Setting up Java, Eclipse, e(fx)clipse, and the project starter code

This guide will help you set up the programming environment and starter code that you will need for this course.

Part 1: Set up Eclipse (and Java)

In this course we (and you) will be using the Eclipse integrated development environment (IDE) to develop and run our Java code. Eclipse is a powerful, industry-grade IDE. While it might be a little confusing at first because of its power, once you get the hang of it, it will provide for a very pleasant programming experience and you'll wonder how you ever programmed without it!

Optional resources for learning Eclipse

If you have never used Eclipse, here are some optional resources that you might want to check out to help you get started after you have followed the set up instructions below:

- https://www.cis.upenn.edu/~matuszek/cit591-2004/Pages/starting-eclipse.html -- Basic tutorial with example of how to write "Hello World." Includes how to do JUnit test too.
- http://agile.csc.ncsu.edu/SEMaterials/tutorials/eclipse/eclipse_tutorial_3.3.html -- Explanation of Eclipse and things that it has. Instruction on how to create project, Package, Interface, Class, etc.

Step 1: Install the Java JDK, if you don't have it already installed

Unless you have installed Java since September 2015, you must do this step to upgrade your JDK. We are using very new features of Java in this project.

1. Go to http://www.oracle.com/technetwork/java/javase/downloads/index.html

2. Click on one of the buttons to download the latest version of the Java SE JDK. These buttons are highlighted in the screenshot below (click either one).

3. Accept the license agreement in the first list of files titled “Java SE Development Kit …” and then download the file that is right for your operating system.

4. Install the JDK as appropriate for your operating system. If you need instructions, you can find them here:

- Windows: http://docs.oracle.com/javase/7/docs/webnotes/install/windows/jdk-installation-windows.html
- Mac: https://docs.oracle.com/javase/8/docs/technotes/guides/install/mac_jdk.html
- Linux: https://docs.oracle.com/javase/8/docs/technotes/guides/install/linux_jdk.html

That’s it. You’ve got the Java SDK on your machine.

Step 2: Download, “install” and setup eclipse

Next, you will install and setup eclipse. You might find the following guide helpful: https://wiki.eclipse.org/Eclipse/Installation#Download_Eclipse and/or you can follow our instructions here:

1. Go to https://www.eclipse.org/downloads/.

2. Select your operating system from the dropdown menu in the upper right. Then, in the row titled Eclipse IDE for Java Developers, click on 32 or 64-bit as appropriate to go to the download page. Make sure that the version installed is Eclipse Mars or later.

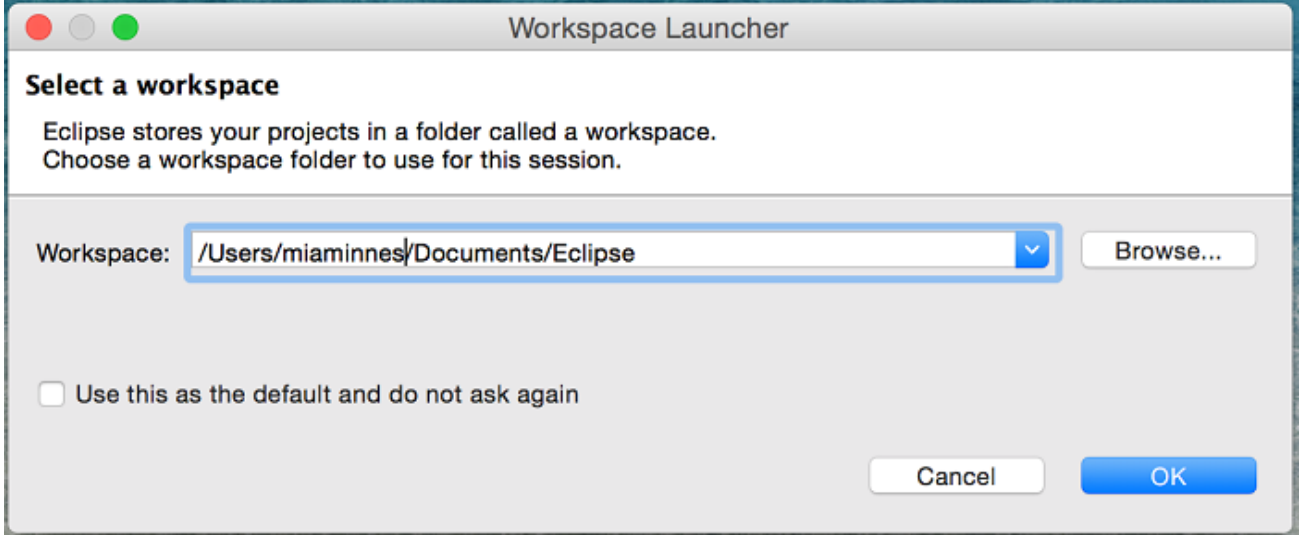
3. Click the Download button to start your download. Save the compressed (.zip or .tar.gz) file wherever you want on your computer.

