

# Setting Up Development Environment

## [Setting Up Git and GitHub Environment](#)

[Git and Github Overview](#)

[Steps to set up git and a github repository](#)

[Git and GitHub Resources](#)

## [Setting up Eclipse Development Environment](#)

[Windows Setup](#)

[Step 1: Download SE Development Kit and Install](#)

[Step 2: Download Eclipse \(Windows\)](#)

[Step 3: Setup Environment Variables](#)

[Step 5: Set Eclipse](#)

[Mac Set Up](#)

[Step 1: Download Java SE Development Kit](#)

[Step 2: Set Up Eclipse](#)

## [Creating a Java Program in Eclipse](#)

[Step 1: In Eclipse Create a Java Project](#)

[Step 2: Create, Compile and Run a Java Class File](#)

[Step 3: Github Desktop: Version and Backup](#)

[Eclipse Resources and Tips](#)

## [Troubleshooting Eclipse](#)

[Issue #1: Exit code 13 error](#)

[Issue #2: Project has a red !](#)

## Setting Up Git and GitHub Environment

### Git and Github Overview

Git and GitHub are important elements of the professional software development world with version control tools that help individuals and teams of developers manage public open-source projects and private projects. You will be introduced to Git as a version control tool and GitHub server to store your versioned files. This allows You can manage your code from multiple computers.



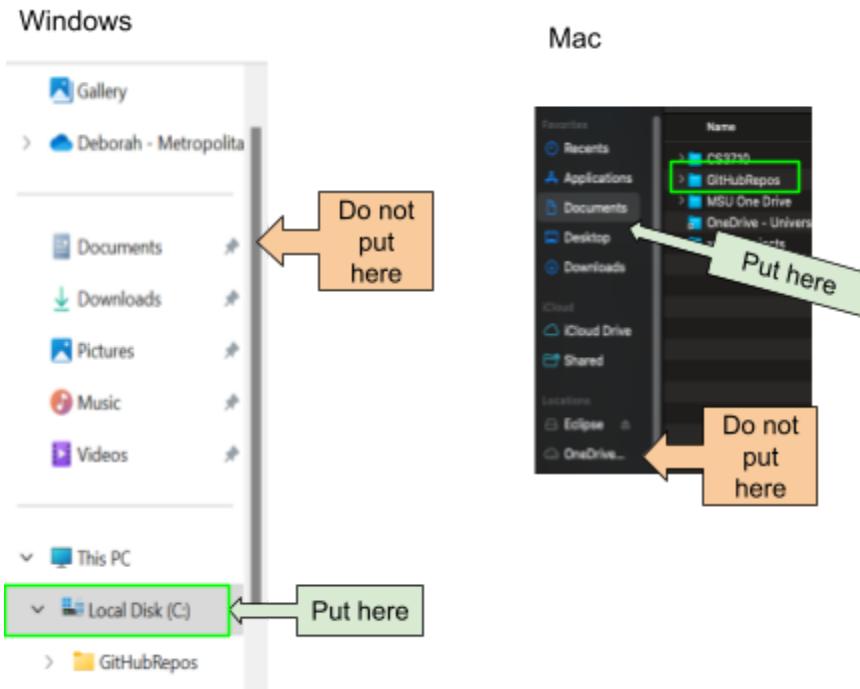
Watch this [video](#) before you follow the steps to help you understand more about git and github. **Do not do the steps in the video and instead follow the steps below.**

You will be using github desktop to manage the file versions on your computer and upload them to

github. **Do not** use your IDE to version your code with git.

## Steps to set up git and a github repository

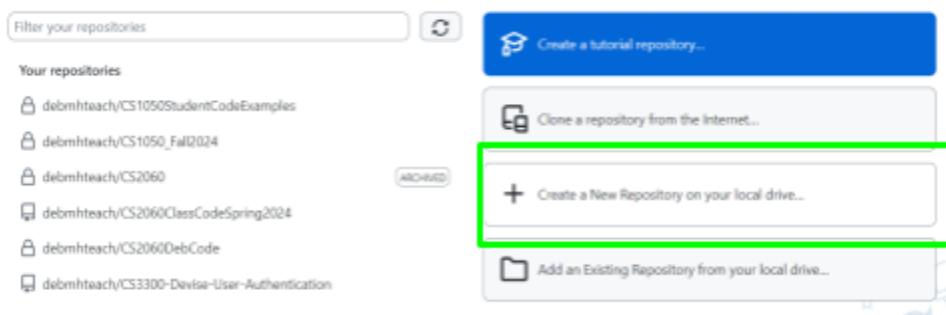
1. Create a folder GitHubRepos for your github repositories. I find it best to not put in my one drive folders. I find on **windows** it is best to put in your C drive. On Mac I put it in my documents but not in OneDrive.



2. Download [GitHub Desktop](#) to connect your git code version to the GitHub server. This allows you to version locally on your computer and push to the Github cloud server.
3. Create a free GitHub [account](#) if you do not have one. Select Join for free. Follow the prompts to create your personal account or organization. You may want to use your personal email rather than school email. You will need to verify your email.
4. After verifying your email you can set up your account. There may be some questions to answer. Do not worry if you are not sure what they are asking. You can just answer what is required.
5. Open github desktop, click Add, Select Add Existing Repository

Let's get started!

Add a repository to GitHub Desktop to start collaborating



6. Fill information to create a repository

Create a new repository

Name: CS1050\_Fall2024      Name of repository Using course name and semester

Description: CS1 Java

Local path: C:\GitHubRepos      Choose...      Choose GitHubRepos folder you created above on your local computer

Initialize this repository with a README:  Check

Git ignore: Java      Select Java

License: None

**Create repository**      **Cancel**

7. Read the information about publishing you repository and click Publish Repository

## No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



### Publish your repository to GitHub

This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.

**Publish repository**

Always available in the toolbar for local repositories or **Ctrl + P**

8. Keep this code private so you are not sharing your solutions with others for this class and Publish Repository.

Publish repository

GitHub.com      GitHub Enterprise

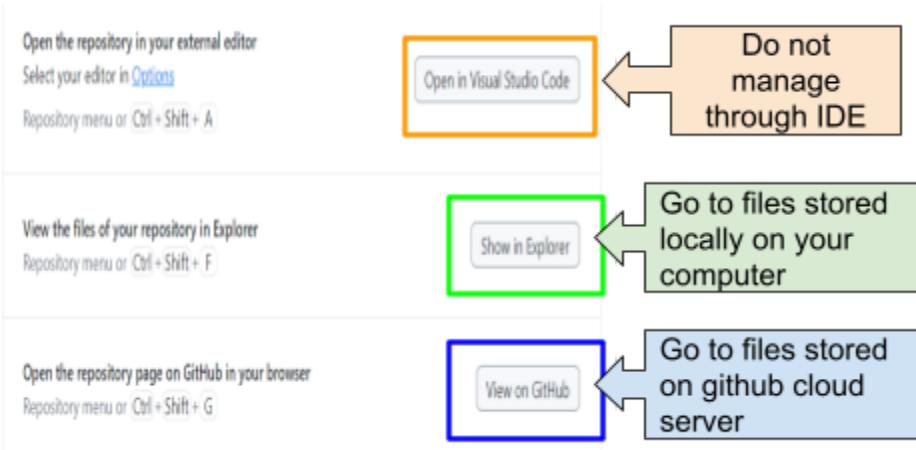
Name: CS1050\_Fall2024

Description: CS1 Java

Keep this code private

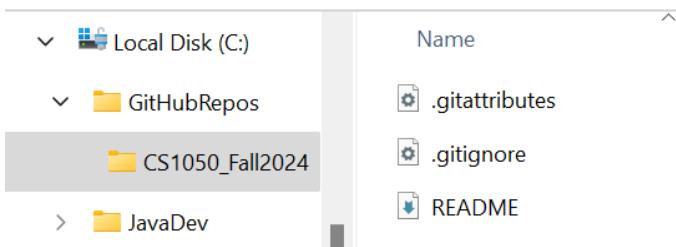
**Publish repository**      **Cancel**

9. Do not open the repository with your external editor. You will manage your files in this class with the Github desktop so you can understand the different git commands. Show your files locally on your machine.

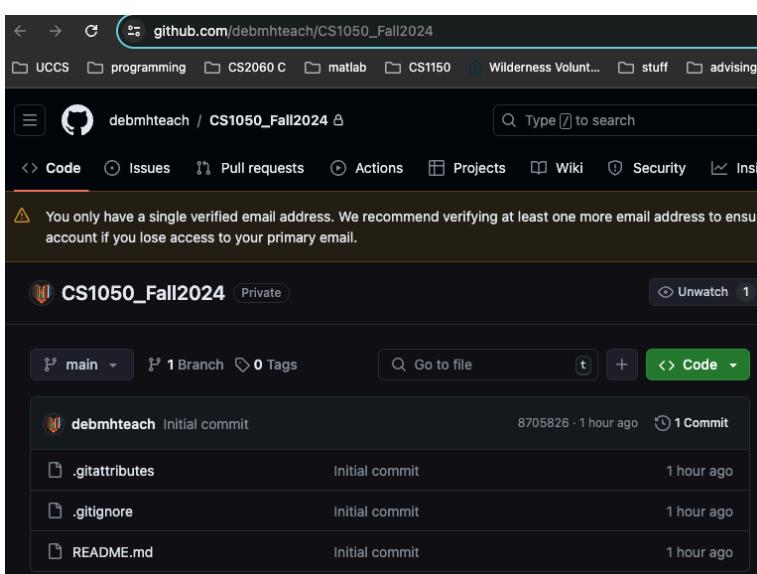
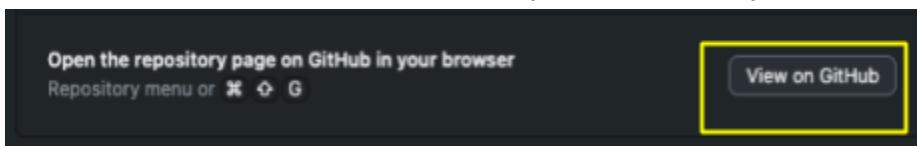


When you show your files in the finder you will see the following. If you do not see the .git information you can reveal hidden files.

