

# Academic Integrity Assignment

## [Overview](#)

## [Terminology](#)

[Command line](#)

[M1 vs. Intel Mac](#)

[Text Editor](#)

### [1. Install the Java Development Toolkit \(JDK\)](#)

#### [1.1 Check if Java is already installed](#)

##### [1.1.1 Open your computer's native command line application](#)

##### [1.1.2 Check the java version](#)

#### [1.2 Download the latest JDK](#)

#### [1.3 Add Java to the system path \(Windows only\)](#)

#### [1.4 Check if Java is now installed](#)

### [2. Create, save, compile, and run a program](#)

#### [2.1. Download a text editor](#)

#### [2.2. Write the source code](#)

#### [2.3. Save your source code to a file](#)

#### [2.4. Navigate to your saved file on the command line](#)

#### [2.5. Compile your program](#)

#### [2.5. Run your program](#)

### [3. Web browser option](#)

#### [3.1. Create a Repl.it account](#)

#### [3.2. Create a new Repl](#)

#### [3.3. Create the Integrity.java file](#)

#### [3.4. Add the source code](#)

#### [3.5. Compile your code](#)

#### [3.6. Run your code](#)

#### [3.7. Download your file](#)

### [4. Profit! Submit](#)

[Grading rubric](#)

### [5. Common Errors](#)

['javac' is not recognized as an internal or external command, operable program, or batch file](#)

[Error: a JNI Error has occurred, please check your installation and try again](#)

[error: illegal character: 'u201c'](#)

# Overview

This assignment will serve two functions. First, it will confirm to us that you understand and agree to abide by the academic integrity guidelines for this course. Second, this will confirm to us that you have set up all the necessary infrastructure to write, compile, and run/test a standalone java program.

This writeup will mostly focus on how to run a java program from the command line. Speaking from my own experience, I can tell you that developing command line fluency early on will make your life exponentially easier when you take different computer science classes with different programming languages. However, if you can't get the command line to work, the writeup will also describe how to create and run java programs from your web browser. **You only need to complete the command line option or the web browser option - not both.**

**The Friday lecture of the first week, as well as student hours during the first two weeks, will be solely dedicated towards helping you with this assignment. If you have any uncertainty, ask for help!**

We will complete this assignment in the following steps:

1. Install the Java Development Kit (JVM)
2. Write the source code
3. Save your source code to a file
4. Navigate to the source code file in your command line program
5. Compile the program
6. Run the program
7. ~~Profit!~~ Submit

If you run into errors, there is a section at the end called [common errors](#). If your error isn't listed, come to student hours.

This assignment will be graded and after you are more than welcome to re-work it as many times as you like. However, this assignment will not factor into your final grade calculation; it is a low-stakes chance for you to practice writing and running simple java programs before you tackle bigger java projects.

## Terminology

Before getting into the instructions for installing java, I want to introduce you to some terminology that we'll be referring to later.

## Command line

You are probably used to using your computer by using your mouse to click, point, and drag and drop to tell the computer what you want to do. However, there is another way to use the computer. Your computer will have a built-in application that allows you to run the computer by typing in commands; the computer will read the command, and then execute it. This used to actually be the only way to use the computer, before they invented the user interfaces that you are used to today.

For this course, we will be using the command line to get the computer to execute all of the java code that we write.

## M\* vs. Intel Mac

All Mac computers have a processor. Older Mac computers use a processor that was designed by Intel. However, more recent Mac computers use an M1 processor that Apple designed themselves (or M2, M3, etc.).

Why is this important? The answer is that you will need to use a different version of Java depending on which kind of Mac you are using.

To see which kind of Mac computer you are using, do the following:

1. Click on the Apple icon in the top-left corner.
2. Click on “About this Mac”
3. Look at the “Processor” section; it will tell you if your computer uses an Intel Mac or M\* (i.e. M1, M2, etc.)

Here is a [video demonstrating how to check this](#).

## Text Editor

A text editor is simply a piece of software that allows you to write, re-write, edit, and manipulate text. Two examples of Text Editor that you might be familiar with are Microsoft Word and Google Docs. These two applications give you a lot of power and flexibility for composing text.

Unfortunately, however, these two applications don't let you create Java source code files. Thus, later in this tutorial I will guide you on what text editor to use.

# 1. Install the Java Development Toolkit (JDK)

## 1.1 Check if Java is already installed

Before installing it, check if you already have Java installed. You probably don't - and that's OK! But let's just do it anyway. This will also give you a head start on becoming familiar with using the command line.

### 1.1.1 Open your computer's native command line application

For Mac, this is called *Terminal*; for Windows, this is called *Command Prompt*.

If you are using a Mac, here are the steps to open the Terminal:

1. Click on the magnifying glass icon in the top right corner.
2. Search for "Terminal"
3. Click on "Terminal" from the search bar. Its icon should be a black box

If you are using a Windows computer, here are the steps to open the Command Prompt:

1. Click the search bar in the bottom left corner
2. Search for "Command Prompt"
3. Click on "Command Prompt" from the search bar. Its icon should be a black box.

If you need a visual demonstration, here are videos for how to open these applications on different types of computers:

- Mac: [open the Terminal application](#)
- Windows 10: [open the Command Prompt application](#)
- Windows 7: [open the Command Prompt application](#)

### 1.1.2 Check the java version

Once you open up the command line application, type `javac -version`. Hit enter. If it lists the version of `javac` installed on your computer, then great! You can skip this section! But it's more likely that it will give you some error message stating that `javac` is not a recognized command. That's ok, and that's what we're here to fix.

If you need a visual representation, here are two videos for how to check if Java is installed for Mac and Windows users, respectively.

- [Check if Java is installed on your Mac computer](#)
- [Check if Java is installed on your Windows computer](#)

## 1.2 Download the latest JDK

Download it from [here](#).

- **M1 Mac:** If you're using a Mac with an M1 chip, download the **Arm 64 DMG Installer**. Then open Finder, go into Downloads, and double click the DMG file that you downloaded.
- **Intel Mac:** If you're using a Mac with an intel chip, download the **x64 DMG Installer**. Then open Finder, go into Downloads, and double click the DMG file that you downloaded.
- **Windows:** If you're using Windows, download **Windows x64 Installer**. After you download it, find the downloaded file in your Downloads folder (using either Finder for Mac, or My Computer for Windows). Double click the file to run the java installer.

## 1.3 Add Java to the system path (Windows only)

If you are on a Mac, you should be able to skip to the next step. If you are on windows, there is one more step. You need to tell your computer how and where to find the java compiler and interpreter that you just downloaded.

The process is basically as follows:

1. Find the directory where java was installed. It will likely be in `C:\Program Files\Java\jdkXXX\bin`. The **part in red** will be dependent on what version of Java that you install. The point is that this should be the directory that contains two files: `javac.exe` and `java.exe`. Copy this directory (using ctrl-C).
2. Click on My Computer
3. Click on Properties
4. Click on Advanced System Settings
5. Click on Environment Variables
6. Scroll through the list of System Variables. Click on `Path`
7. Click "edit" to edit it.
8. If you are on Windows 10, click "new", and then paste the directory that you copied in step 1.
9. If you are on an older version of Windows, it's a little different, and you have to be careful.
  - a. When you click "edit", Windows will bring up a small text area to edit the value of the `Path` variable. Initially, the entire string will be selected/highlighted. **Whatever you do, do not press delete or anything like that while you have the entire string selected.**
  - b. Press the right arrow key so that your cursor is moved to the very end of the `Path` variable.
  - c. First, add a semicolon to the end of the string.
  - d. Second, paste the directory that you copied in step 1.

