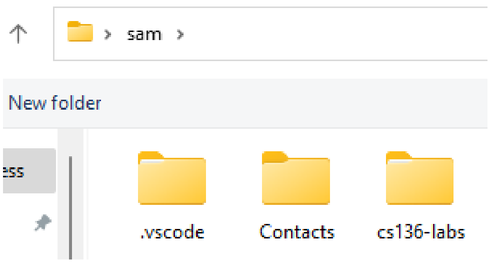
If you're reading this document, you should have git and VSCode set up on your computer, and you should have downloaded the starter code.

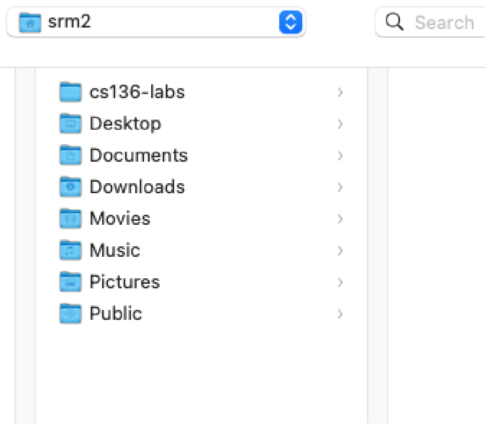
Now, let's open a file, change it, test to make sure it runs, and commit the changes using git.

Opening the Code

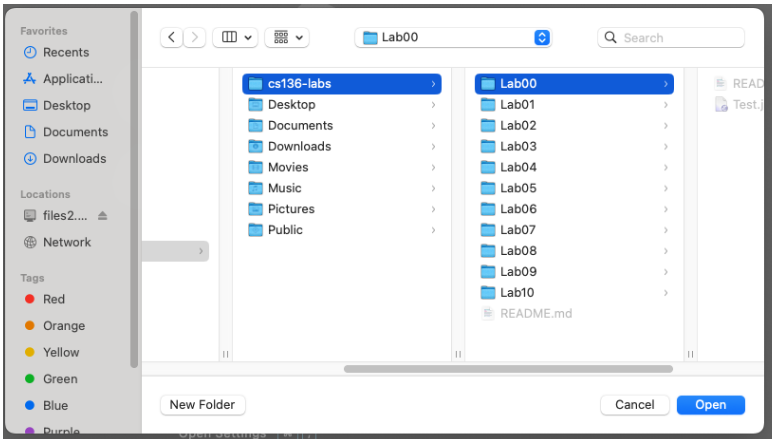
To open the code, you should use the "Open Folder" command (not the "Open File" command).

Go to File → Open Folder. You'll see the normal file browser; it probably looks like one of these:

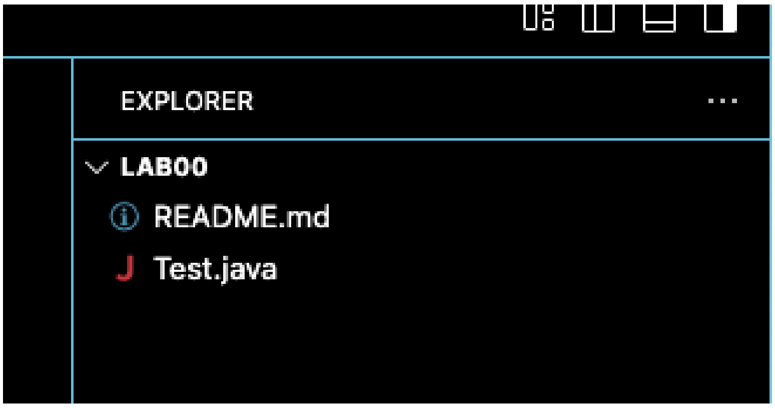




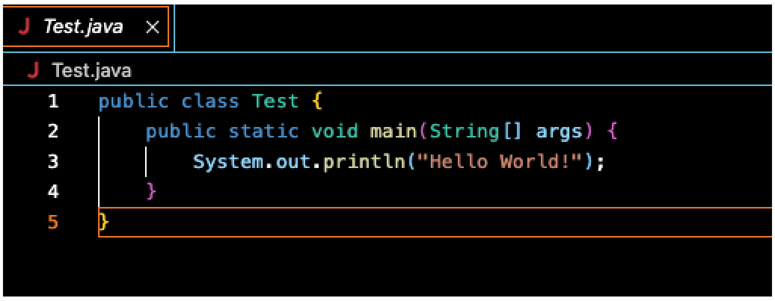
Click (double-click on Windows) on the folder "cs136-labs" then on the folder "Lab00". Then hit "Open."