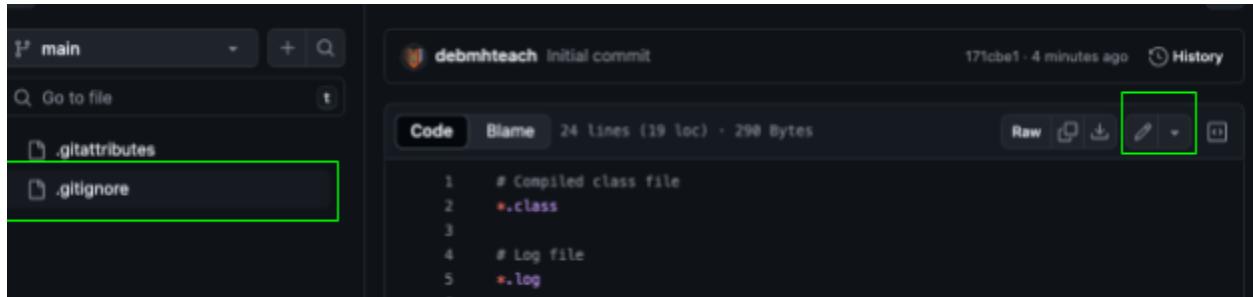
- **Mac:** To reveal hidden files in Finder, go to Go > Computer > Macintosh HD and press Shift + Command + . (period).
- **Windows:** [View hidden files and folders in Windows - Microsoft Support](#)



10. Click View on Github to see the repository was pushed to your cloud GitHub server.



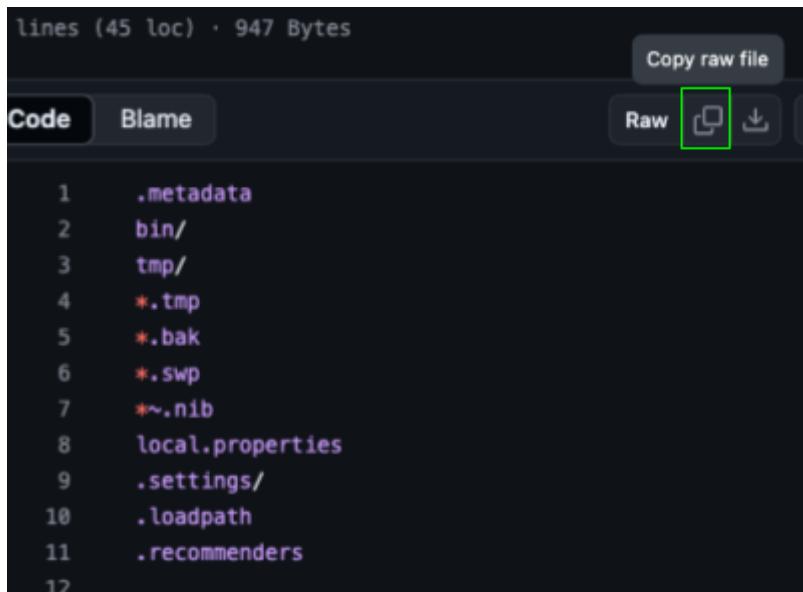
11. Open the .gitignore file and click click to edit.



```
# Compiled class file
*.class
#
# Log file
*.log
```

12. This contains the files associated with developing java applications that should not be versioned and stored in the repository. This was added when you selected java for the gitignore when setting up your repository. Now you will need to add what to ignore for Eclipse and Mac.

1. Go to [gitignore/Global/Eclipse.gitignore at main · github/gitignore · GitHub](https://github.com/github/gitignore/blob/main/Eclipse.gitignore).  
Copy and Add the information to the end of the gitignore.



```
.metadata
bin/
tmp/
*.tmp
*.bak
*.swp
*~.nib
local.properties
.settings/
.loadpath
.recommenders
```

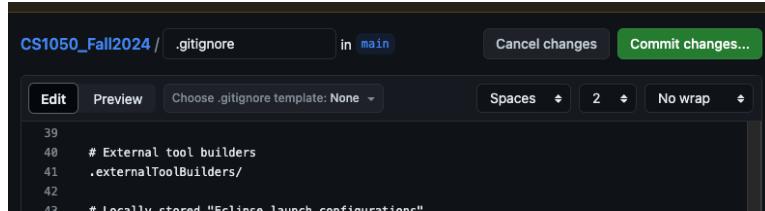
2. Add the following below that information

```
# Eclipse Core
.project

# JDT-specific (Eclipse Java Development Tools)
.classpath
```

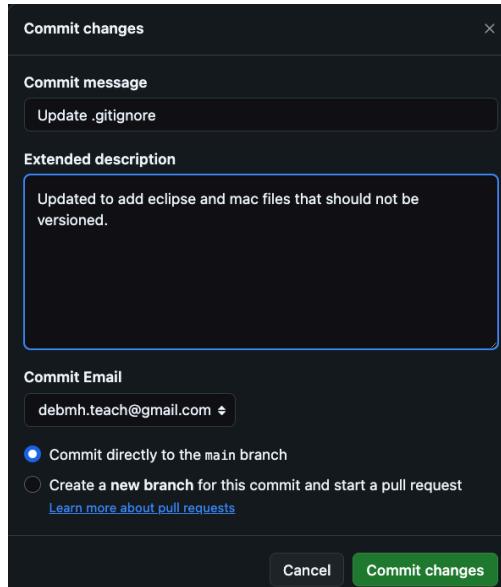
3. If you are using a Mac, I add more to the .gitignore file that I do not want to be versioned and stored in the repository. Go to <https://github.com/github/gitignore/blob/main/Global/macOS.gitignore>

4. Click to commit changes. This means create a new version.

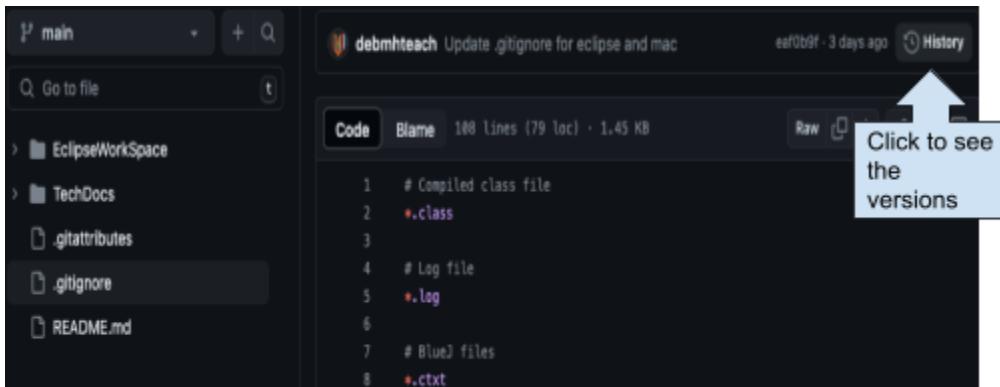


```
CS1050_Fall2024 / .gitignore in main
Cancel changes Commit changes...
Edit Preview Choose .gitignore template: None
Spaces 2 No wrap
39 # External tool builders
40 .externalToolBuilders/
41
42 # Locally stored "Eclipse launch configurations"
43
```

5. Update the commit message to be more informational as shown below.



6. Go to your repository on the github remote server to see the changed file and two versions.



Create good descriptions for example update .gitignore for eclipse and mac

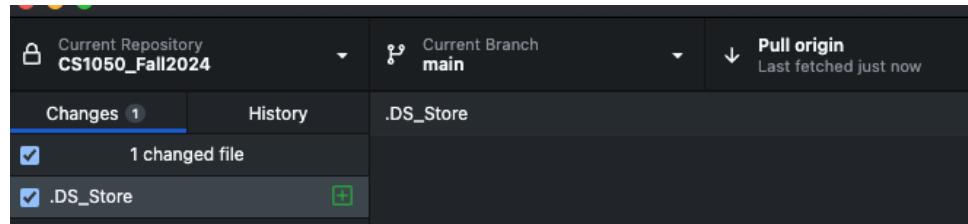
Commits

History for CS1050\_Fall2024 / .gitignore on main

Commits on Jul 19, 2024

- Update .gitignore for eclipse and mac** (Good)
- Update .gitignore** (Bad description)
- Update .gitignore** (Good)
- Mac ignore files** (Good)
- Initial commit**

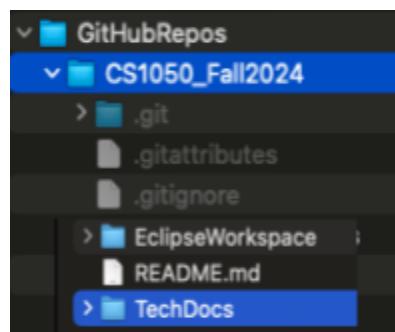
7. Go back to your github desktop and you will need to pull origin to download the updated file to your local computer. Click Fetch and then Pull Origin



8. Go to your local CSCourse\_Semester folder to store your github repository and add these directories: EclipseWorkSpace, TechDocs.

#### File Structure

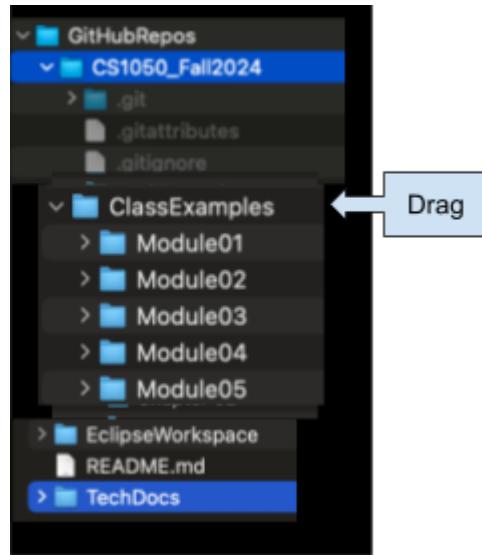
- CSCourse\_Semester
  - EclipseWorkSpace
  - TechDocs



9. Download the zip file containing class java examples

- a. [ClassExamples.zip](#)

10. Unzip, open ClassCodeExamples, drag the folder into your CSCourse\_Semester repository folder you created. You should see



11. Go back to your GitHub Desktop and you will see that there were many changes.  
You need to version the changes **locally on your machine**.
  - a. You need to add a comment above description “Add Book Examples”
  - b. The click Commit to main

Changes 300+ History BookExamples/Appendix/E/BankAccount/allclasses-frame.html

1091 changed files

BookExa.../allclasses-frame.... [+]

BookE.../allclasses-noframe.... [+]

BookExampl.../BankAccount.... [+]

BookExampl.../BankAccount.... [+]

BookExa.../constant-values.... [+]

BookExa.../deprecated-list.h... [+]

BookExamples/A.../help-doc.... [+]

BookExamples/A.../index-all.... [+]

BookExamples/Appe.../index.... [+]

BookExampl.../overview-tree.... [+]