10. Once you have finished step 8 or 9, press “ok” until you’ve closed all the windows.

If you are on Windows 10, [here is a video](#) that walks you through the process of adding java to your Windows environment. If you are using an older version of Windows, you can follow [this video](#) instead. The process is nearly identical, but it looks just a little different, which is why I included two videos.

## 1.4 Check if Java is now installed

Follow the same process that you followed in section 1.1.2. When you type `javac -version` it should list the javac version that’s been installed on the computer. If you are using command prompt on windows, you will have to re-start the command prompt before testing it again.

**Make sure this part is working before you move on. If it’s not, ask for help.**

# 2. Create, save, compile, and run a program

## 2.1. Download a text editor

As mentioned above, Microsoft Word and Google Docs are powerful, but they are not designed to write computer programming code.

If you are on a Windows machine, download the latest version of [Notepad++](#). If you are on a Mac, download the latest version of [BBEdit](#).

**Note:** There are other text editors that will get the job done for this class, and you are free to download/use any of them. You can do a google search for other text editors, and see if you find something you like better. That said, the two above are both free, and both extremely well-suited for writing java code, which is why I encourage them. In particular, if you are on Windows, I really cannot overemphasize just how much I love Notepad++, and how jealous I am that it doesn’t run on Mac.

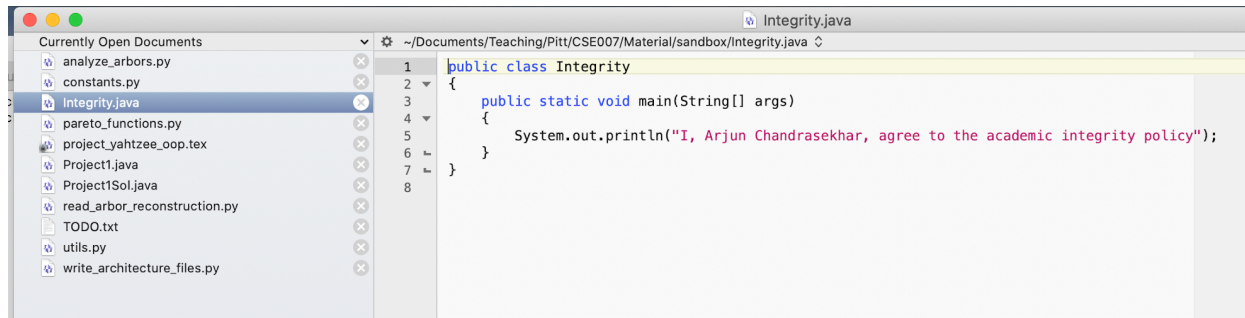
## 2.2. Write the source code

Open up the text editor that you downloaded in the last section. Open up a new file, and put the following code into your file (either by typing it manually, or using copy/paste).

```
public class Integrity
{
    public static void main(String[] args)
    {
```

```
        System.out.println("I, [INSERT NAME], agree to the academic  
integrity policy");  
    }  
}
```

The part in **blue** should not be copied verbatim; you should replace it with your actual name. Here is what my source code file looks like in BBEdit.



**Be careful about copying and pasting the quotation marks.** A lot of times the quotation marks in google docs copy weirdly into your text editor. If you decide to copy/paste the code (rather than typing it all from scratch), you will probably want to delete and manually re-type the quotation marks.

## 2.3. Save your source code to a file

Save the file. In particular, do the following:

1. Go the top-right corner and click “File”
2. Click “Save as”
3. Select the folder on your computer where you want to save your code (I suggest making a special folder just for this course).
4. If you are using Notepad++, click on the “Save as type” menu, and select “java source code file”
5. Type the name `Integrity.java`, and click save.

**You need to name your file `Integrity.java`. If you name it something different, it won’t compile.** This includes getting the capitalization correct.

In principle, you can save it to any location that you want on the computer. Personally, I recommend that you create a special folder to save all of your `CS54-184` code. I would also nest this folder inside your main `Documents` folder.

Make a note of the location where you saved the file.

## 2.4. Navigate to your saved file on the command line

Open up the native command line application on your computer like you did at the start. Navigate to the location where you saved your source code file. To do this, you will need to use the following commands:

- `ls`: list all of the files in the current directory (Terminal only)
- `dir`: list all of the files in the current directory (Command Prompt only)
- `cd directory`: move from the current directory into `directory`
- `cd ..`: move from the current directory to the parent directory

For more information, I have written up this [tutorial](#) on how to use the command line. You can also watch a video on a [terminal tutorial](#) or a [command prompt](#) tutorial

Once you think you have navigated to the correct directory, perform an `ls/dir` and you should see your source code file listed. That signals that you are ready to move on to the next section.

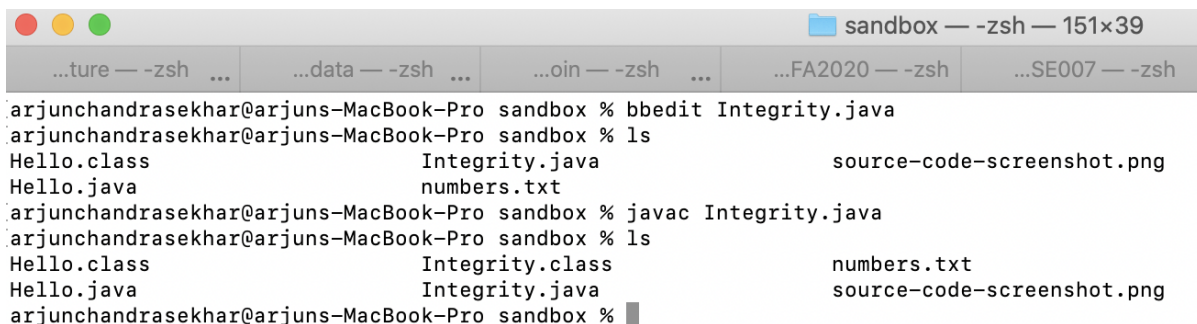
## 2.5. Compile your program

First, compile the program. Perform the following command: `javac Integrity.java`

List all of the files in the current directory (using either `ls` or `dir`). You should see a new file called `Integrity.class`

**Take a screenshot of your command line application after you compile your program.**

This screenshot will be part of your submission, and it will let us know that you actually succeed in compiling your program from the command line. This is what my screenshot looked like:



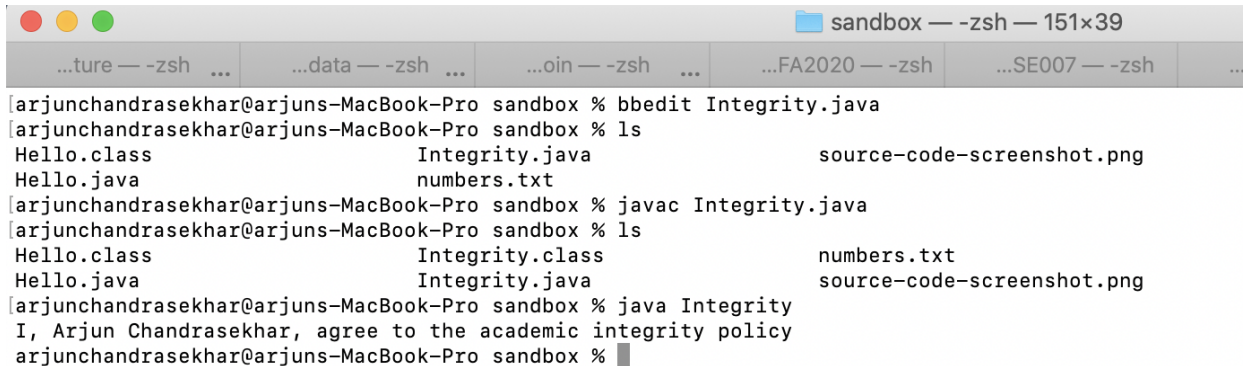
```
sandbox — -zsh — 151x39
arjunchandrasekhar@arjuns-MacBook-Pro sandbox % bbedit Integrity.java
arjunchandrasekhar@arjuns-MacBook-Pro sandbox % ls
Hello.class          Integrity.java        source-code-screenshot.png
Hello.java           numbers.txt
arjunchandrasekhar@arjuns-MacBook-Pro sandbox % javac Integrity.java
arjunchandrasekhar@arjuns-MacBook-Pro sandbox % ls
Hello.class          Integrity.class       numbers.txt
Hello.java           Integrity.java        source-code-screenshot.png
arjunchandrasekhar@arjuns-MacBook-Pro sandbox %
```

## 2.5. Run your program

Next, run the executable file that you just created. Perform the following command: `java Integrity`

It should print out the integrity agreement message.

**Take a screenshot of what happened when you ran your program.** Here is what my screenshot looked like:



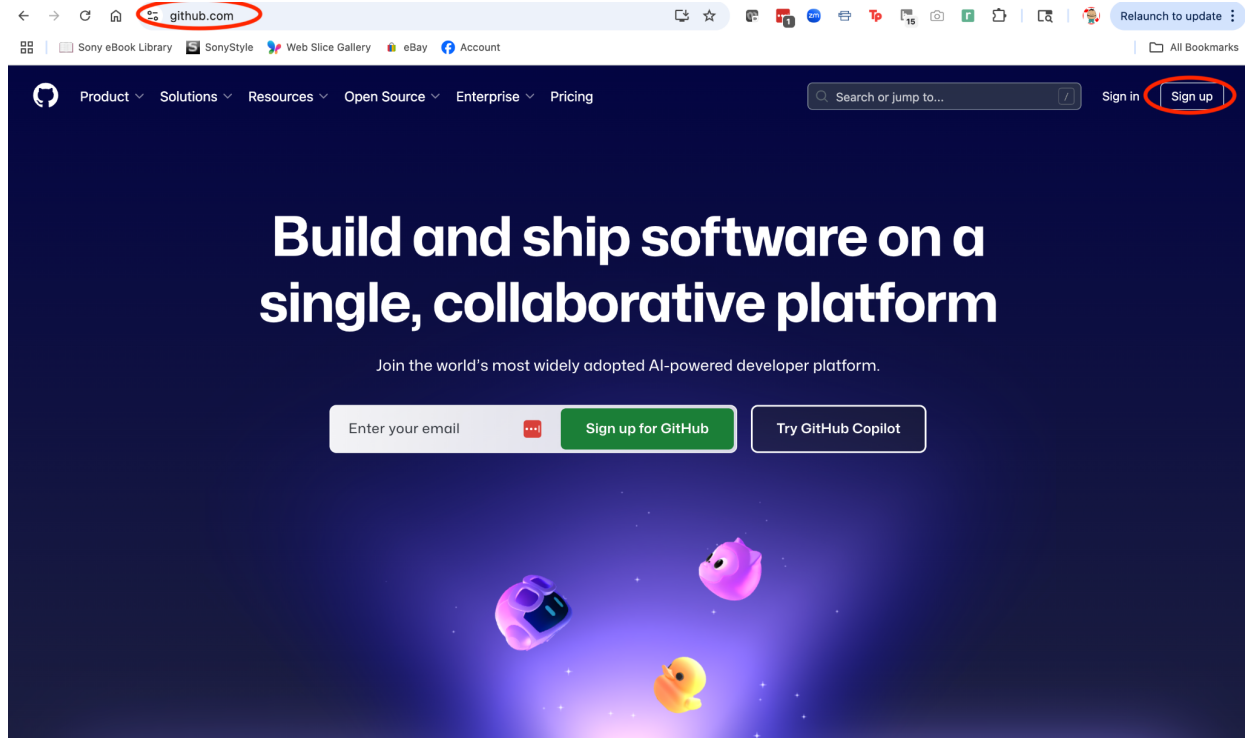
```
[arjunchandrasekhar@arjuns-MacBook-Pro sandbox % bbedit Integrity.java
[arjunchandrasekhar@arjuns-MacBook-Pro sandbox % ls
Hello.class          Integrity.java        source-code-screenshot.png
Hello.java           numbers.txt
[arjunchandrasekhar@arjuns-MacBook-Pro sandbox % javac Integrity.java
[arjunchandrasekhar@arjuns-MacBook-Pro sandbox % ls
Hello.class          Integrity.class       numbers.txt
Hello.java           Integrity.java        source-code-screenshot.png
[arjunchandrasekhar@arjuns-MacBook-Pro sandbox % java Integrity
I, Arjun Chandrasekhar, agree to the academic integrity policy
[arjunchandrasekhar@arjuns-MacBook-Pro sandbox % ]
```

### 3. Web browser option

**If you successfully compiled and ran your code from the command line on the computer, you can skip to step 4.** However, if you are not able to complete the assignment on the command line, the backup option is to write and run your program from the web browser. To do this, you will use **Github's codespace feature**. Here are the steps:

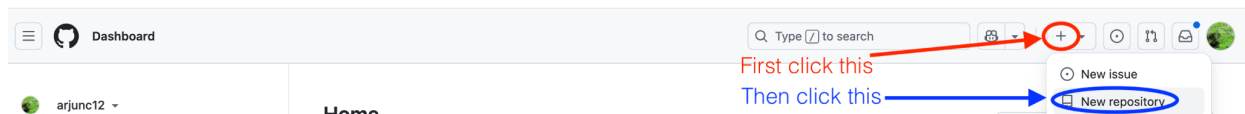
#### 3.1. Create a Github account

Go to <https://github.com/>. In the top right corner, click "Sign up" to create your account. You'll use these same credentials to log in going forward.



## 3.2. Create a Repository

In the top right corner click the plus (+) button, and then create a new repository.



Fill out the form.

- Use whatever name you like. I suggest something like `Southwestern-CS1-FA2025`, but it doesn't really matter.
- **Please make the repository private.** Making it public will allow other students to see and copy your work.
- **Initialize it with a README.** It just makes it easier, the user interface looks really weird if you don't do this.

## Create a new repository [Try the new experience](#)

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

### Repository template

No template ▾

Start your repository with a template repository's contents.

Owner \*

 arjunc12 ▾

Repository name \*

Southwestern-CS1-FA2025

✓ Southwestern-CS1-FA2025 is available.