Once you do this, VSCode will open a side bar with the files from the folder in it. It should look something like this:



Click on “Test.java”. The file will open in VSCode:



```java
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Running Java Code

Use Control-` to open the terminal (That is to say: hold the control key, and then hit the key immediately above the Tab key. This is not an apostrophe!)

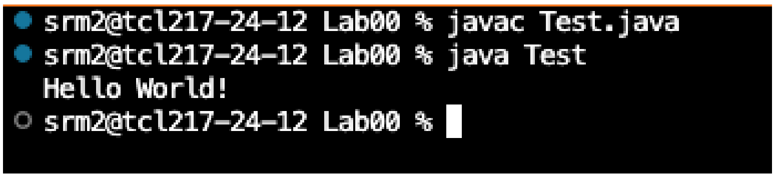
First, type the following and hit enter (the > at the beginning is the prompt; don’t type that)

```
> javac Test.java
```

This will compile your Java program. Then, run it with the following (note that you do *not* write .java this time):

```
> java Test
```

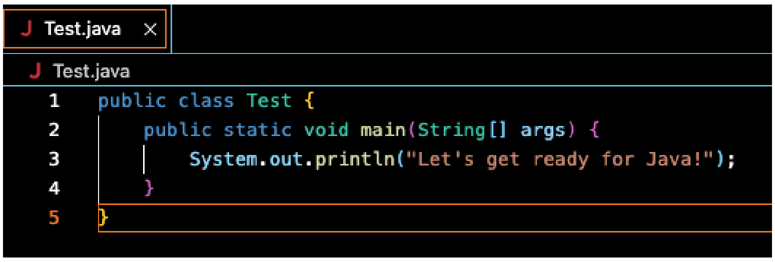
The result should look something like the following:



```
srm2@tcl217-24-12 Lab00 % javac Test.java
srm2@tcl217-24-12 Lab00 % java Test
Hello World!
srm2@tcl217-24-12 Lab00 %
```

Changing the Code

You can see in the code that it says “Hello World!” Let’s change that to a different message. Delete `Hello World` (leave the quotes), and instead type in a fun message. I typed: `Let’s get ready for Java!`



```java
public class Test {
    public static void main(String[] args) {
        System.out.println("Let's get ready for Java!");
    }
}
```

Save the file using Control-S or command-S. Then, compile and run the file again to make sure it works; the program should now output your message.

Pushing your Changes to Git

Let’s make sure your changes are reflected in the git repo.

This requires three steps. (These steps probably seem silly or arduous now, but they are actually very helpful when you have larger repos.)

Add the file: First, *make sure you have saved your changes*. There should not be a blue dot next to the tab that says `Test.java`.

Type the following in the terminal (and hit enter):

```
> git add Test.java
```

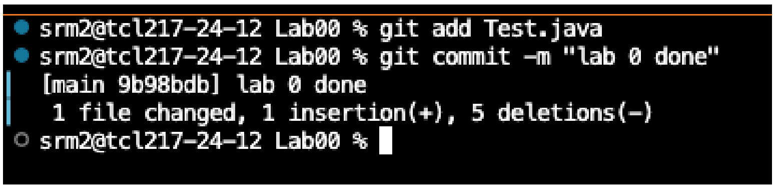
This tells git that the next commit you make (see below) will include the changes in `Test.java`.

Commit the file: Type the following in the terminal (and hit enter):

```
> git commit -m "lab 0 done"
```

This tells git to keep track of the changes you've made on your computer. Think of this as "saving" the changes, but not uploading them to the cloud.

Your terminal should look something like this:



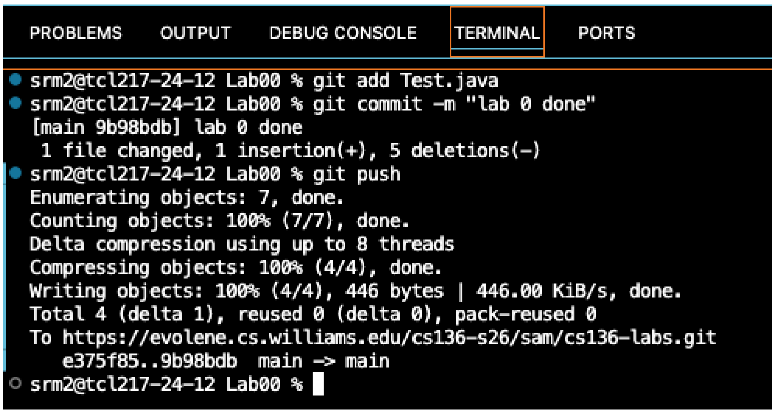
```
srm2@tcl217-24-12 Lab00 % git add Test.java
srm2@tcl217-24-12 Lab00 % git commit -m "lab 0 done"
[main 9b98bdb] lab 0 done
 1 file changed, 1 insertion(+), 5 deletions(-)
srm2@tcl217-24-12 Lab00 %
```

Push the commit: Type the following in the terminal (and hit enter):