**Added BookCode**

Description

Commit to main

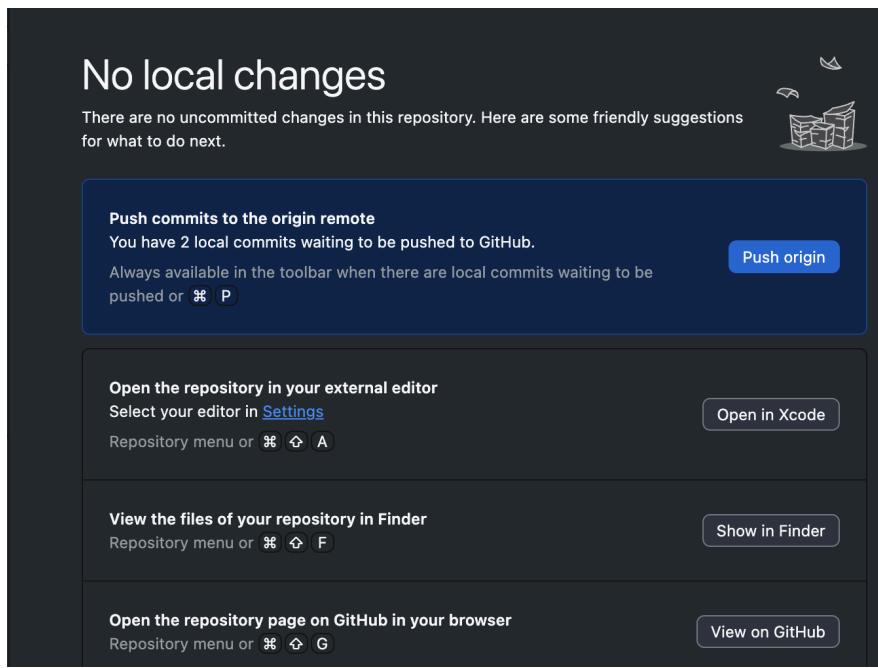
Committed just now

```

@@ -0,0 +1,31 @@
+ <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional
R/htm14/loose.dtd">
+ <!--NewPage-->
+ <HTML>
+ <HEAD>
+ <!-- Generated by javadoc (build 1.6.0-beta2) on Thu Dec
6, 2006 at 11:45:45 EST -->
+ <TITLE>
+ All Classes
+ </TITLE>
+
+ <META NAME="date" CONTENT="2006-12-07">
+
+ <LINK REL ="stylesheet" TYPE="text/css" HREF="stylesheet.css">
+
+ <HEAD>
+
+ <BODY BGCOLOR="white">
+ <FONT size="1" CLASS="FrameHeadingFont">
+ <B>All Classes</B></FONT>
+
+ <BR>
+
+ <TABLE BORDER="0" WIDTH="100%" SUMMARY=">
+ <TR>
+ <TD NOWRAP><FONT CLASS="FrameItemFont"><A HREF="BankAcc
1t:Unnamed&gt;" target="classFrame">BankAccount</A>

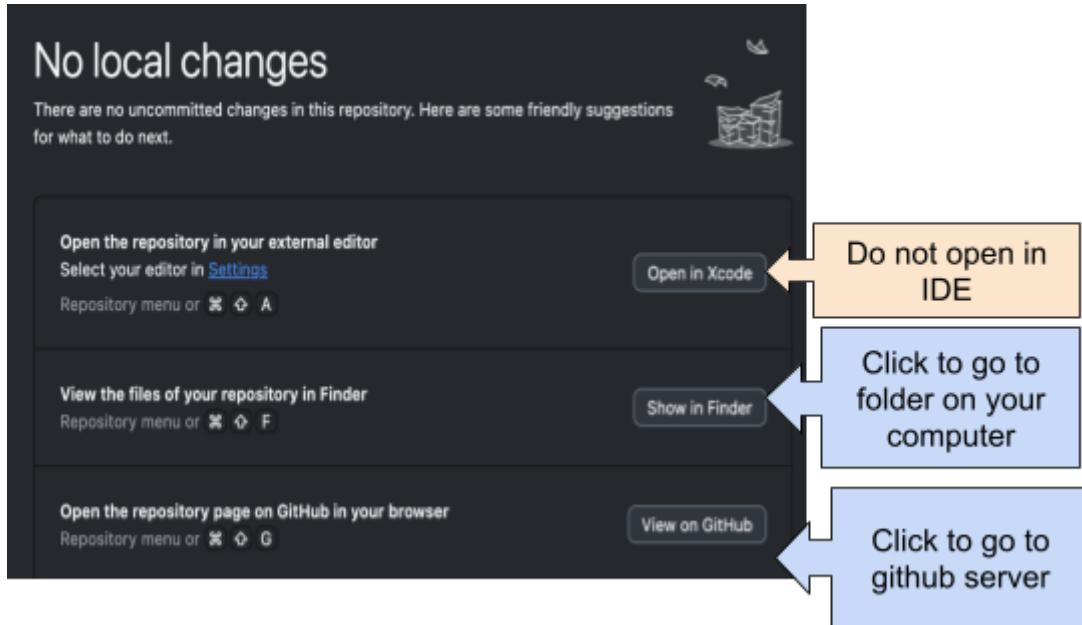
```

12. Next you need to push the changes to your github remote server by clicking push to origin. Now they are versioned on your local computer and on the Github remote server.



13. Now you will see two or three options

- Do not select open in external editor
- Click to show in finder to see where your files are on your local machine
- Then click view on github to see the files backed up on the GitHub server



14. Remember to use github desktop only to do the following

- commit to version with git on your local computer
- push to backup all your work and versions on github server

## Git and GitHub Resources

- [Getting started with GitHub Desktop](#)
- [github docs](#)
- [About Git - GitHub Docs](#)
- [Cloning a repository](#)
- [Pushing commits to a remote repository](#)
- [Git Guides - git pull · GitHub](#)

## Setting up Eclipse Development Environment

**Need two things to write Java programs using Eclipse**

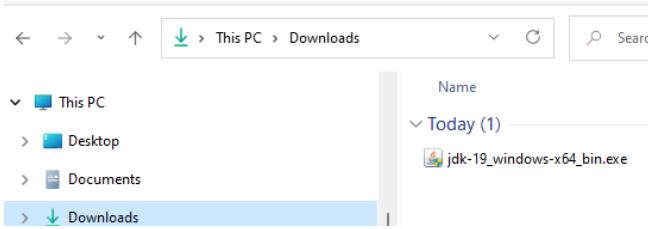
- 1) Java Runtime Environment (JDK)
  - a. Eclipse is a Java based IDE so 1<sup>st</sup> thing you need to do is install the **JDK**
- 2) Eclipse
  - a. Eclipse is an integrated development environment used to create Java programs

Click to go to [Windows Setup](#) or [Mac Set Up](#)

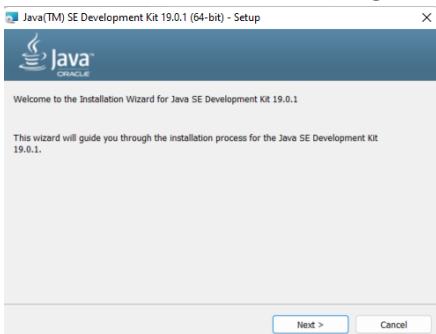
# Windows Setup

## Step 1: Download SE Development Kit and Install

1. [Download this file.](#)
2. Once the download completes, start the installation by double clicking on the downloaded .exe in the browser or in the directory it is saved in.



3. You will be asked if you want to allow this app to make changes to your device - click **Yes**. Note: The installation can take a bit to get started – it can be slow - so be patient.
4. You will see the **Welcome** dialog – click **Next** to start download



5. The next dialog offers some options – click **Next**.
  - o Notice that the JDK will be installed in the **Program Files** directory
  - o This is where it should be installed so don't change the location

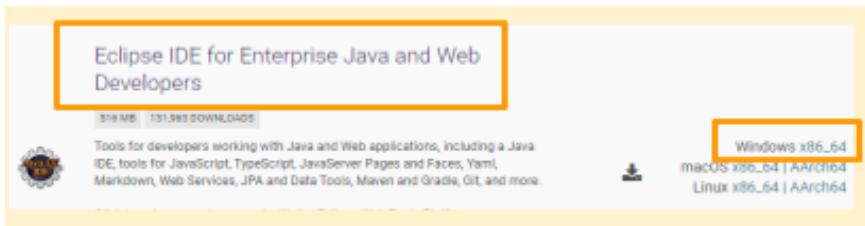


6. When installing the status window may or may not display. – be patient it takes time to get started. When the window shows successfully installed click close.



## Step 2: Download Eclipse (Windows)

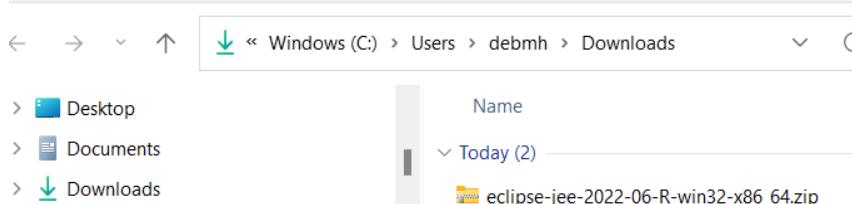
1. Go to: <http://www.eclipse.org/downloads/packages/>
2. Scroll down to Eclipse IDE for Enterprise Java and Web Developers (DO NOT use the installer)
3. Do not worry about the version if they are different than these screenshots



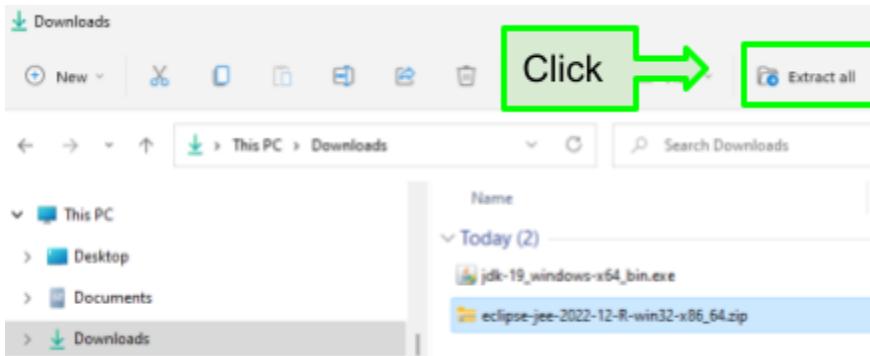
4. You should end up on a page with a big download button - click that button!



5. Windows The **Eclipse zip file** will download into the **default download location** (the “download” folder in your “user” account) unless you have set up your browser to save to a different location.

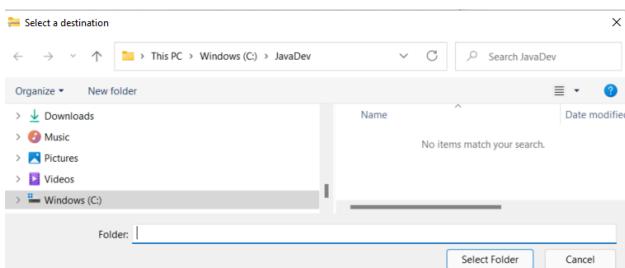


6. Eclipse is a bit different since you don't actually “install” it. Instead, because it is a zip file you will extract it to a directory of your choice.



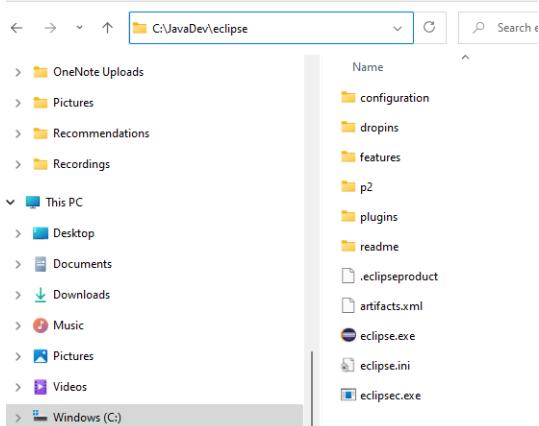
7. The “Select a Destination and Extract Files” dialog will be displayed. In the dialog, select where to place Eclipse. Click Browse and read below. Select show extracted files when complete. The unzip will start after you click extract– this could take a while depending on your system.
- On my personal systems, my preference is to create a **“JavaDev”** directory on my C drive and unzip Eclipse there - **C:\JavaDev**.
  - When you start to do more development, having a development directory on your machine helps keep all development tools in one place.
  - **NOTE: Placing the eclipse file too deep into the file structure causes an error.**

You can click new folder to create a JavaDev folder



8. When the unzip completes, go to the location you selected and view all the files

- On my system that location is **C:\JavaDev\.eclipse**
- Note: if you ever need to uninstall Eclipse, just delete the eclipse directory.
- Note, when reinstalling or updating eclipse, make a backup copy of the eclipse workspace directory in case you run into issues!