4. Uncompress the downloaded file to any location on your computer. Note: Eclipse will run from here. There is nothing else you need to do to “install” it. However, Windows users should note the following warning from https://wiki.eclipse.org/Eclipse/Installation#Download_Eclipse

“Note that there is a known problem with the built-in decompression utility on all current versions of Windows. We recommend that you use a more robust decompression utility such as the open source 7zip when decompressing an Eclipse download. Some people report success when initially decompressing Eclipse into a root directory (e.g. c:\) and then moving it to a more appropriate home (e.g. c:\Program Files\Eclipse)”

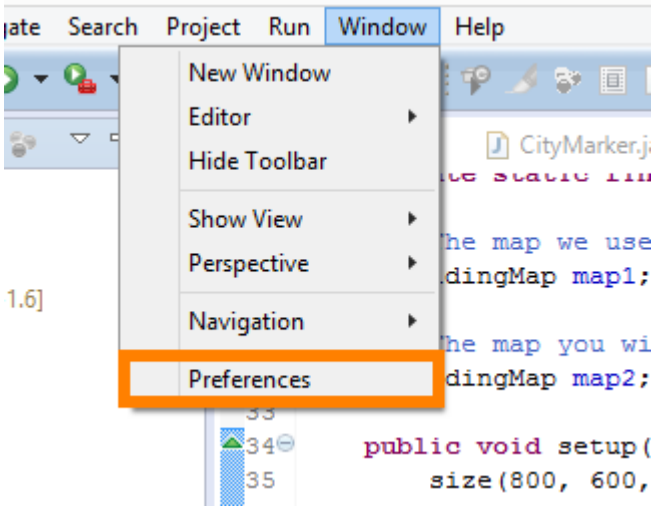
5. Once Eclipse is unzipped, run it by either double-clicking on eclipse.exe (Windows and Mac) or typing eclipse at the command line (Linux--make sure it is on your path).

6. Eclipse will ask you to select your workspace. This is where eclipse will store all of your code and project files. We recommend you choose a directory that gets backed up regularly (e.g. on Google Drive, for example). Optionally, make this the default workspace (so Eclipse will not ask you every time).

