Installing/Configuring Java & Eclipse

If at any point during the installation/set-up process you are having difficulty, please post on Piazza. For something like this, we strongly encourage you to post publicly. Often times, an install problem that you are having is a problem another student might be having as well.

Part 1: Install Java

- In order to use Java, you need to first install the **Java Development Kit (JDK)**
  - This is the package of tools for *developing* Java-based software
- You'll also need the **Java Runtime Environment (JRE)** which includes **the Java Virtual Machine (JVM)**
  - This is the environment for *running* Java applications
    - The **JVM** is what actually runs compiled Java bytecode
- Download and install the **JDK**, which includes the **JRE**: https://www.oracle.com/java/technologies/downloads/
  - Download the latest version of the JDK for your OS

Part 2: Install Eclipse

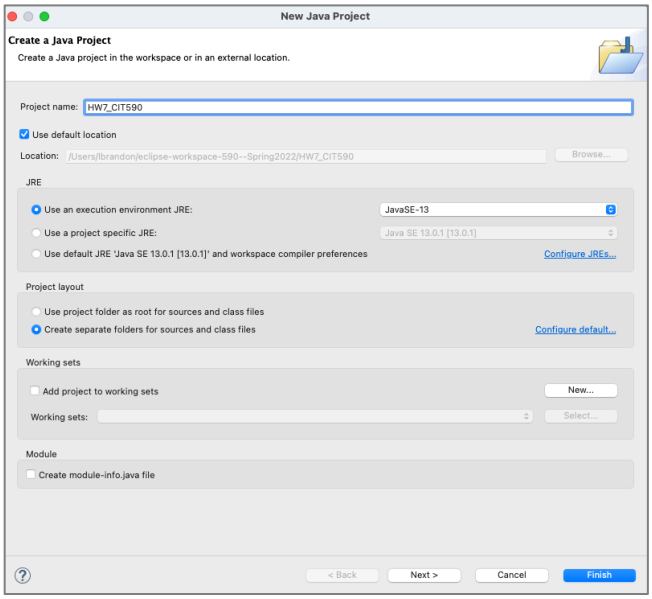
- Install Eclipse via https://www.eclipse.org/downloads/
  - Scroll down to locate and download the latest version of Eclipse.
  - Clicking on the link will take you to a final screen where you can download the actual file for installation.
- Once the file has finished downloading, extract the compressed files with the default software on your computer. This will probably happen automatically if you double click the downloaded file.
- Run the Eclipse Installer by double-clicking it or right-clicking and choosing "Open".
- You will be asked what you want to install. Choose "Eclipse IDE for Java Developers".
- Once the installation is complete, launch Eclipse.

- Please pick the default workspace option (unless you have a really strong need to change it and know what you're doing).
- If necessary, close the welcome screen.

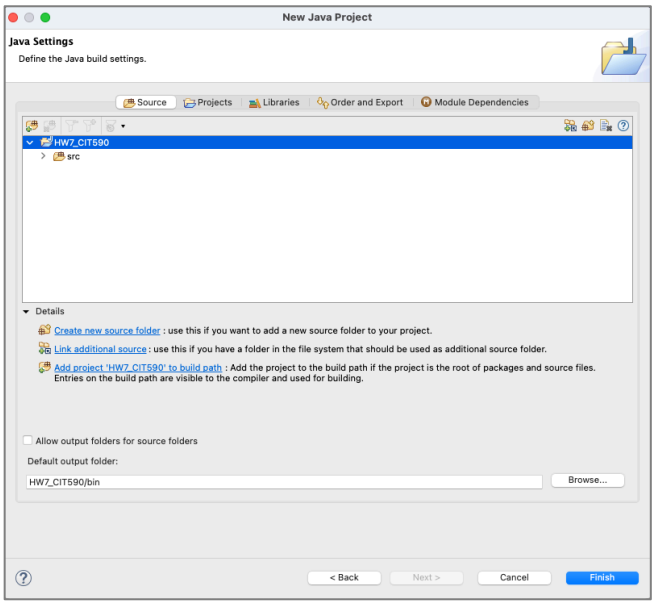
Part 3: Create a Project

If you closed Eclipse after Part 2, re-open it and if necessary, close the welcome screen.

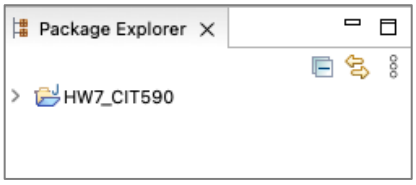
- Create a new project: File → New → Java Project
- For example, name the project "HW7_CIT590"
- Use the default output folder. Do not edit any of the other project settings in the New Java Project pop-up window -- confirm all of the options match below.