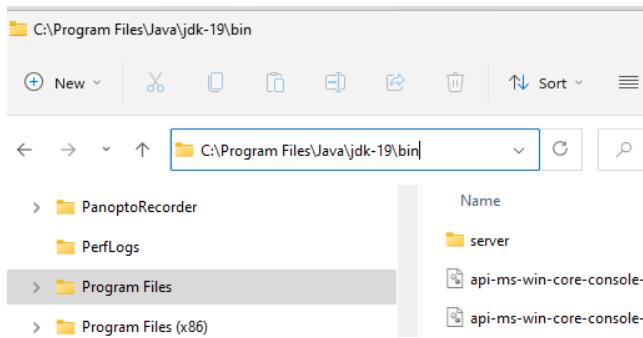
## Step 3: Setup Environment Variables

**Note** that your version will probably be 22 now so just follow instructions for jdk-22

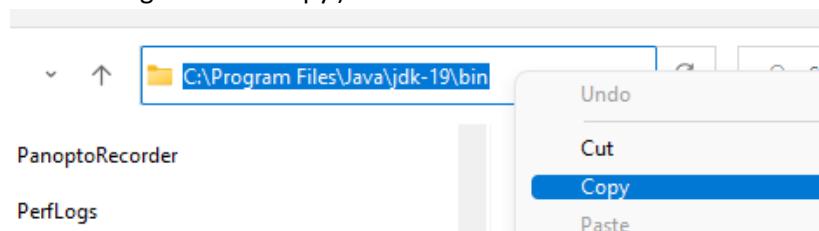
1. Find where the JDK is on your hard drive.
2. It should be in the **Program Files** directory in the **Java** directory:

**C:\Program Files\Java\jdk-22**

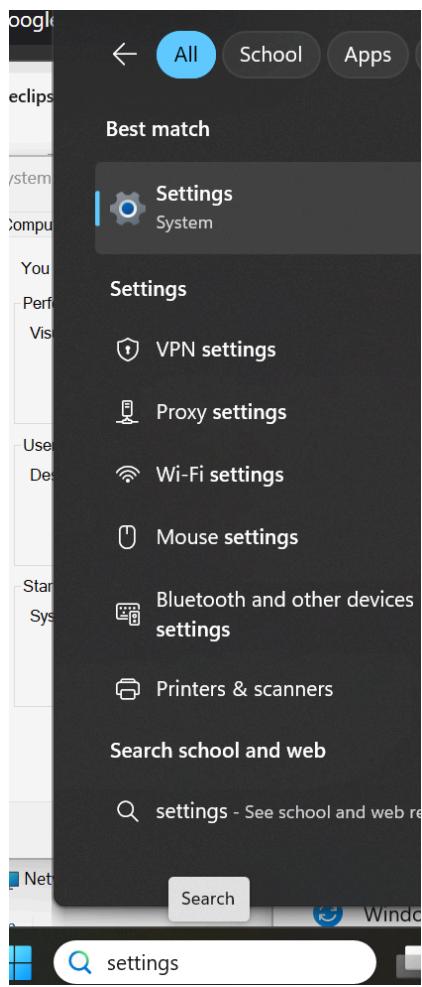
3. Go one more level down to the **bin** directory



4. In the File Explorer window, click in the little window where the path is shown which will highlight the path.
  - o Make sure the path includes the **bin** directory: **C:\Program Files\Java\jdk-22\bin**
  - o Copy the path to the **bin** directory (since it's highlighted use CTRL-C to save it in the buffer or right click to copy.)



5. Search for settings and open.

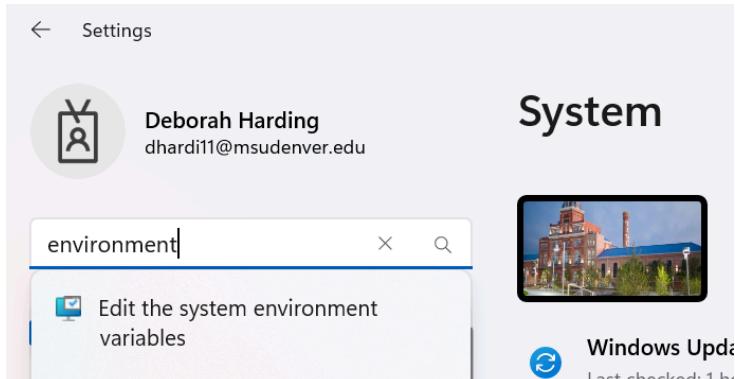


6. A new window will appear showing your system information. On the right, scroll down to **Related Settings** and click "**Advanced system settings**". Might be different based on the OS version so see below for windows 10 and windows 11.
  - o windows 10

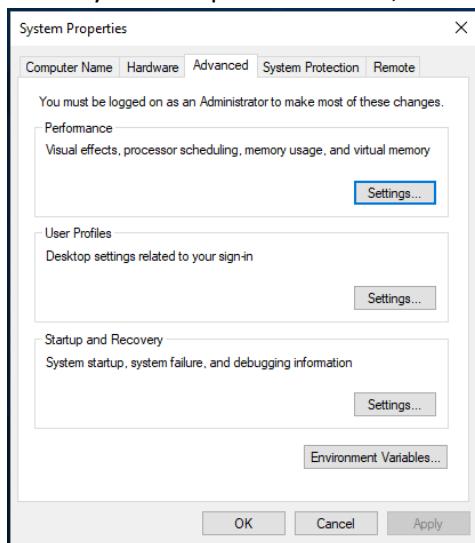
#### Windows 10

A screenshot of the Windows 10 Settings app. The left sidebar shows various system settings like Display, Sound, Notifications &amp; actions, Focus assist, Power &amp; sleep, Storage, Tablet, Multitasking, Projecting to this PC, Shared experiences, and Clipboard. The main pane is titled 'About' and includes sections for Support (Manufacturer: HP, Website: Online support) and a note stating 'This page has a few new settings' with a link to 'View more info'. In the 'Related settings' section, there are links to BitLocker settings, Device Manager, Remote desktop, and System protection. The link 'Advanced system settings' is highlighted with a green box and a red arrow pointing to it.

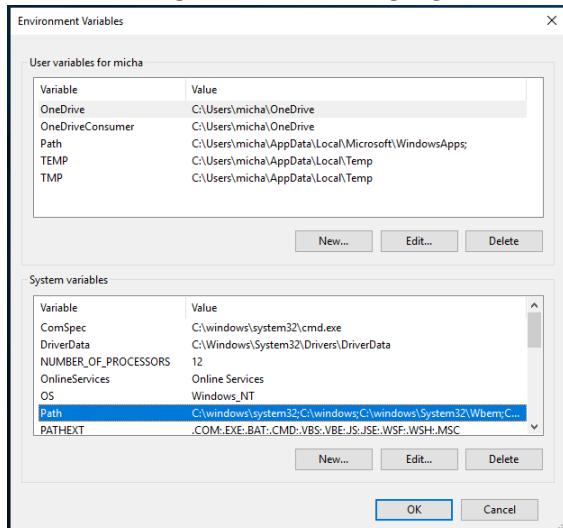
- Note if you have Windows 11, I searched for environment and selected Edit the system environment variables.



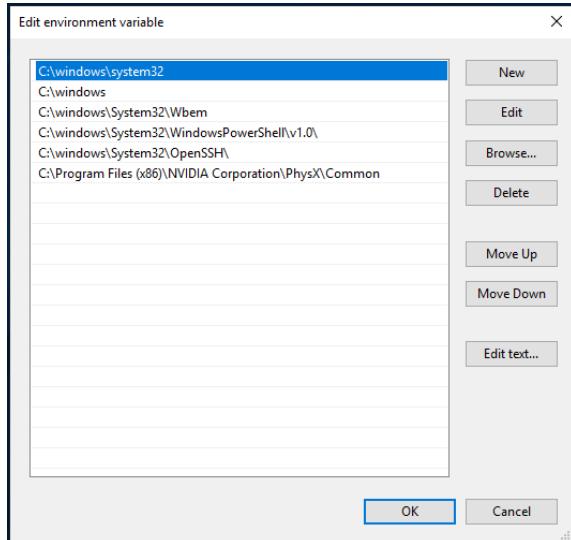
- In the System Properties window, click the "Environment Variables..." button



- The Environment Variables window will display. In the lower window - "System Variables" - find "Path" and single click on it to highlight

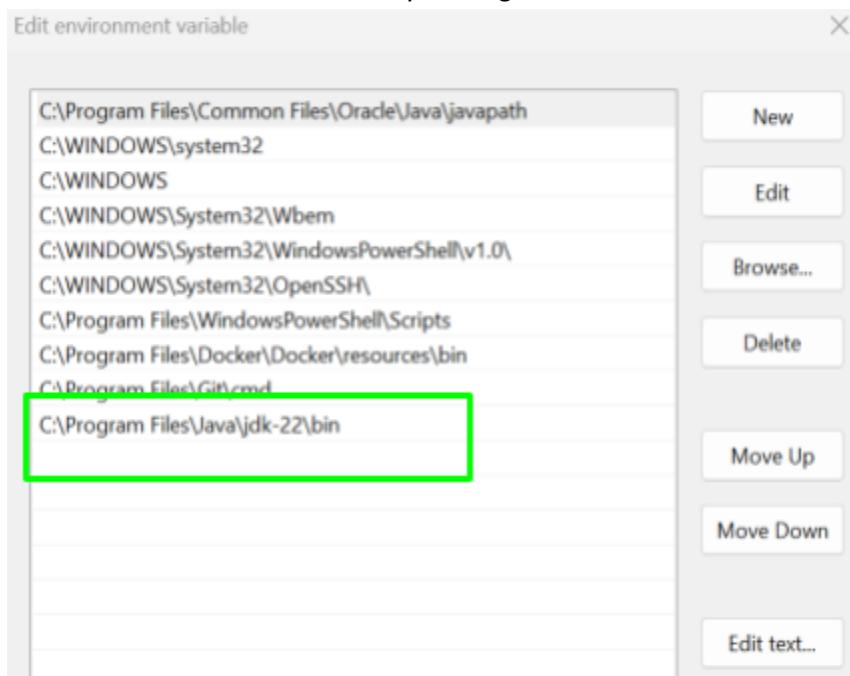


9. Once "Path" is highlighted, click **Edit** and this window will appear.



10. Click **New**. Paste the path for the **JDK bin** directory into the little window adding it to your path.

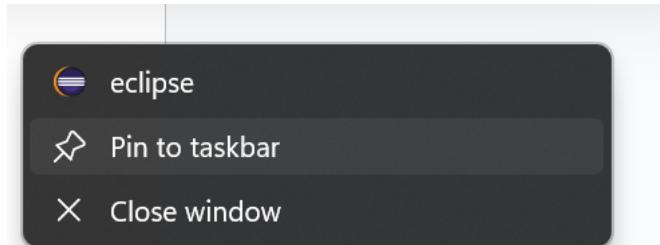
- o Click on the little window
- o Use CTRL-V to paste the path saved in the buffer into the little window
- o Close all those windows by clicking OK



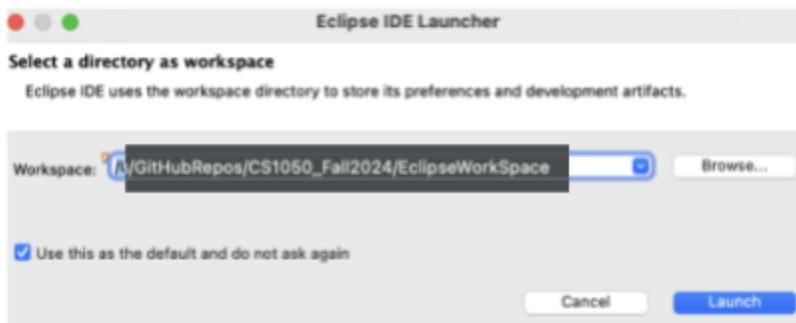
## Step 5: Set Eclipse

Windows

1. Go to the directory where you extracted Eclipse (on my system that would be `c:\JavaDev\eclipse`).
2. Double click the **eclipse.exe** file
  - o If you don't see the **.exe** file extension – double click the file with the round purple icon
  - o To see file extensions in file explorer, click "View" then check "File name extensions"
  - o You can right click eclipse and pin to taskbar to make it easier to launch



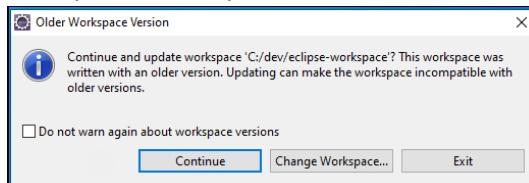
3. Eclipse will ask where you want your **workspace**. A **workspace** is where the `.java` and `.class` files for your class work and assignments will be stored. You will put this in the `EclipseWorkSpaceFolder` you created so you can version the files using git and back up on the GitHub server. Click use this as the default.