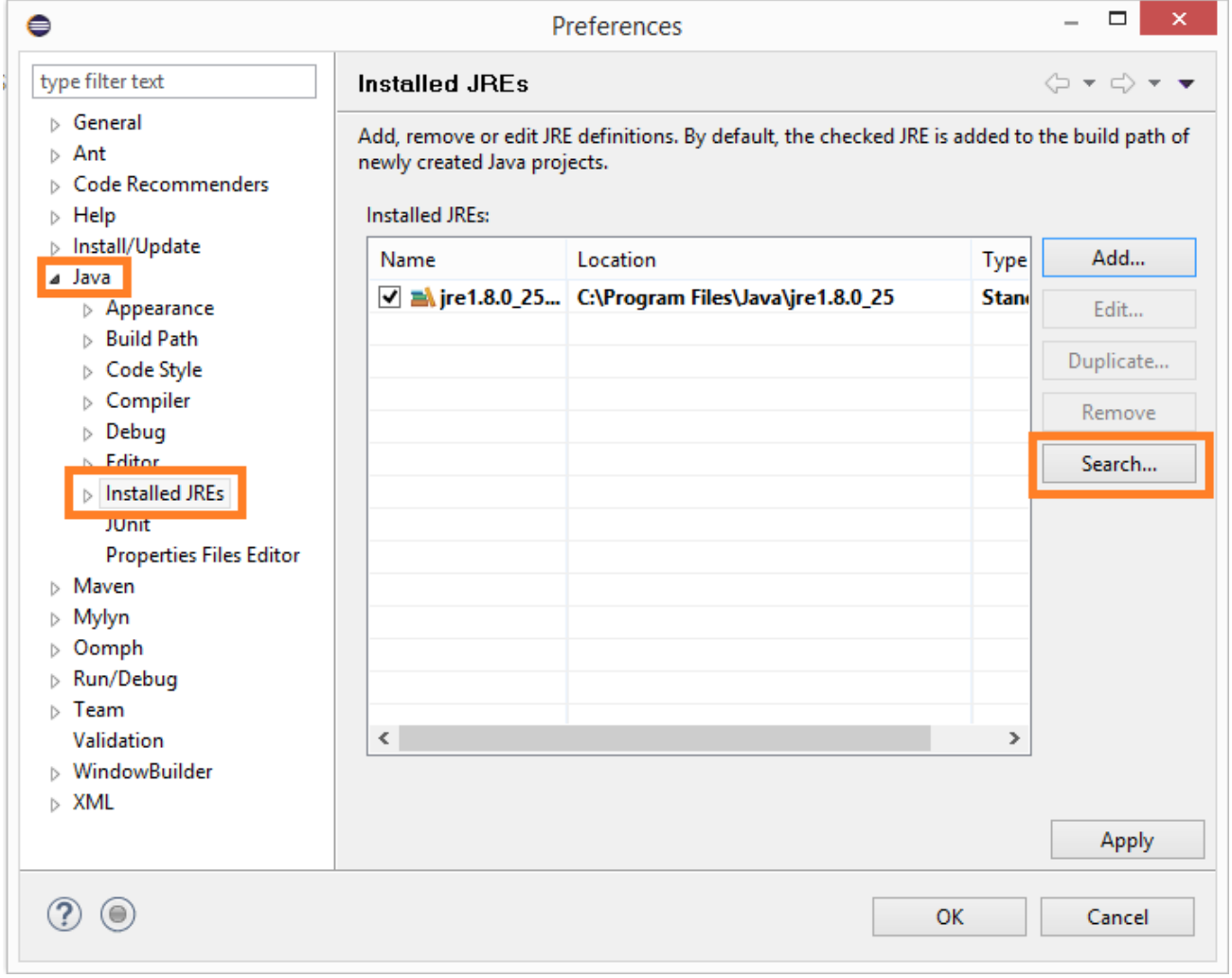


7. Then you will see a welcome screen that has links to a bunch of information including tutorials and overviews. We encourage you to try out some of these, but if you just want to dive in and get started, click the Workbench icon in the top right corner.

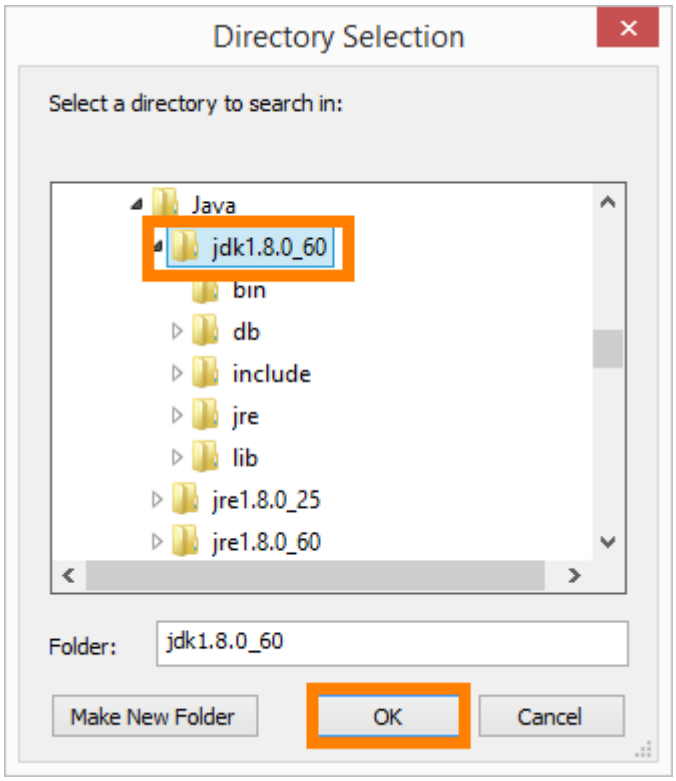
8. (**Required for Mac Users**. Otherwise Optional.) Finally, set Eclipse to use the JDK you installed in step 1. Go to Window->Preferences:



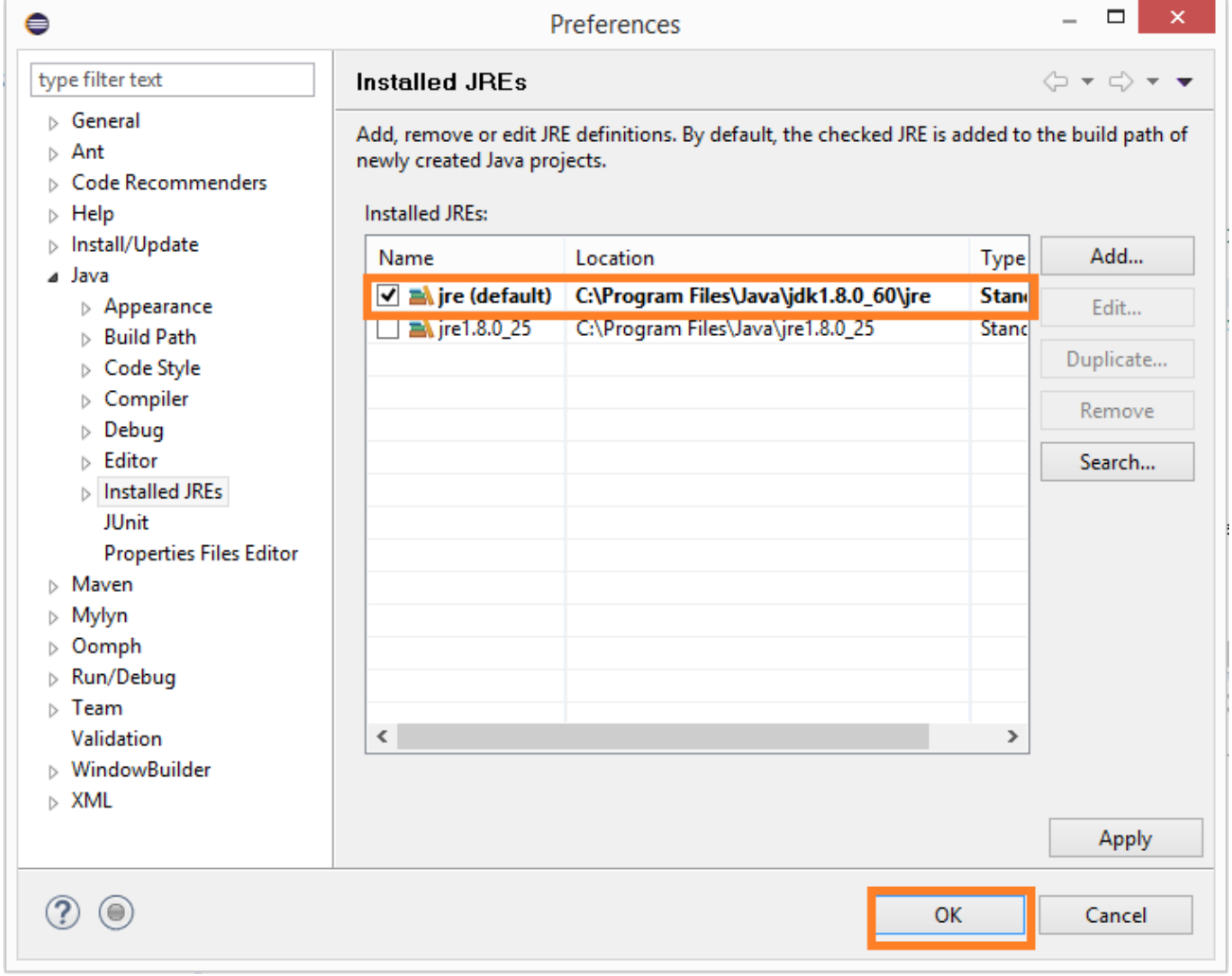
Then, in the window that opens select Java->Installed JREs in the menu on the left, then click Search:



Navigate to where you installed the JDK in step 1. Make sure you select the JDK directory and not the newly installed JRE directory! Then click OK.



After a moment, eclipse should list a second JRE in the Java->Installed JREs window. Select the JRE in the newly installed JDK folder, and click OK:

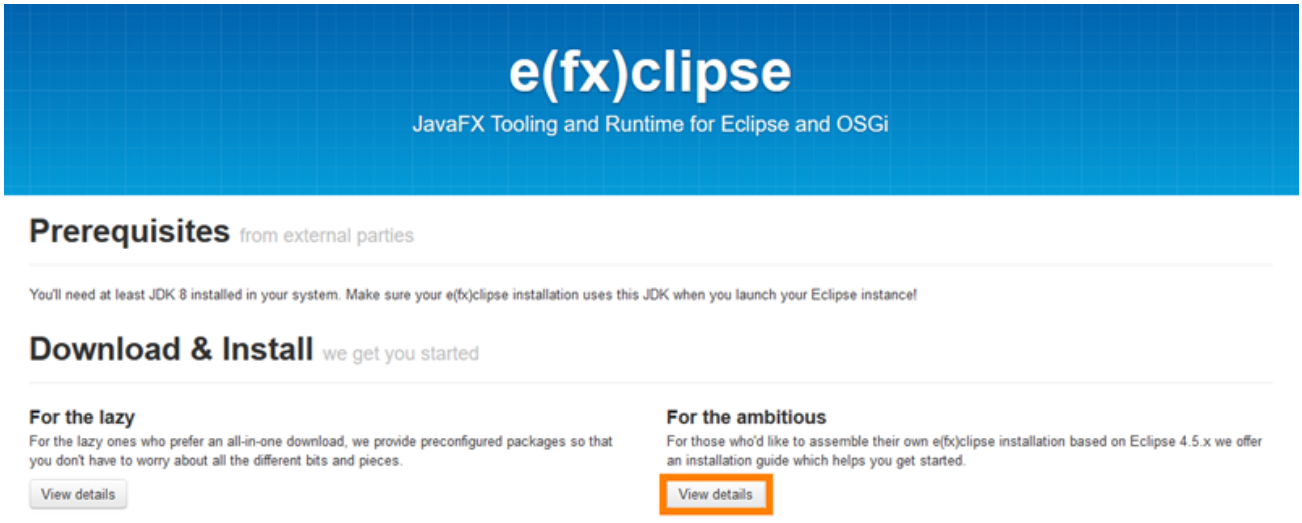


Step 3: Install e(fx)clipse

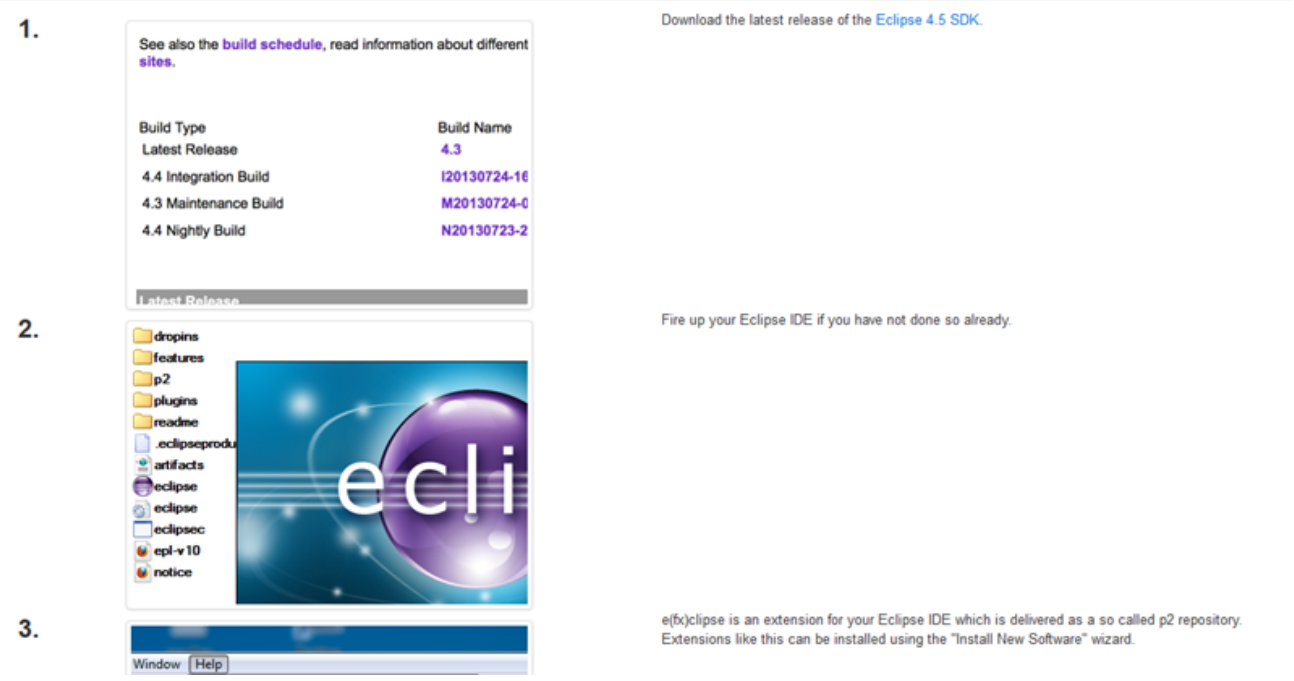
In this course the GUI you will be using is written using the Java FX libraries, and to use these libraries in Eclipse you must install the e(fx)clipse extension for Eclipse.

1. Go to www.eclipse.org/efxclipse/install.html

2. Next to "For the Ambitious" click on "View details" as highlighted in the screenshot below:



3. Follow the instructions on that page, starting at step 2 or 3, as appropriate (you've already done step 1, and probably have already done step 2 also:



Note, for step 4, after you select "Install New Software", you will want to click the "Add..." button in the upper right to add the repository.

That's it! You're ready to start working with the project code.