- Uncheck "Create module-info.java file"
- Click Next

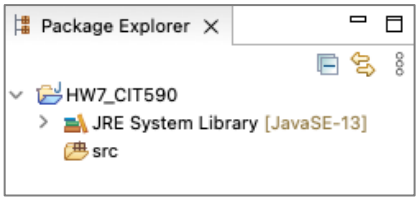


- Click Finish
- The project will appear in the Package Explorer on the left hand side:

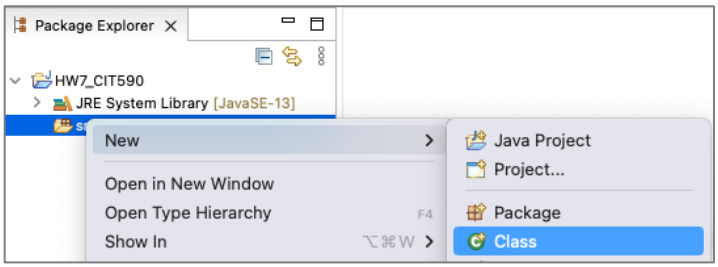


Part 4: Create a Class

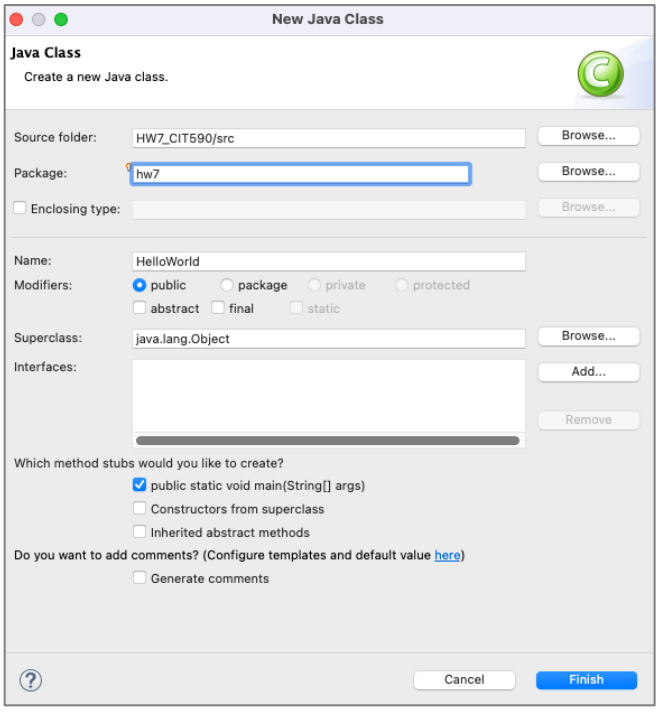
- Click the arrow on the left of the project name to open its contents.



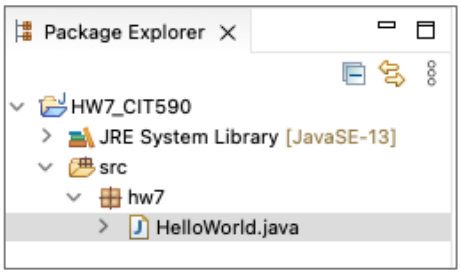
- Right click on the src folder. “src” is short for source.
- Select New → Class



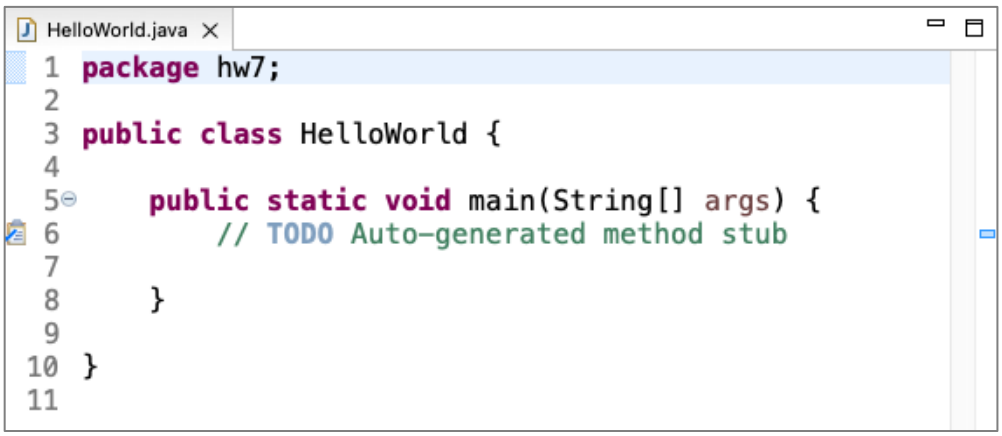
- Create a new Class using the New Java Class pop-up window.
    - For example, name the class “HelloWorld”
    - For example, name the package “hw7”
        - Please enter the class name and package exactly as we have written them. If you change the capitalization or spelling, you will lose points.
    - Check the box that says “public static void main(String[] args)”
    - Uncheck the box that says “Inherited abstract methods”, if it is checked.
    - Confirm all of the options match below.