4. Click launch

**Note:** if you're updating the version of Eclipse on your machine you will see the following dialog

- o Note: Before you update Eclipse to a newer version make a backup copy of your current workspace in case you run into issues.



5. The Welcome screen for Eclipse will appear. Click the **Hide** icon on the right side.

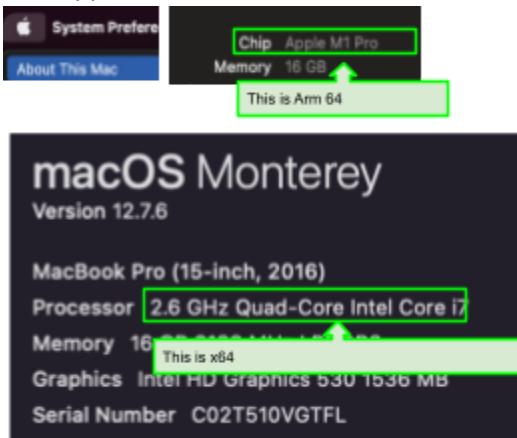


6. You are now set up to create, compile and run Java programs! Go to [Creating a Java Program in Eclipse \(Windows and Mac\)](#)

# Mac Set Up

## Step 1: Download Java SE Development Kit

1. First you need to find out if your Mac is using X64 (x86-64) architecture or ARM 64 (M1 or M2).
  - Click apple and select about this Mac. If the chip is M1 or M2 you are using ARM 64.

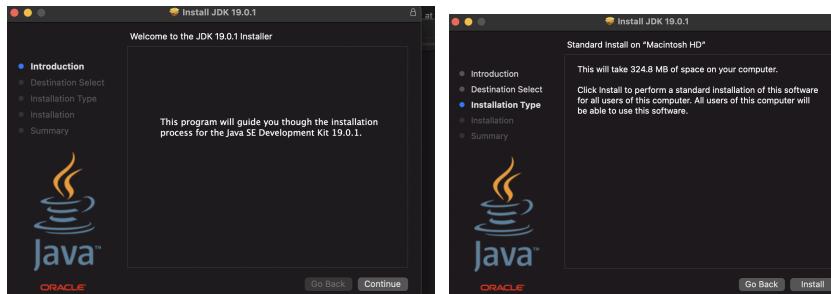


2. the access to the website for this step has changed so if you need
  - if using M1 or M2 chip download the [ARM 64 download this file](#)
  - otherwise download this file [jdk-22\\_macos-x64\\_bin.dmg](#)

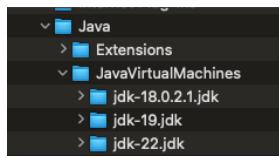
3. Double click the JKD pkg



4. The install window will appear. Click continue and then install. Enter password to install the software. Then click close



5. Here is where the JDK is located. You may have more than one installed. This will be addressed later.

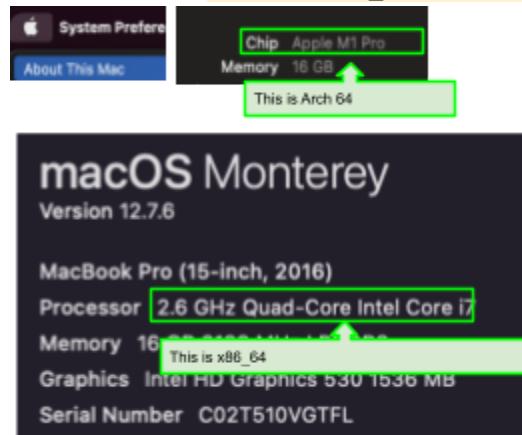


## Step 2: Set Up Eclipse

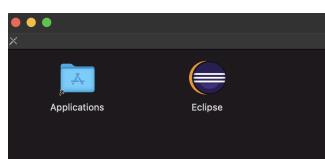
Some of the screenshots may look different as they are from an earlier version.

7. Go to: <http://www.eclipse.org/downloads/packages/>
8. Scroll down to Eclipse IDE for Enterprise Java and Web Developers (DO NOT use the installer)

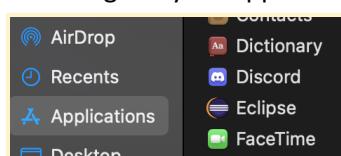
9. When I installed, the current version was: 2022-12R Don't worry if the version is different. On the right-hand side, click "Mac OS X86\_64 or if new core processor AArch64")



10. You should end up on a page with a big download button - click that button!
11. Go to the download location. Double click dmg file.
12. Drag eclipse into your application folder.

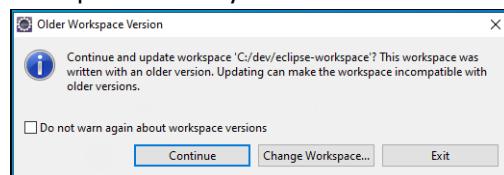


13. You can go to your applications folder to open Eclipse



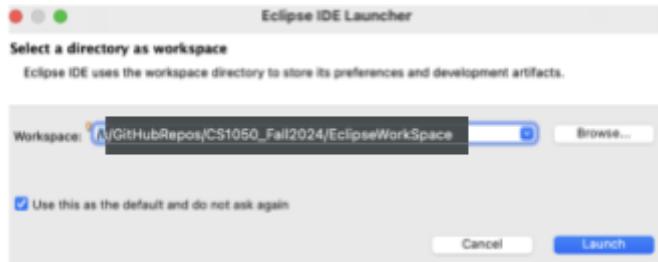
**Note: if you're updating the version of Eclipse on your machine you will see the following dialog**

- Before you update Eclipse to a newer version make a backup copy of your current workspace in case you run into issues.

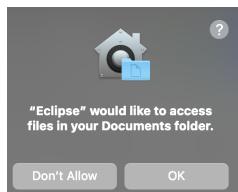


14. Eclipse will ask where you want your **workspace**. A **workspace** is where the .java and .class files for your class work and assignments will be stored. You will put this in the

EclipseWorkSpaceFolder you created so you can version the files using git and back up on the GitHub server. Click use this as the default.



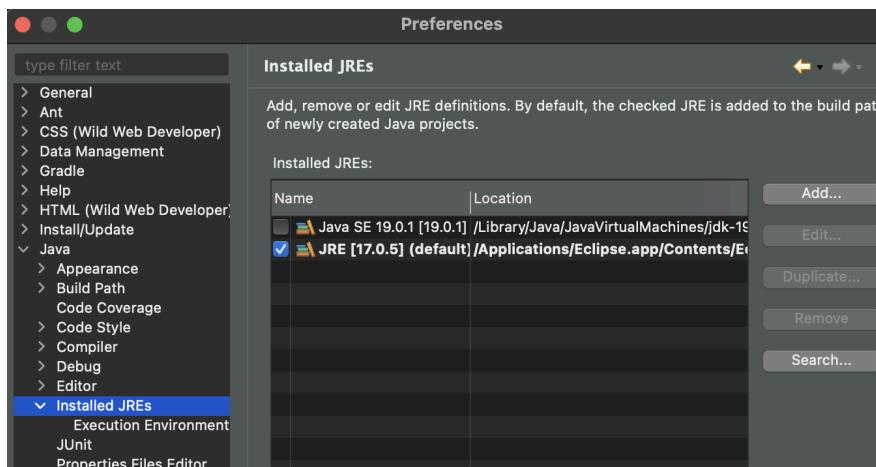
15. Click ok to allow eclipse access.



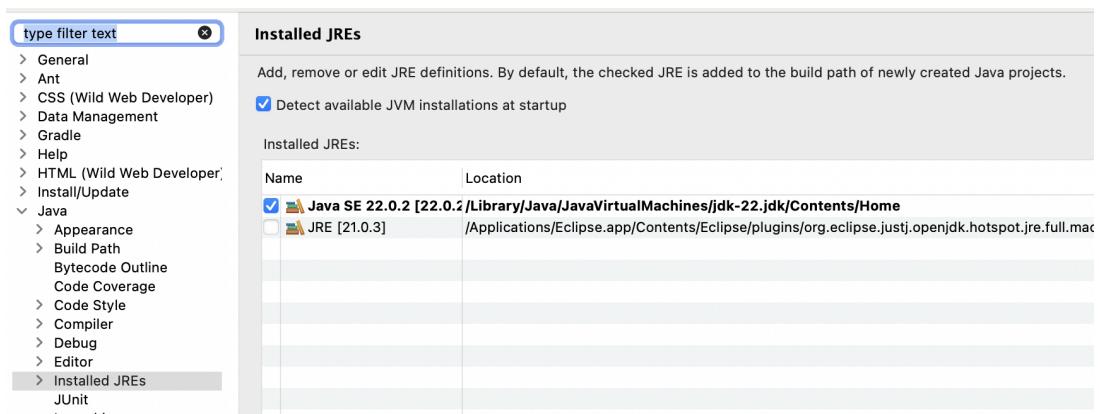
16. The Welcome screen for Eclipse will appear. Click the Hide icon on the right side.



17. Set up JRE. Go to Eclipse menu, select Preferences and navigate to Java -> Installed JREs



18. Scroll down to Java->Installed JREs. Select Java SE #. Click apply and close.



19. Now that the Java SE is set go to [Creating a Java Program in Eclipse \(Windows and Mac\)](#) below.

## Creating a Java Program in Eclipse

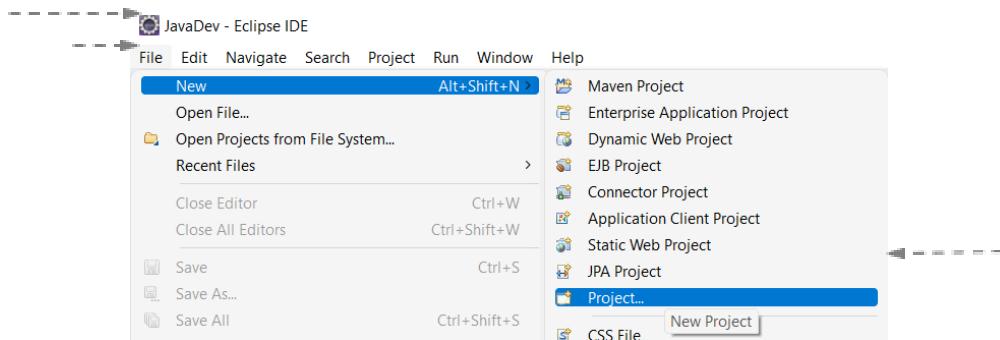
Here are the steps for creating a simple Java program to get started

- Quick look at programming from a general view
- Everything will be covered in more detail in later lectures

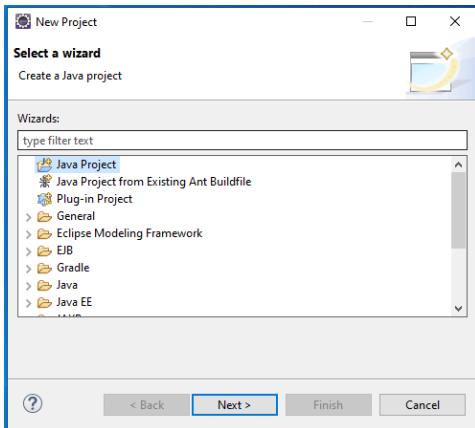
## Step 1: In Eclipse Create a Java Project

We will look to have a project folder for exploring code called cs1050classwork. Later you will make more projects to organize code for different types of assignments. This is to help organize your different source code for this course.