Part 2: Download and set up the starter code

How to make sure you always have the latest version of the starter code

Here you will download all of the code needed for the whole course. This course is a programming-heavy course with complex starter and grading code. It launched on Oct 26 and although we have tried our hardest to find and fix any errors, but unfortunately errors may still exist in the code we are providing to you. We will fix these errors as fast as possible and post the corrected code, but we will need a way to get you the updated code. Here's how that will work:

Each time we find a significant problem with the starter code we will (1) send an email to all active learners in this course and (2) upload a new zip file and annotate it with the date it was most recently published.

At the beginning of every programming assignment you should make sure nothing has changed since you downloaded the starter code. You should come back to this page to see if you have the most recent version of the zip file. If you do not, you should:

1. Download the updated zip file.
2. If the email you received specified a single file or two has changed, grab the updated file(s) from the zip file you just downloaded and copy it into your workspace directory, replacing the old version of the file.
3. Otherwise, if you are not sure what file has changed, or want to be extra careful to get all the updates: Copy any files you have created or modified out of your workspace directory. Then delete your existing project (make sure you've made copies of the files you've changed first and put them

somewhere totally different on your file system!), and follow the instructions below for setting up the project from the new zip file you downloaded. Finally, copy the files you had modified back into the appropriate locations, overwriting the starter code files. It should not be the case that we need to modify any of the files you would have changed, but if we do then we will include specific instructions in the email we send you about the changed starter code.

How to download and set up the starter code

1. Download the starter code for this course here:

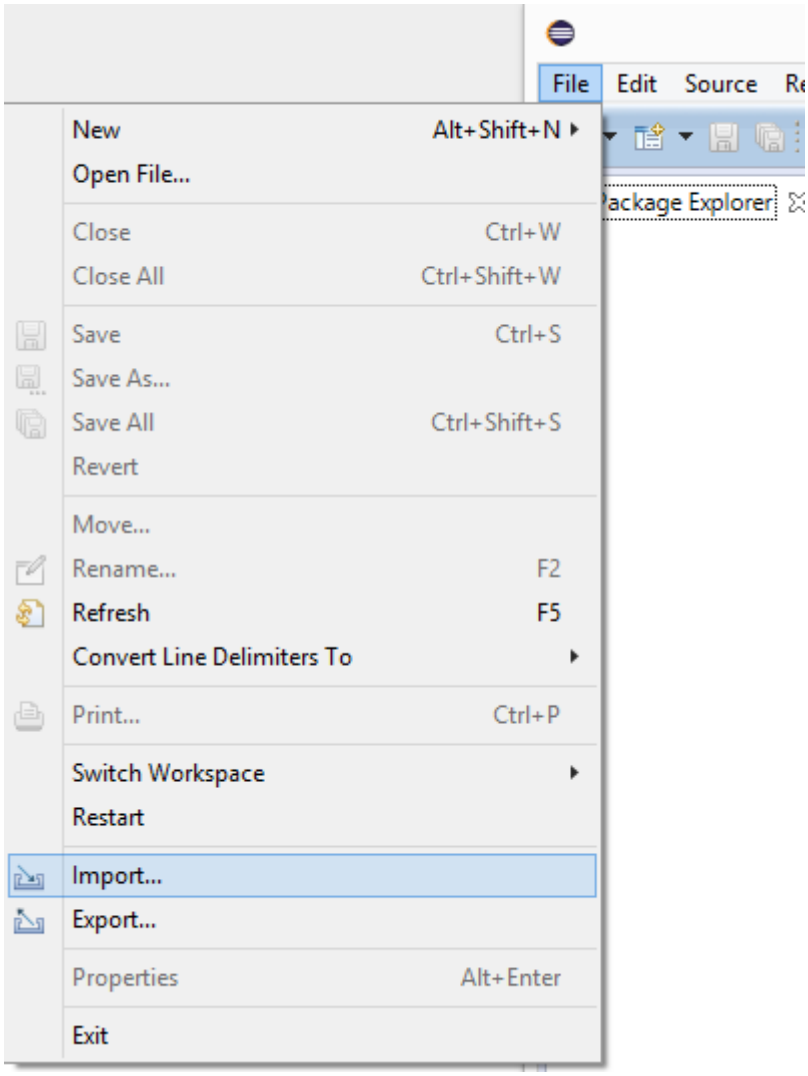
Course2StarterCode.zip

This starter code was most recently updated Jun 21, 2016 10:35am Pacific Standard Time.