    - Click Finish.
- Now, the Package Explorer should look like this:



- And there should be a file open, ready to edit, that looks like this:



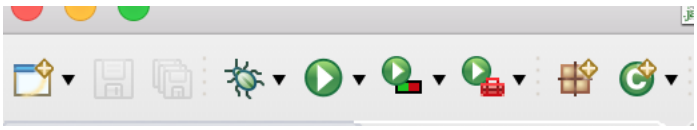
Part 5: Writing Code in Java

- Inside the *main* method, remove the comment that says:
  // TODO Auto-generated method stub

- Inside the *main* method, write the following line of code:
  System.out.println("Hello, World!");

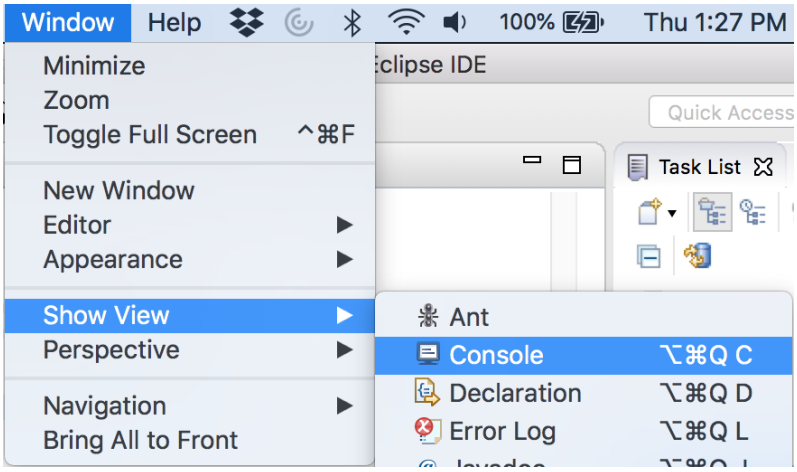
- Save the file (using the Command-S or Ctrl-S shortcut should work fine).

- In the upper left hand corner, click Run. It's the green circle with the play button.



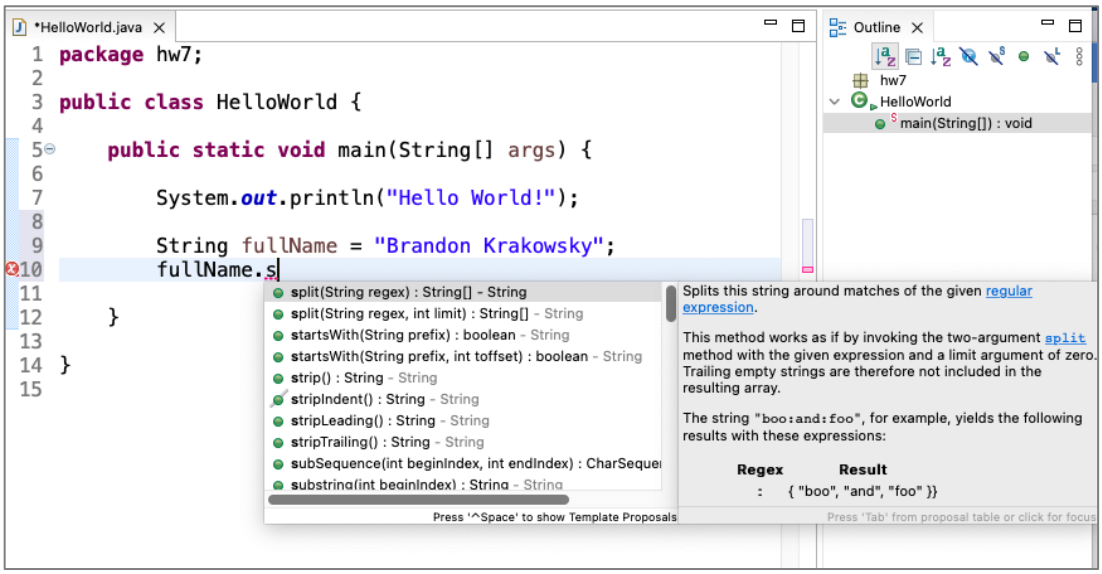
- The Console should appear in the bottom panel and Hello, World! should be printed there.

  - If you don't see the console, go to Window → Show View → Console



Getting Help

For some of the code, you may need to look up documentation. The best place to start is in Eclipse itself. If you're coding with a particular type of Object, you can start typing your code and utilize code assist to look up method documentation.



You can also reference the online Java API Specification. For example, here's the documentation for the String class: https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html