Use whatever name  
you want

Great repository names are short and memorable. Need inspiration? How about [special-parakeet](#) ?

Description (optional)

Please make it private

☐

Public

Anyone on the Internet can see this repository. You choose who can commit.

☒

Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Check this, it's easier

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

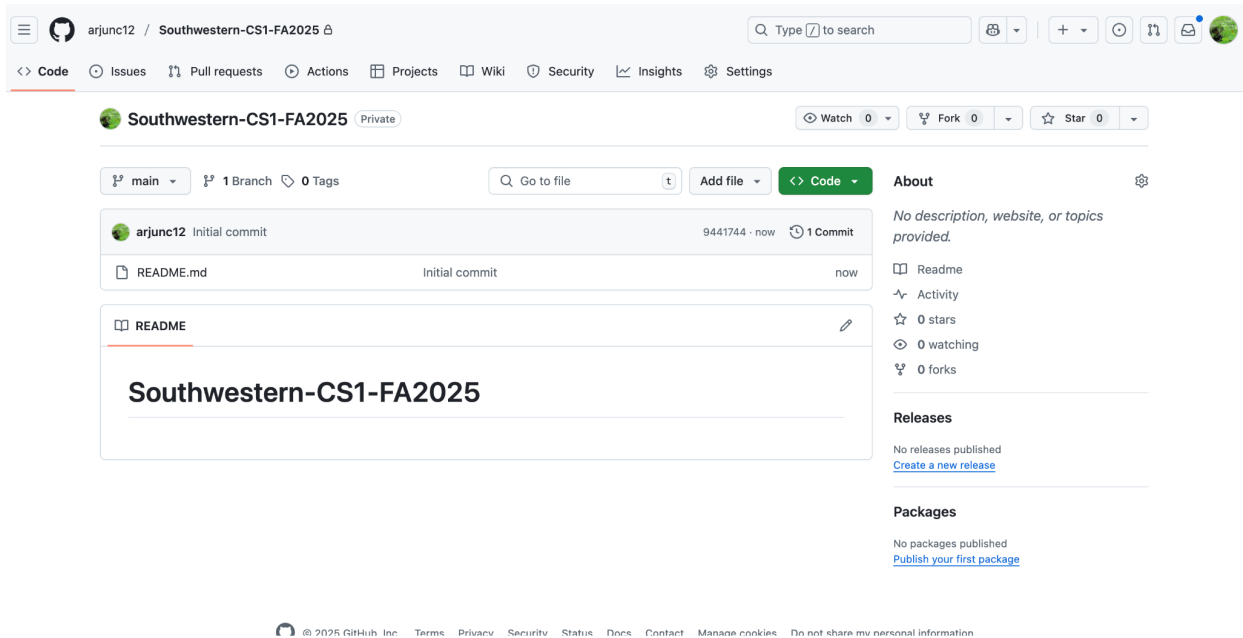
This will set `main` as the default branch. Change the default name in your [settings](#).



You are creating a private repository in your personal account.

Create repository

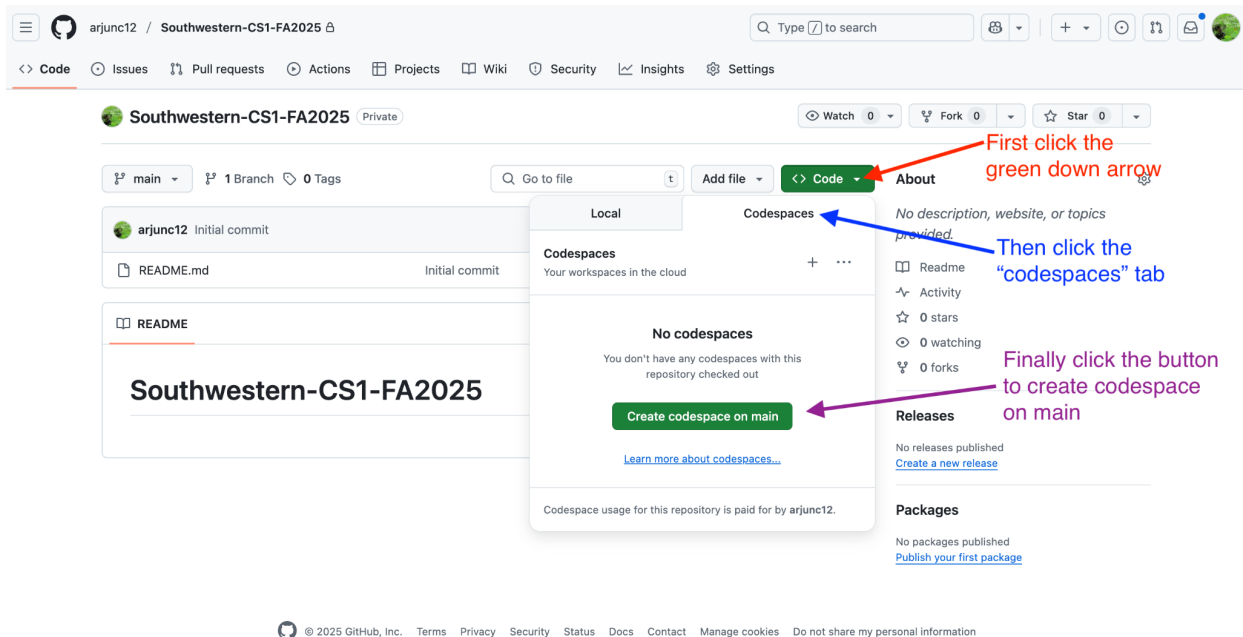
Once you click the button to create the repository, it should look something like this:



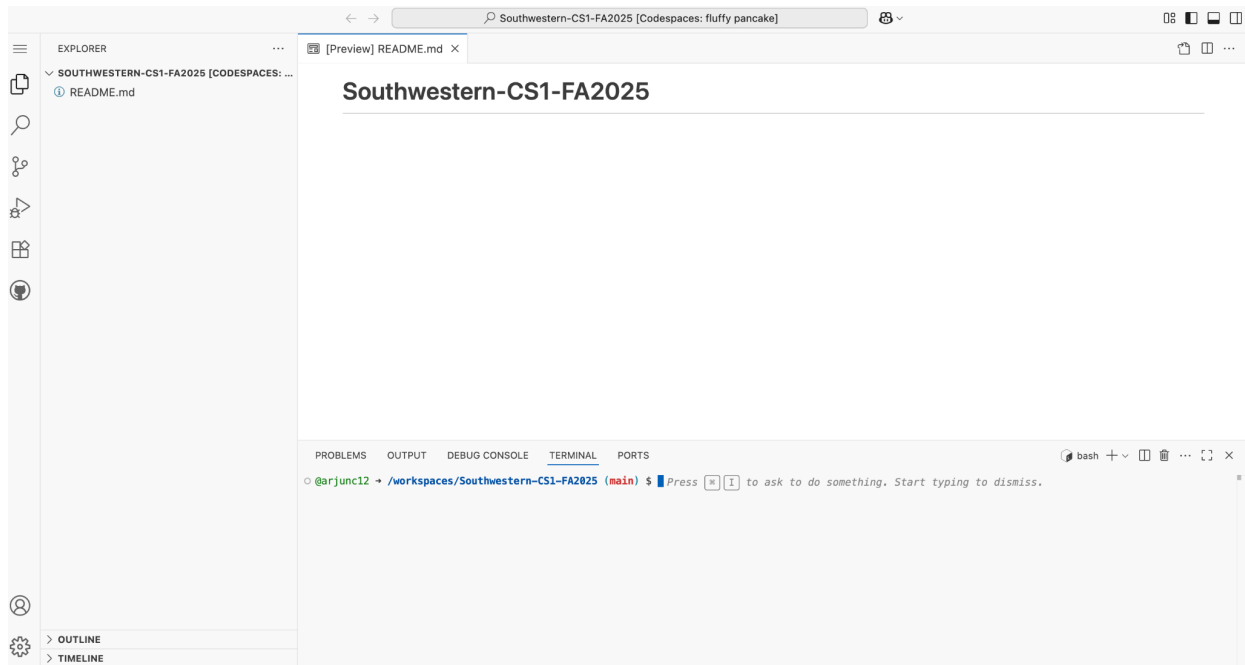
### 3.3. Create the `Integrity.java` file