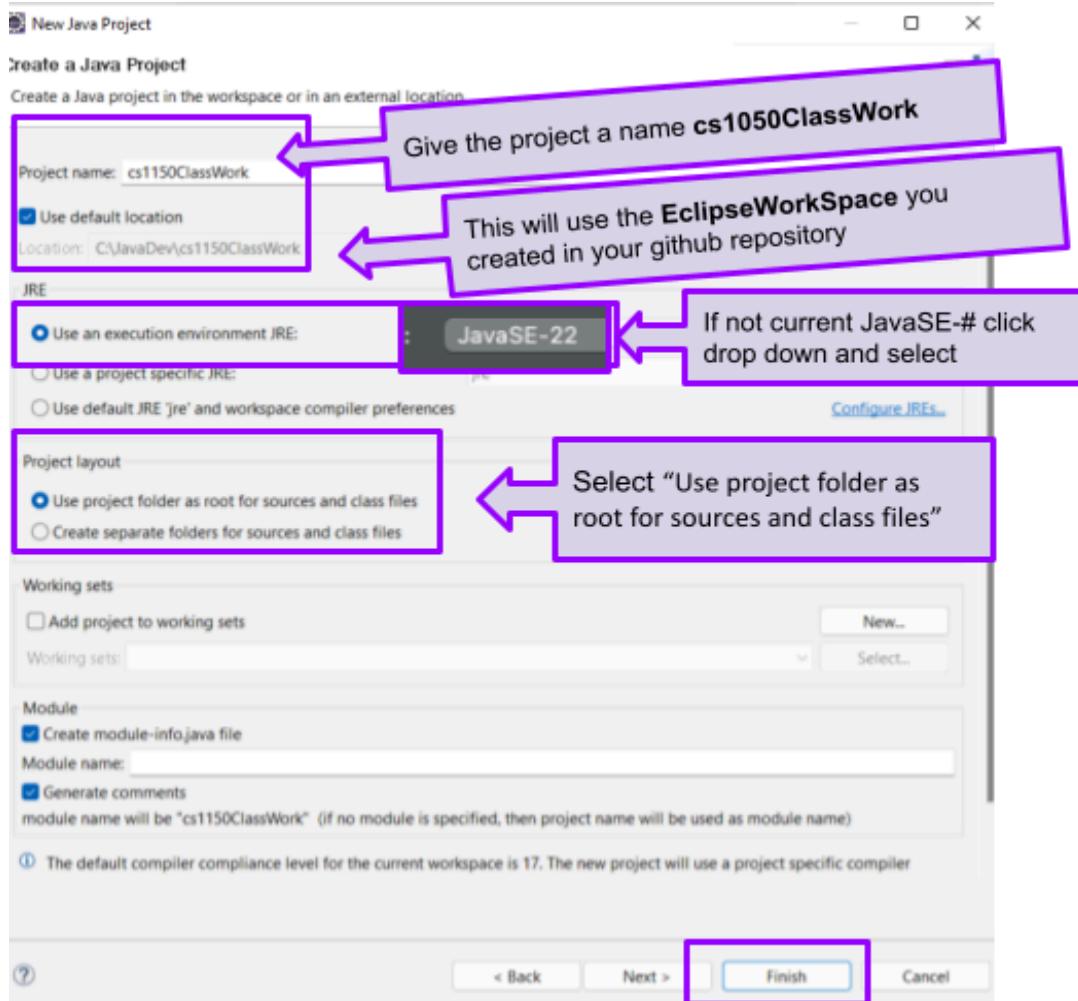
1. The first time you create a project you will see a slightly different menu than on future creations.
2. On the top menu, select **File->New-> Project** to display the New project dialog



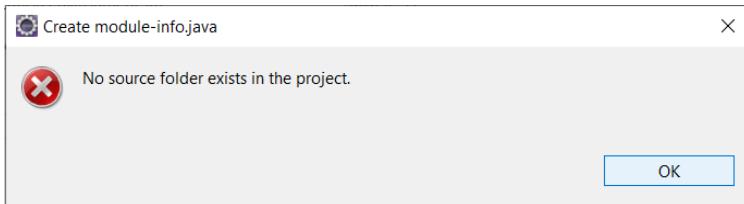
3. In the New Project dialog that appears, select **Java Project** then click **Next**.



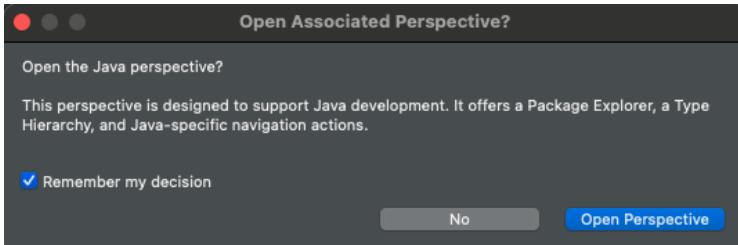
4. Set the following in the **Create Java Project** dialog and then click **finish**. You will use this same process to create future projects for other class assignments.



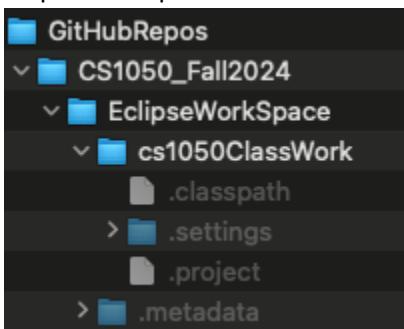
5. You may see the following dialog – click “OK”



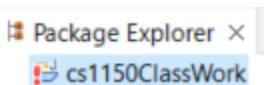
6. You may also see a dialog that asks you about the Java perspective. Click Remember by decision and click “Open Perspective”



7. If Welcome appears close it
8. If you go to your local repository you will see the project was created in the EclipseWorkSpace



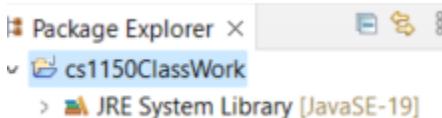
1. **IMPORTANT NOTE:** before proceeding please read:
  - o If there is a red ! next to the project name in Eclipse, your environment is **NOT** setup properly. See the [troubleshooting section](#) at the end of this document for simple steps to fix this **before you create a Java class**. If you don't resolve this issue, you will be unable to get your code to work. After you fix this go to the next step to create a java class.



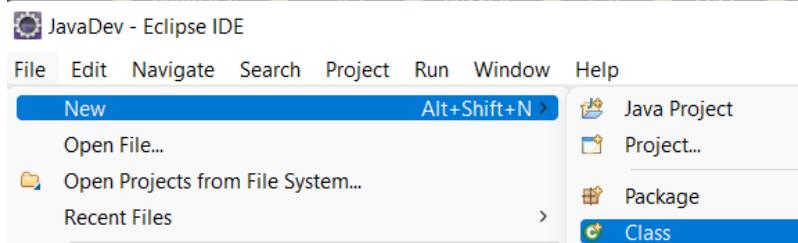
## Step 2: Create, Compile and Run a Java Class File

You will create a new java class in the project folder you want. You must do the following everytime you want to create a new java source file.

1. Make sure you have selected the project folder you want to create it in, selected.

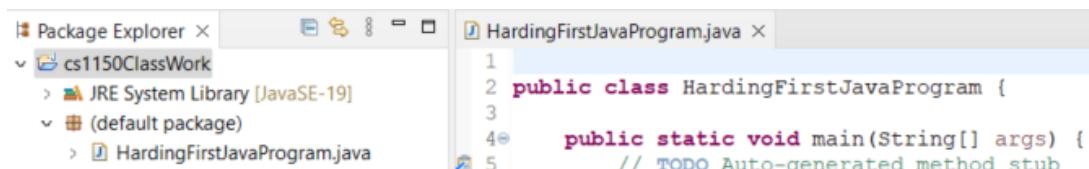
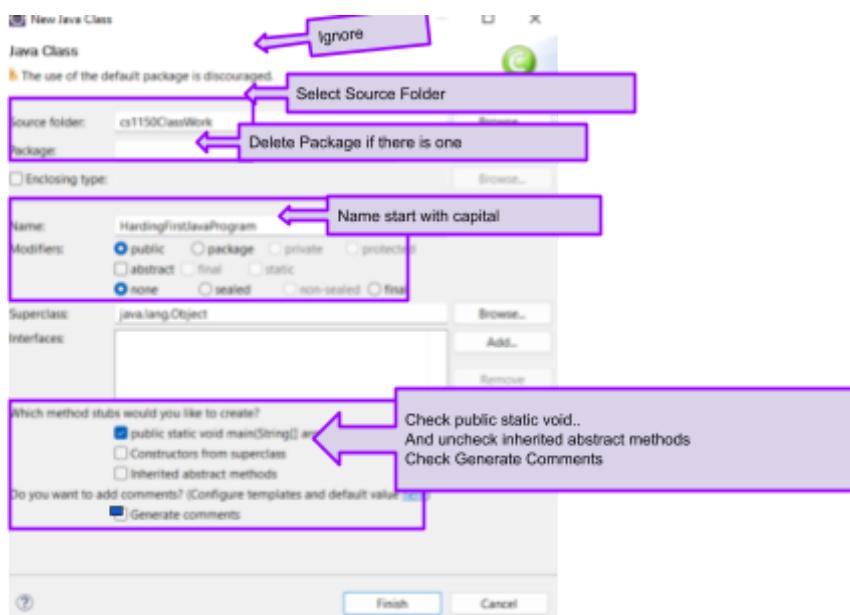


2. Go to **File->New->Class** (or **File->New->Other** then pick **Class** in the dialog)



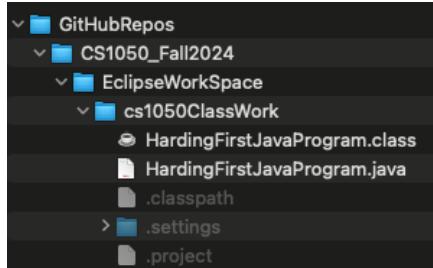
3. Set the following (see screen below)

- Source Folder – should be filled in with the name you gave your project
- **Package should be empty**
- Name - name for file starts with capital letter
- Which method stubs would you like to create?
  - Check: **public static void main(String[] args)**
  - Uncheck: Inherited abstract methods
- Check Generate Comments
- Click Finish

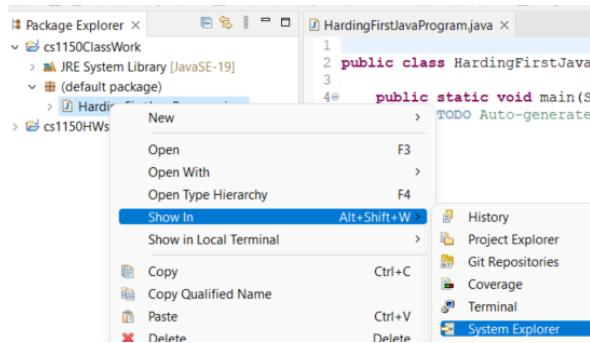


5. You will repeat this step #3 (creating a java class) each time you want to create a new file.

6. Go to your project located in you local repository

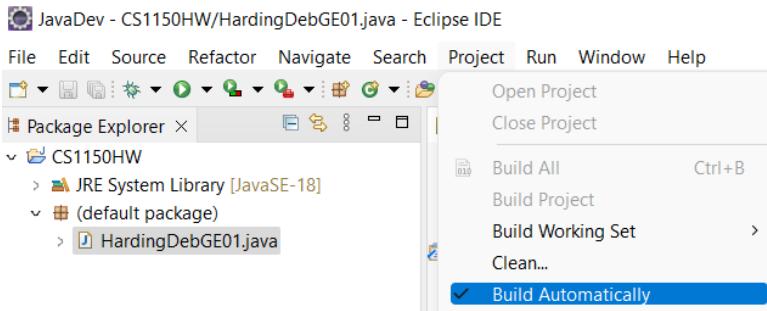


Another way to get to the directory is to right click the file then go to Show In -> System Explorer



7. Eclipse automatically built the .class file

- o In Eclipse, if you go under Projects, you will see “Build Automatically” is selected
- o This is how the .class file was created automatically for you.