```
> git push
```

This pushes the commit to the cloud. Now the instructors can see the changes you've made.

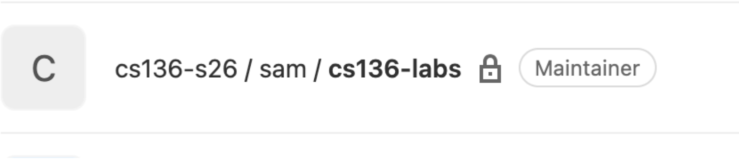
Altogether, your terminal should look something like this:



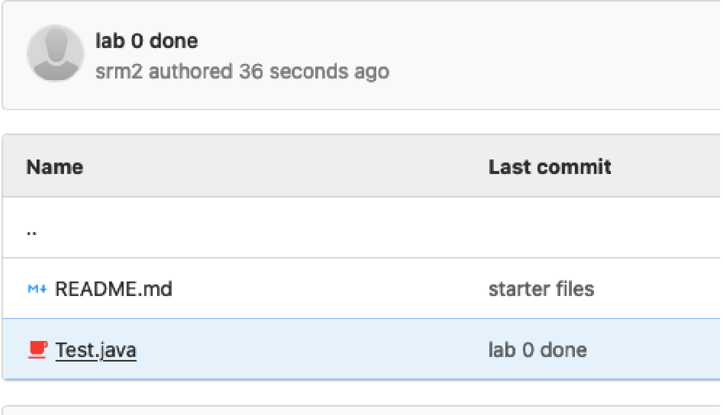
```
srm2@tcl217-24-12 Lab00 % git add Test.java
srm2@tcl217-24-12 Lab00 % git commit -m "lab 0 done"
[main 9b98bdb] lab 0 done
 1 file changed, 1 insertion(+), 5 deletions(-)
srm2@tcl217-24-12 Lab00 % git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 446 bytes | 446.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
To https://evolene.cs.williams.edu/cs136-s26/sam/cs136-labs.git
   e375f85..9b98bdb  main -> main
srm2@tcl217-24-12 Lab00 %
```

Make sure it worked! Let's double-check that everything worked.

Go to evolene.cs.williams.edu and sign in with your CS credentials. Just as you did to download the lab, click on your repo:



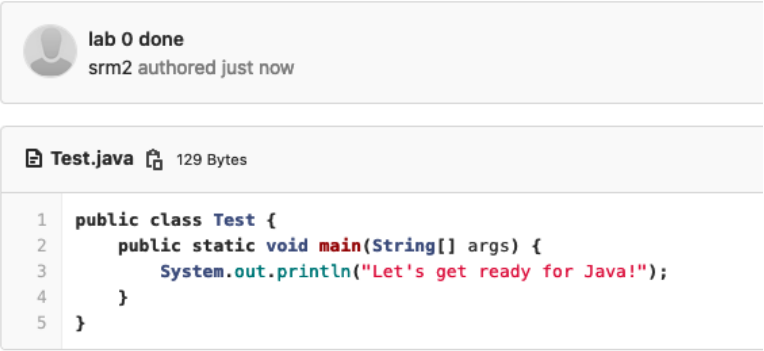
Then (this part is new), click on `Test.java`:



lab 0 done
srm2 authored 36 seconds ago

Name	Last commit
..	
README.md	starter files
Test.java	lab 0 done

The result should look something like the following—your changes should be reflected in the repo.



lab 0 done
srm2 authored just now

Test.java 129 Bytes

```java
public class Test {
    public static void main(String[] args) {
        System.out.println("Let's get ready for Java!");
    }
}
```

Congrats!