#### 3.3.1 Open up a codespace

Click the green “Code” button - specifically the green arrow. Click on the codespaces tab. Then click the button to create a new codespace on main.

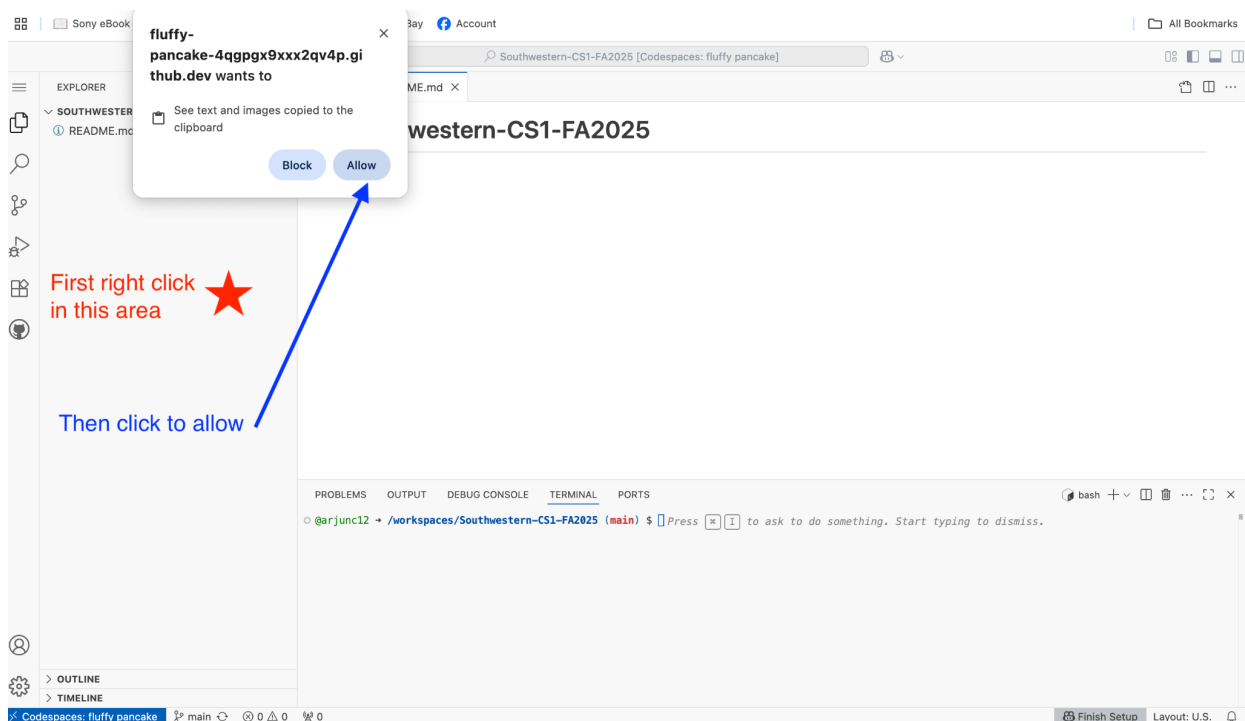


Your codespace should look like this

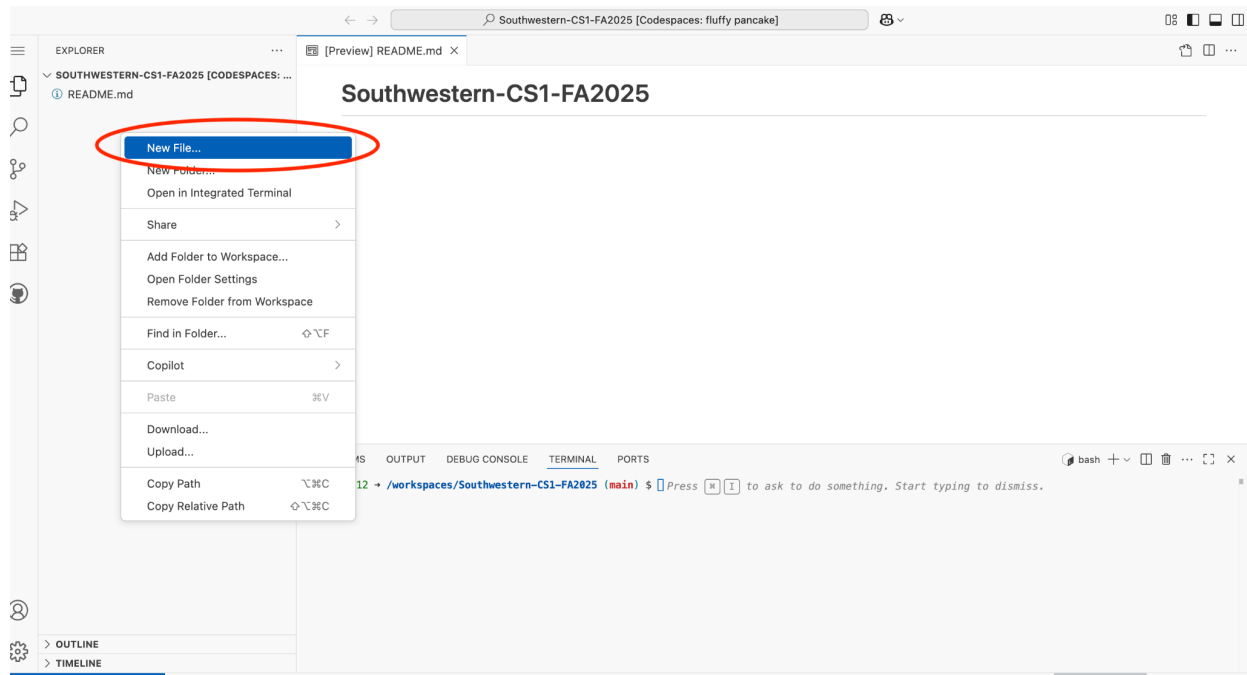


### 3.4. Add the source code

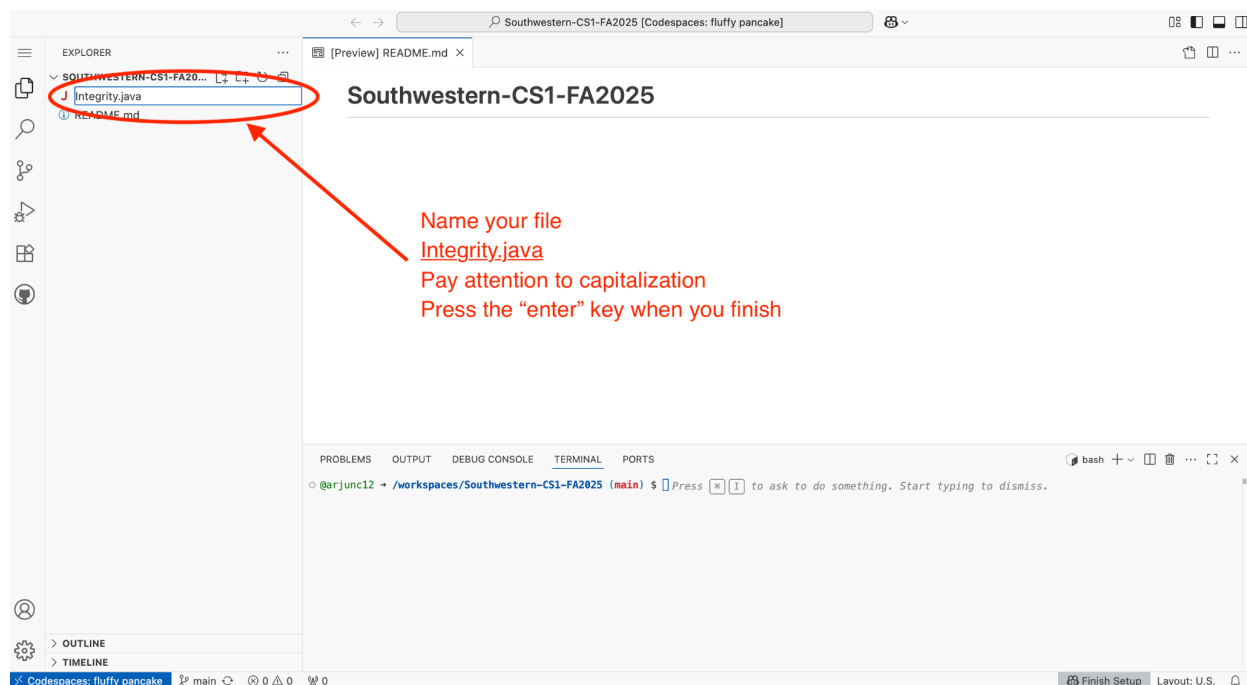