8. Add code to your first file. **System.out.print("Hello Java");**

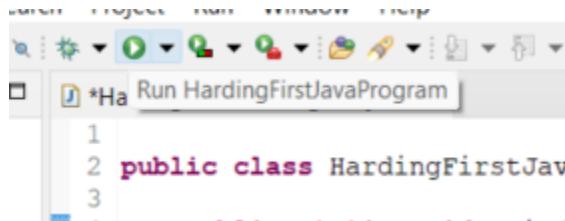
```
/*
 *
 */
public class HardingHello {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("Hello");

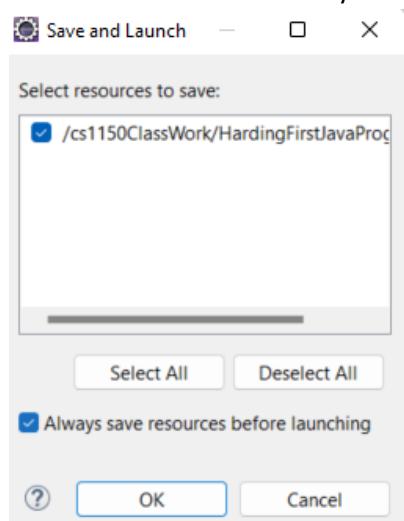
    }
}
```

```
}
```

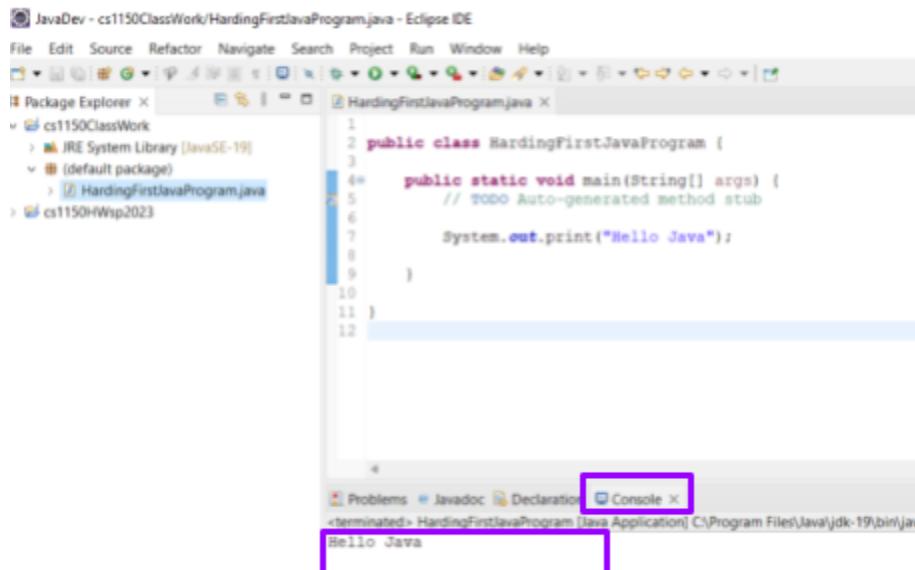
9. To run your program Select "Run" from the top menu



10. If the **Save and Launch** dialog appears, select “Always save resources before launching” so you do not need to see this every time.



11. You should see the output in the **console window** on bottom of the screen



12. Celebrate!

## Step 3: Github Desktop: Version and Backup

1. Go to your github desktop and you should see the following. This is showing your new .java file. You need to version the file locally by committing. Put

The screenshot shows the GitHub Desktop application interface. At the top, it displays the current repository as 'CS1050\_Fall2024' and the current branch as 'main'. Below this, the 'Changes' tab is selected, showing '1 changed file' named 'HardingFirstJavaProgram.java'. The code editor on the right shows the following Java code:

```
00 -0,0 +1,28 00
1 + /**
2 + *
3 + */
4 +
5 + /**
6 + *
7 + */
8 + public class HardingFirstJavaProgram {
9 +
10 +     /**
11 +      * @param args
12 +      */
13 +     public static void main(String[] args) {
14 +         // TODO Auto-generated method stub
15 +
16 +         System.out.print("Hello Java");
17 +
18 +     }
19 +
20 + }
```

A yellow box highlights the bottom-left corner of the window, which contains a modal dialog with the following content:

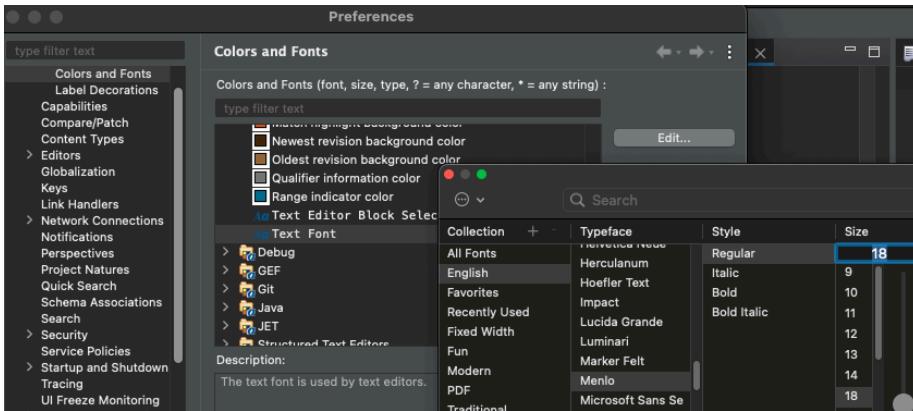
Java Hello Program  
Created a java program for GE01!  
Commit to main

2. Next you will push the update to the github server.

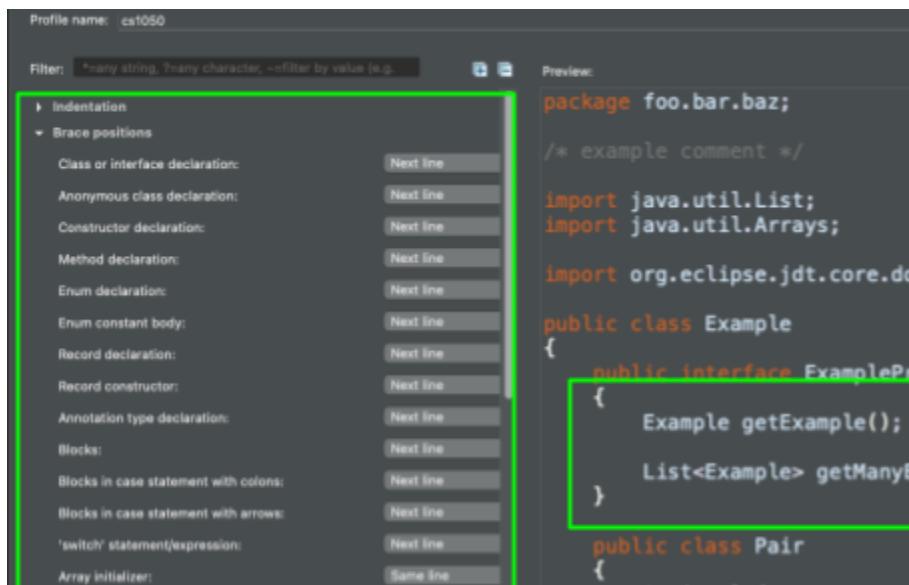


## Eclipse Resources and Tips

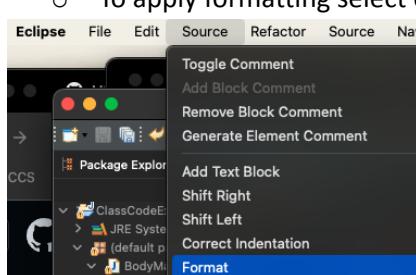
- To find information about Eclipse
  - Help -> Welcome
  - For tips and basic information click **Overview** then **Workbench Basics**
  - Click on Hide or Workbench to get back into eclipse
- Set up your theme, colors, fonts. Go to **Eclipse ->Settings -> general-> Appearance -> Colors and Fonts**. For example, you can edit the font and size for the text editor  
[6 Best Fonts for Coding to Keep Your Eyes from Eyestrain | by plabs.id](#)



- Set up formatting. Go to **Eclipse ->Settings ->**
  - [Eclipse: Changing the Java Code Style Formatting](#)
  - Note: I keep the defaults except for the brace positions to be on the next line. When first developing code a common issue is not having the open and close braces correct and this makes it easier to troubleshoot..



- To apply formatting select code and go to Source -> Format



- Use the Java EE perspective
  - A “perspective” is just a layout of views.
  - A “perspective” controls what you see in certain menus and toolbar
  - If your “perspective” gets messed up
    - Window -> Perspective -> Reset Perspective...
- If lose a view, click on **Window -> Show View**
  - Clicking in a view makes it active

- Learn to use your debugger in Eclipse. Here is a resource [Using Eclipse Debugger](#) to help you get started or search for your own resource - video, website, ai etc.

## Troubleshooting Eclipse

This section covers two possible issues that you might run into

- **Exit code 13 error**
- **Project has a red !**

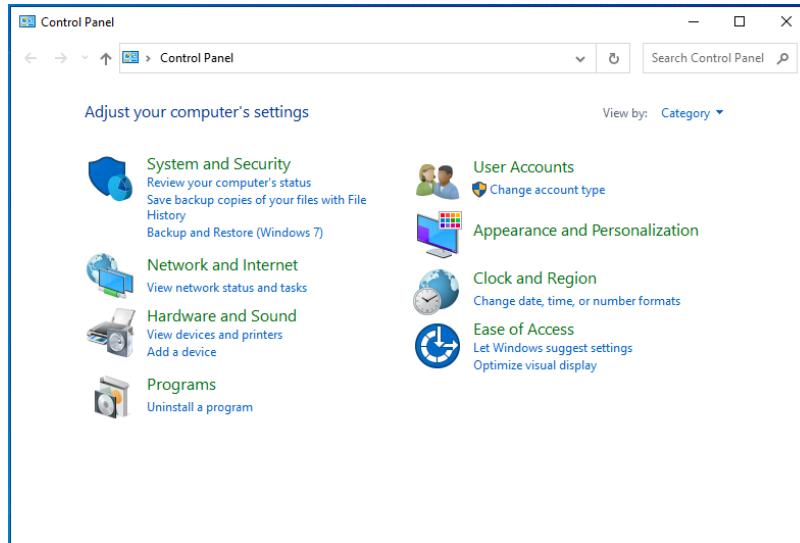
### Issue #1: Exit code 13 error

- This error occurs when there is more than one version of the JRE on your system.