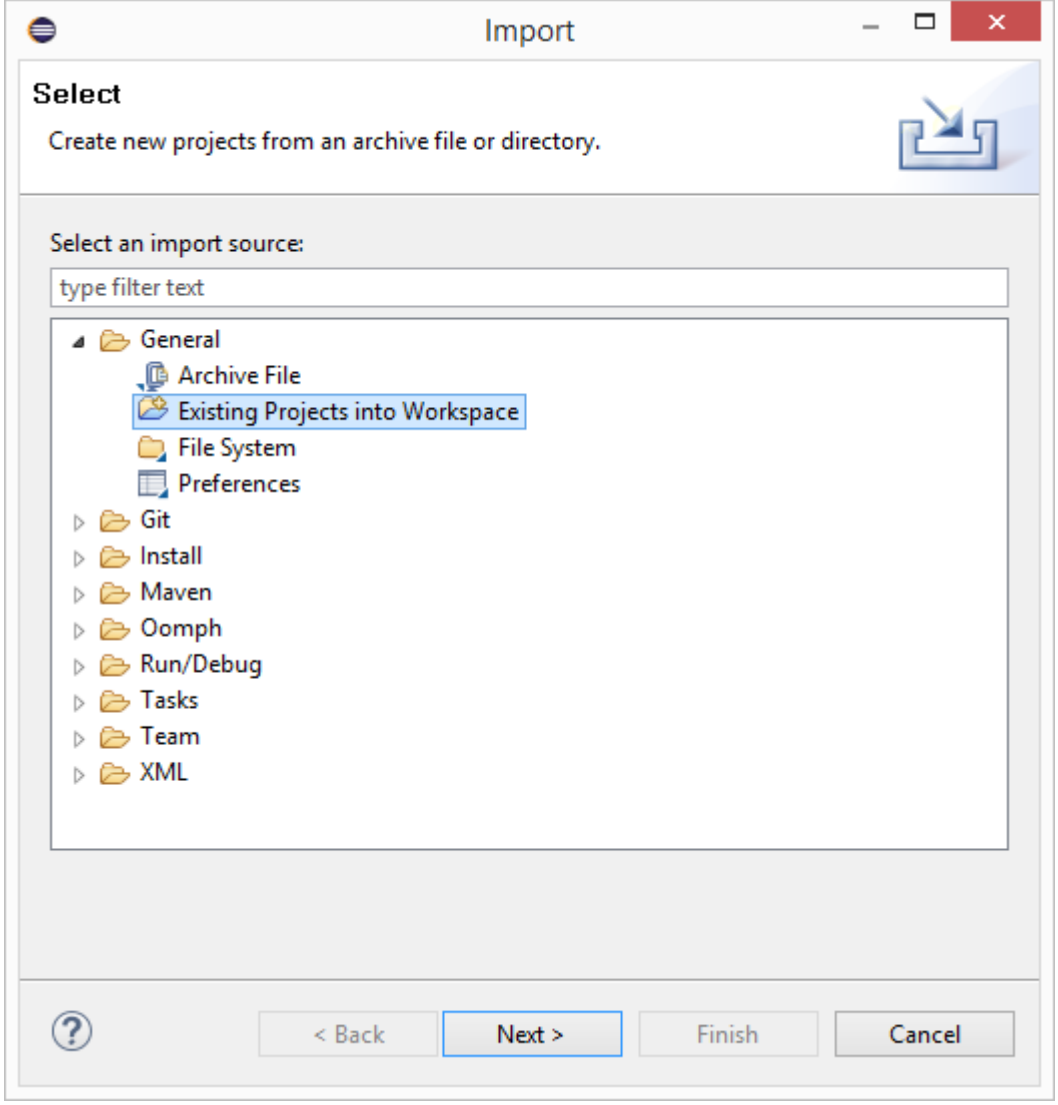
2. Extract the zip file somewhere outside of your Eclipse workspace folder.