First you'll need to right click in the file explorer on the left. It should ask you if you want to allow the website to copy-paste. Allow it. Here's what that will look like:



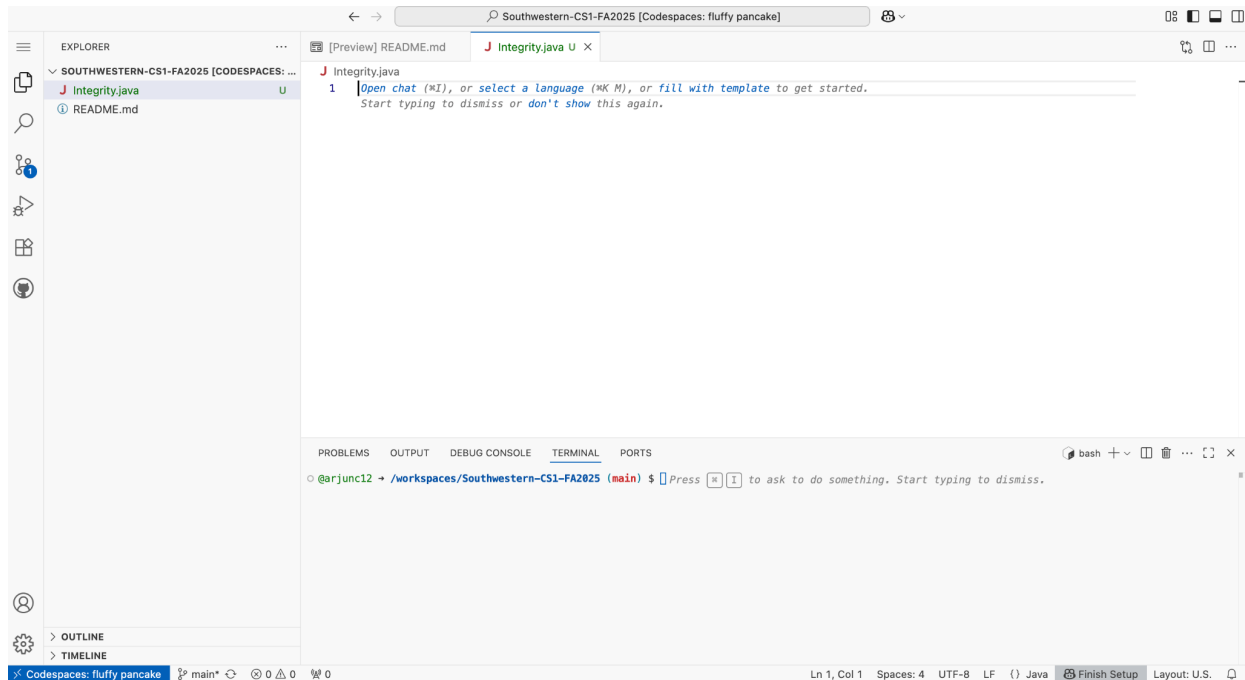
After allowing the right-click, click to add a new file



Name your new file **Integrity.java**. Pay close attention to capitalization.