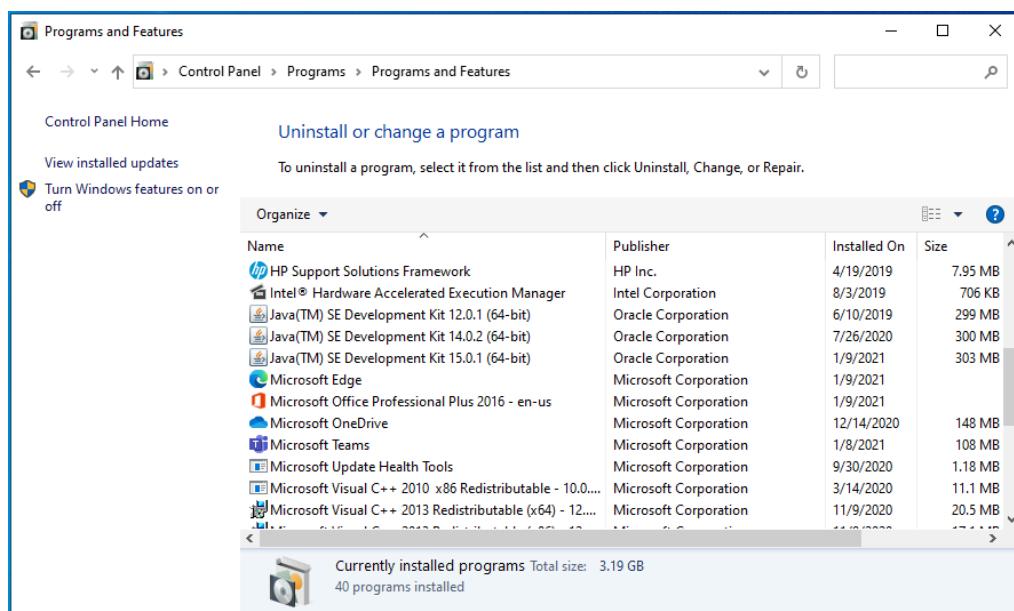


- Note that you have 2 directories on your C drive:
  - Program Files
  - Program Files (x86)
- You correctly installed the JDK into the Program Files
  - The Program Files directory is the correct location for the JDK so don't mess with it.
- Exit code 13 occurs when there is a java directory in **Program Files (X86)**

- o If you have a **Java** directory in **Program Files (x86)**, Eclipse will use it instead of the one you just installed in Program Files.
- o You need to **UNINSTALL** the java directory in **Program Files (X86)**
- o Uninstall is different from deleting, so DO NOT delete the directory.
- To fix this issue:
  - o **Uninstall** the JRE in the **Program Files (X86)** directory
  - o The new JDK that you just installed already contains the JRE.
  - o Open the Control Panel
  - o Select under Programs “Uninstall a program”



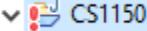
- o You next screen will appear with a list of all the programs installed on your system
  - Locate the JRE
  - I don't have the JRE on my system so the screen grab only shows the JDK's so I can't show what this looks like.
  - Once you locate the JRE, select it, then select uninstall

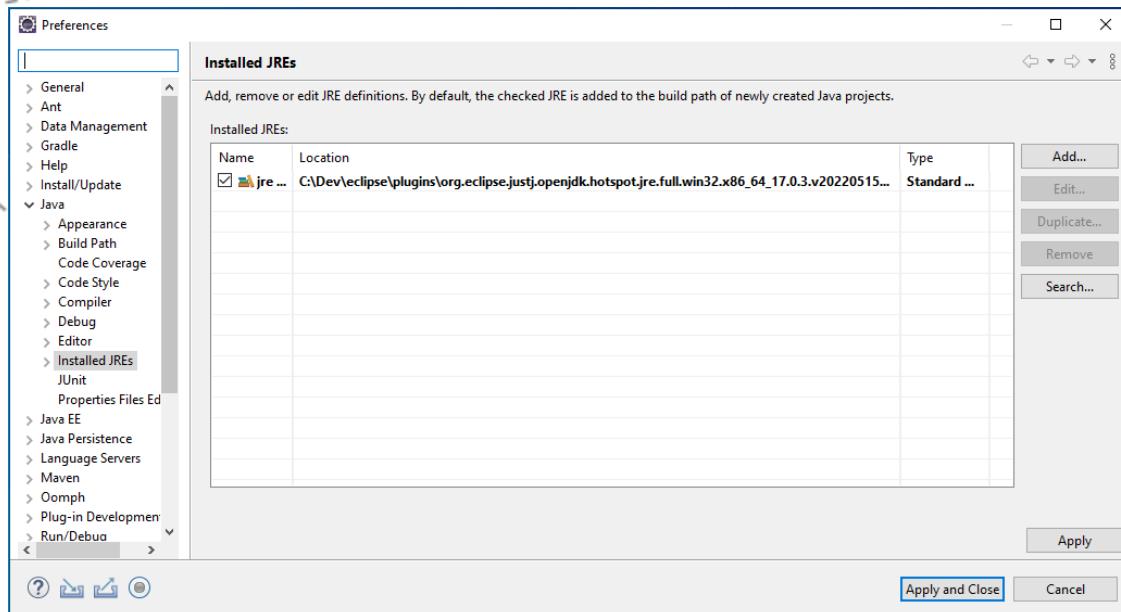


- You should now be able to start Eclipse.

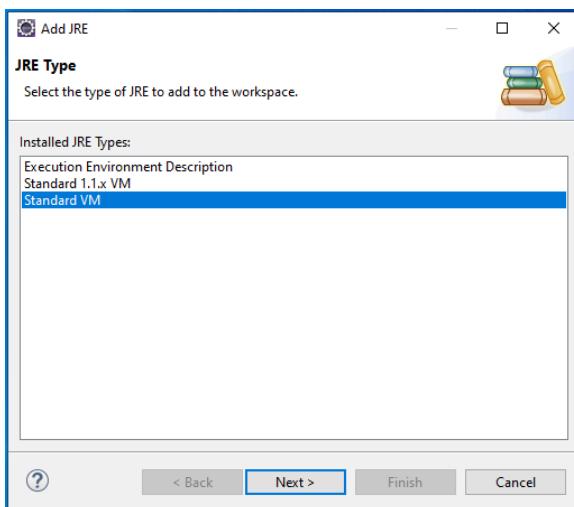


## Issue #2: Project has a red !

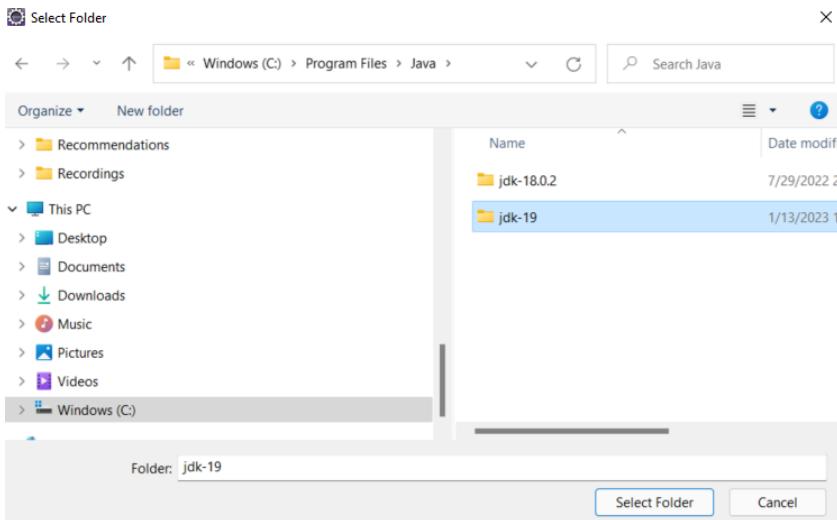
1. A project with a red ! indicates that Eclipse needs to be told about the new JDK.
  2. Eclipse will not work properly until you fix this issue!
- ▼  CS1150
3. In Eclipse, on the top menu, go to **Window -> Preferences**
  4. In the preference dialog, click arrow > next to **Java** to open Java, then select **Installed JRE's**



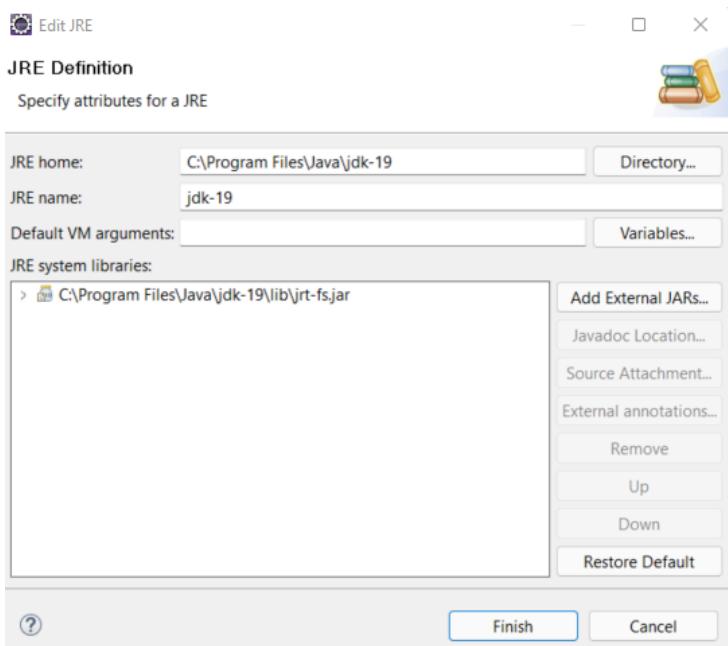
5. Click **Add**. In Add JRE window, select **Standard VM** then next



6. In the updated **Add JRE** screen, to the right of JRE Home, click **Directory...**
7. Browse to C:\Program Files\Java\jdk-22
8. Highlight jdk-22 and click **Select Folder**

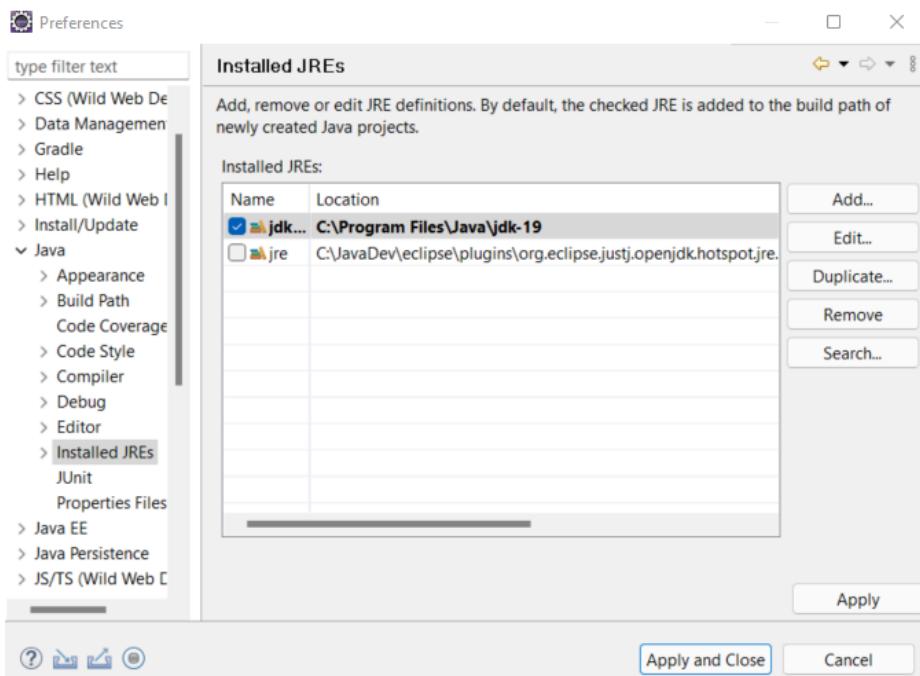


9. The Add JRE dialog will now be filled in, click **Finish**



10. You will now see in Installed JREs dialog a new entry with **jdk-22**

11. Select the box to the left of the jdk-22



12. Click **Apply and Close** button

13. The CS1150HW project should no longer have the red ! Return to [Create a Java Class](#) If you still have an issue, send me an email.