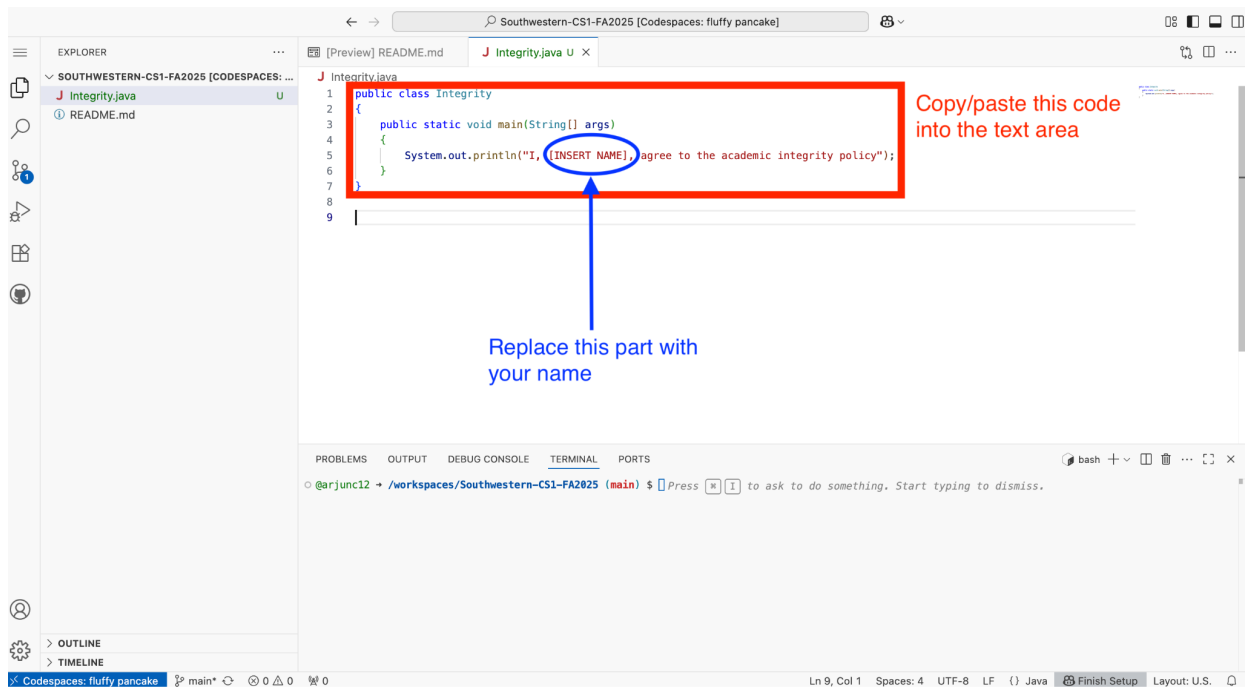


After you create the file your screen will look like this:



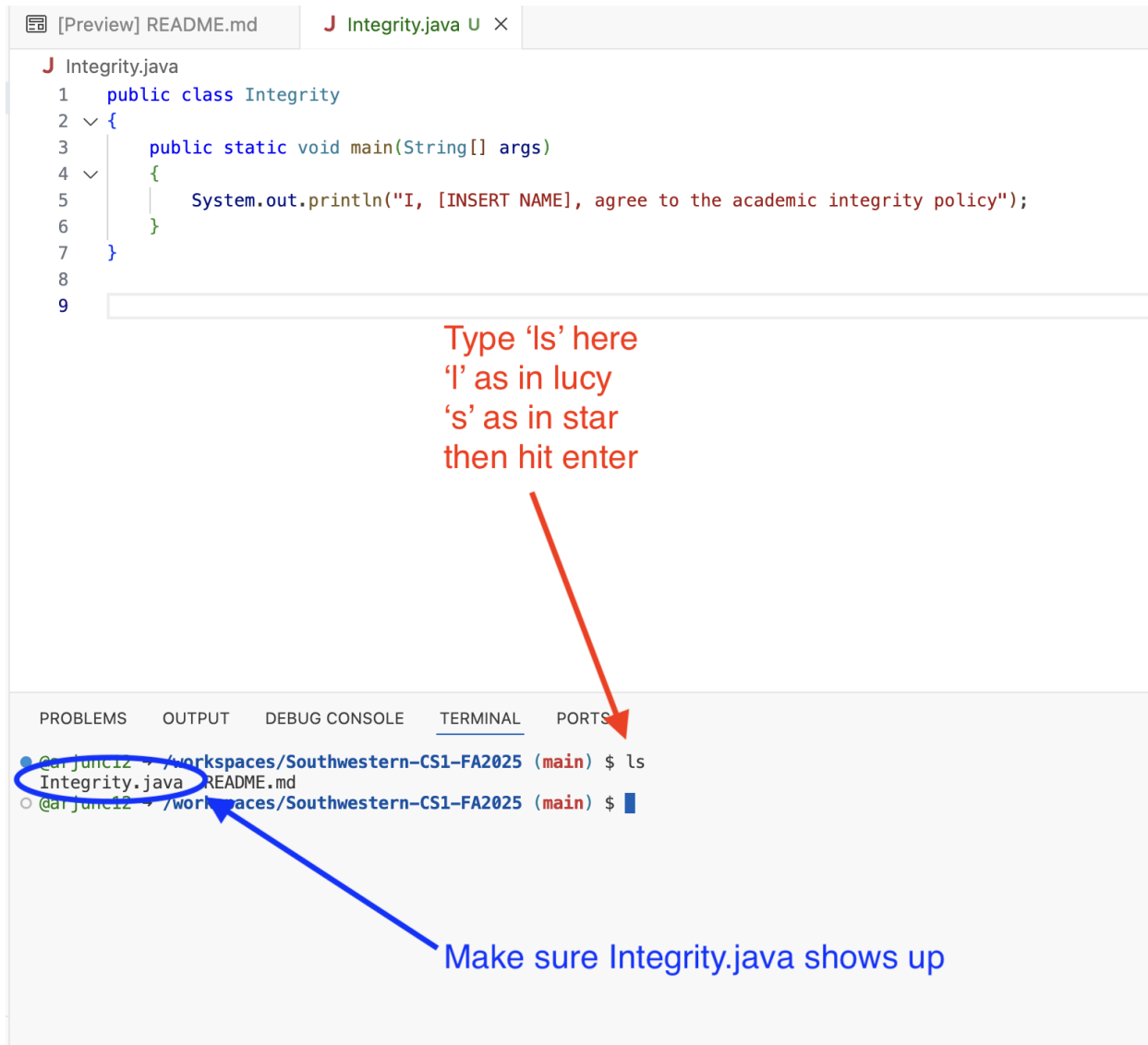
Copy and paste the integrity source code into your `Integrity.java` file. In particular, put the following code into your file:

```
public class Integrity
{
    public static void main(String[] args)
    {
        System.out.println("I, [INSERT NAME], agree to the academic
integrity policy");
    }
}
```

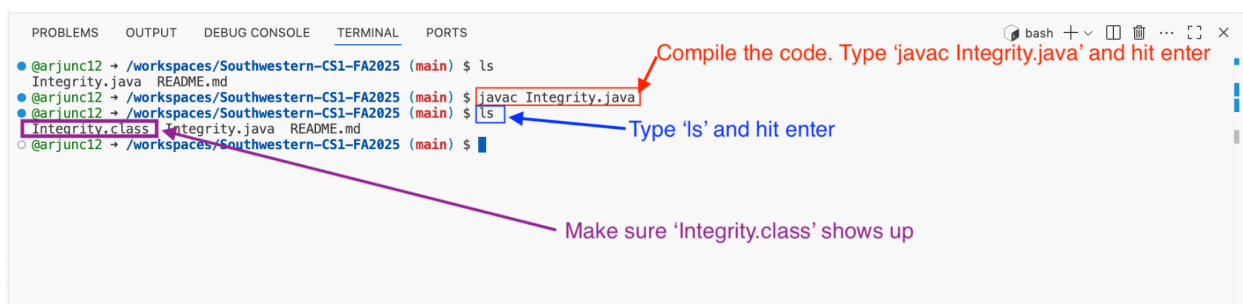


### 3.5. Compile your code

Now let's get familiar with the console. First, let's list out all of the files in the project. Click in the terminal box on the bottom of the window. You should see a cursor appear. Then, type `ls` and hit enter. You should see the console list out several file names.



Next, compile your code. Go back to the console. Type `javac Integrity.java` and hit enter. After this finishes, type `ls` and hit enter to re-list all of the files. You should see a new file called `Integrity.class` appear.



Take a screenshot of what happened after you compile your program.

### 3.6. Run your code

Go back to the console. Type `java Integrity` and hit enter. This will run your source code. It will then output the text from the program.



The screenshot shows a VS Code terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
@arjunc12 → /workspaces/Southwestern-CS1-FA2025 (main) $ ls
Integrity.java  README.md
@arjunc12 → /workspaces/Southwestern-CS1-FA2025 (main) $ javac Integrity.java
@arjunc12 → /workspaces/Southwestern-CS1-FA2025 (main) $ ls
Integrity.class  Integrity.java  README.md
@arjunc12 → /workspaces/Southwestern-CS1-FA2025 (main) $ java Integrity
I, [INSERT NAME], agree to the academic integrity policy
@arjunc12 → /workspaces/Southwestern-CS1-FA2025 (main) $
```

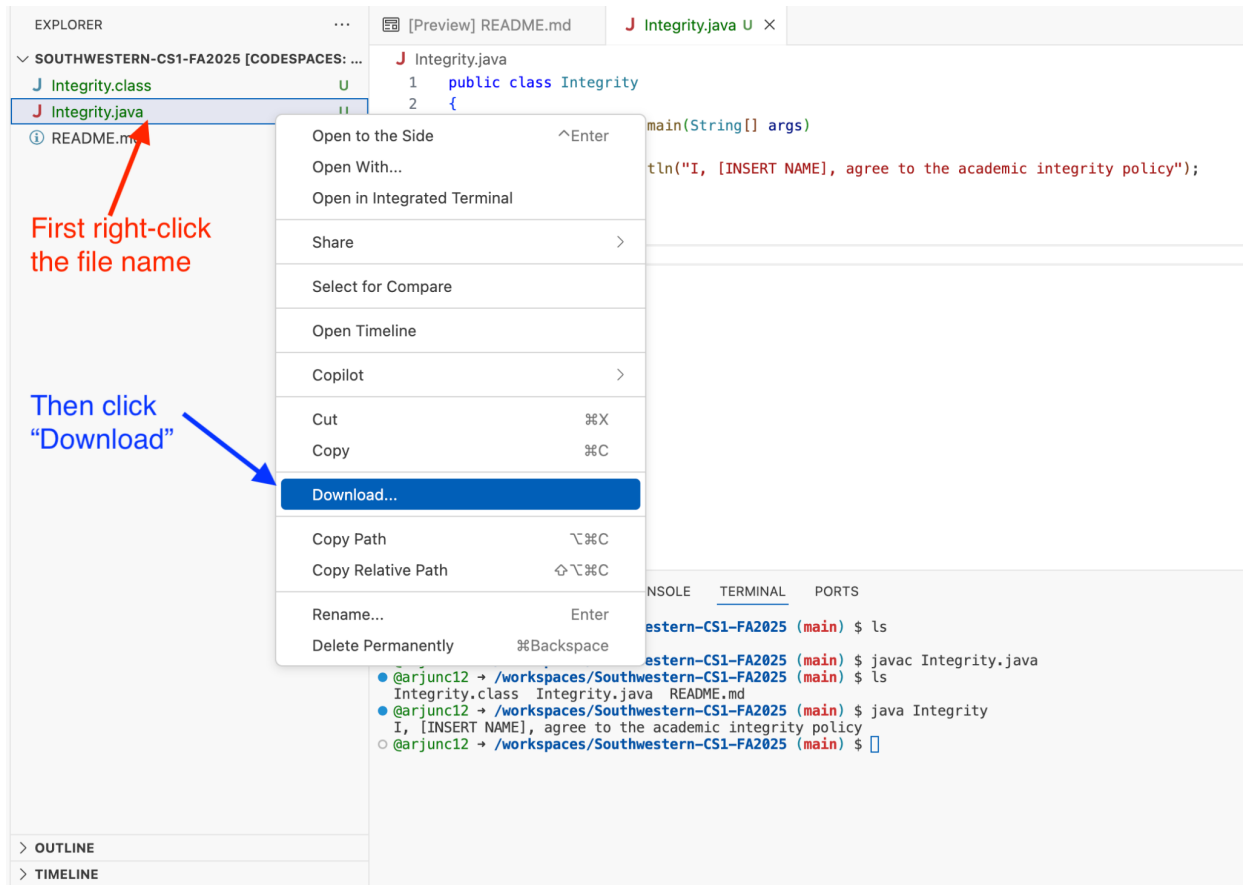
Annotations in the image:

- A red box highlights the command `java Integrity` in the terminal.
- A red arrow points from the text "Type 'java Integrity' and hit enter" to the red box.
- A blue box highlights the output line `I, [INSERT NAME], agree to the academic integrity policy`.
- A blue arrow points from the text "You should get output that looks like this" to the blue box.

Take a screenshot of what happened after you ran your program.

### 3.7. Download your file

Finally, download the source code file that you created. Click on the three vertical dots right next to `Integrity.java` in the file explorer. Select the Download option. This will save the contents of `Integrity.java` to your personal computer.



## 4. Profit! Submit

Please submit the following files:

- Your source code file `Integrity.java`
- A screenshot of what happened when you compiled the program
- A screenshot of what happened when you ran the program

Please do NOT submit any of the following:

- An image or screenshot of your code - we can't run or test this. You should submit a screenshot of the commands you ran in the terminal or the command prompt, but for your code you should submit the actual file with the java code.
- `Integrity.class` - this is just machine code, and it doesn't let us evaluate the code that you actually wrote.
- A source code file with any name other than `Integrity.java`. Your two screenshots can have any name, but your source code file must be named `Integrity.java`. You will not receive credit for the source code file if it has the wrong name.

Please submit all of your files on Moodle.

# Grading rubric

`Integrity.java` file compiles and runs correctly: 30 points

Screenshot of compiling the program: 10 points

Screenshot of running the program: 10 points

## 5. Common Errors

**Please read this section!!!!** Reading this section could save you a lot of time. Here are some common errors and how to fix them:

**‘javac’ is not recognized as an internal or external command, operable program, or batch file**

This indicates that you did not add javac to the Windows system path. Go back to [step 1.3](#) and make sure you did it correctly. Please watch the video if you are struggling with it. When you think you’ve finished this step, you’ll need to close and then re-start the command prompt to check that you installed it correctly.

**Error: a JNI Error has occurred, please check your installation and try again**

This probably means that you had another java executable in your path before you started installing java yourself. Go back to step 1.3 and do the following:

1. Open up the Environment Variables like you did before
2. Look for some other java path that is different from whatever you added. In my experience, this is usually something along the lines of Oracle javapath
3. Click on the foreign java element
4. On the left, click the “Move Down” button. Keep clicking it until the foreign java element is at the very bottom of your list of path variables.

**error: illegal character: ‘u201c’**

Sometimes when you copy/paste code from one editor to another, something gets, for lack of a better term, “lost in translation”. In this case, the quotation marks from google slides and/or google docs may not match the quotation marks in your text editor.

The error comes from the following line:

```
System.out.println("I, [INSERT NAME], agree to the academic integrity  
policy");
```

In particular, the two quotation marks (**I, [INSERT NAME], agree to the academic integrity policy**), are causing the issue. Luckily the fix is simple:

1. Delete the two quotation marks
2. Manually re-type the two quotation marks in the same place that they were before. **Make sure you have only one set of quotes on each side of the text.**
3. Compile and run the program again. The errors should be gone.

**Note:** if you saw this error, then you probably have a bunch of other errors, but those are all red herrings. This error is the root cause.