3. Open Eclipse if it is not already open.

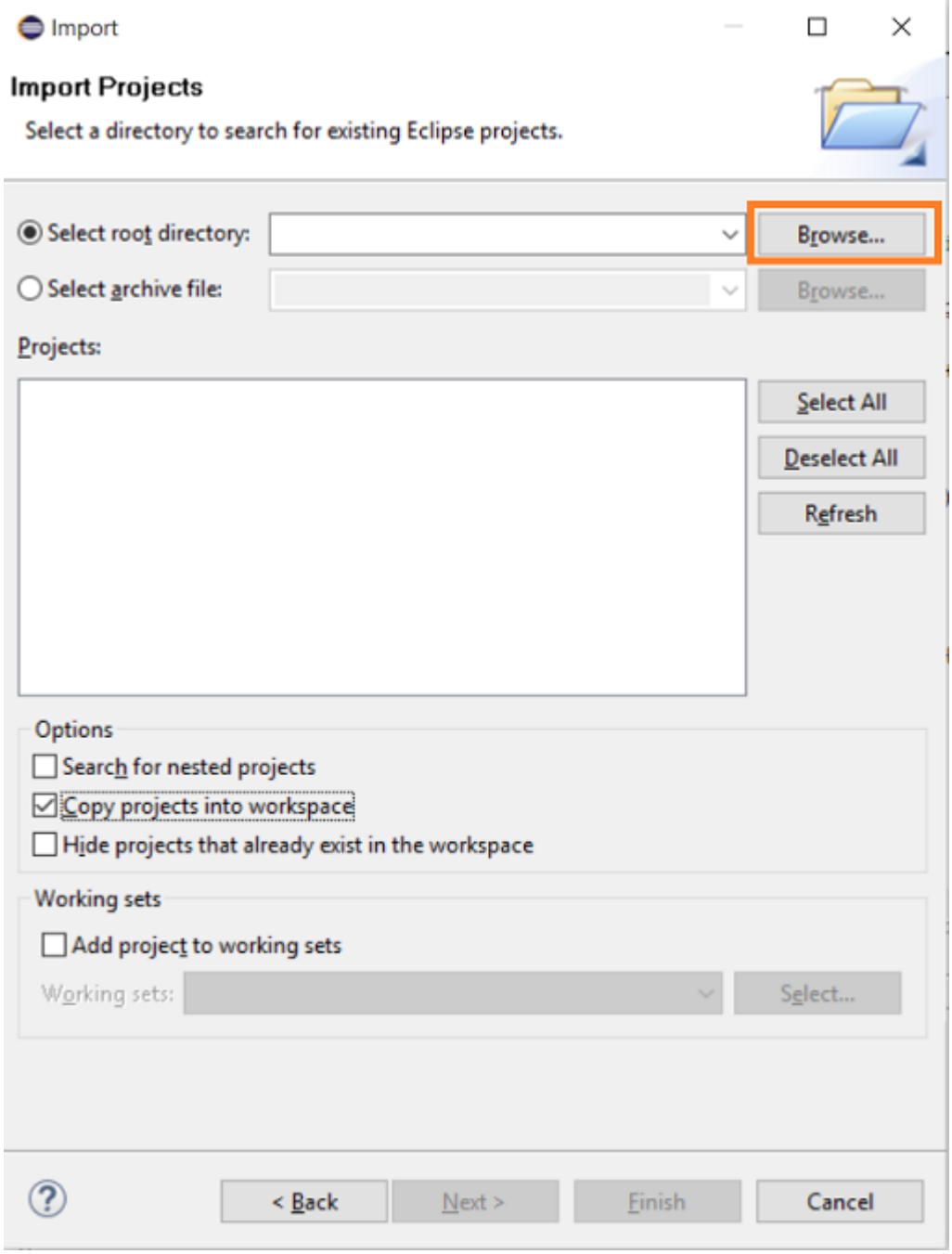
4. From the File menu in Eclipse select Import...



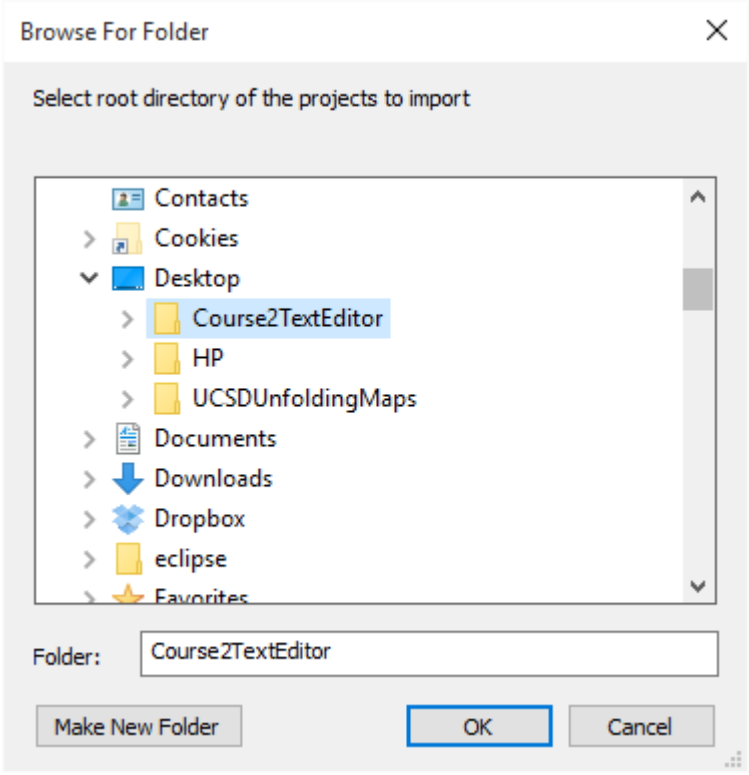
5. In the Import window expand General and select Existing Projects into Workspace. Then click Next >



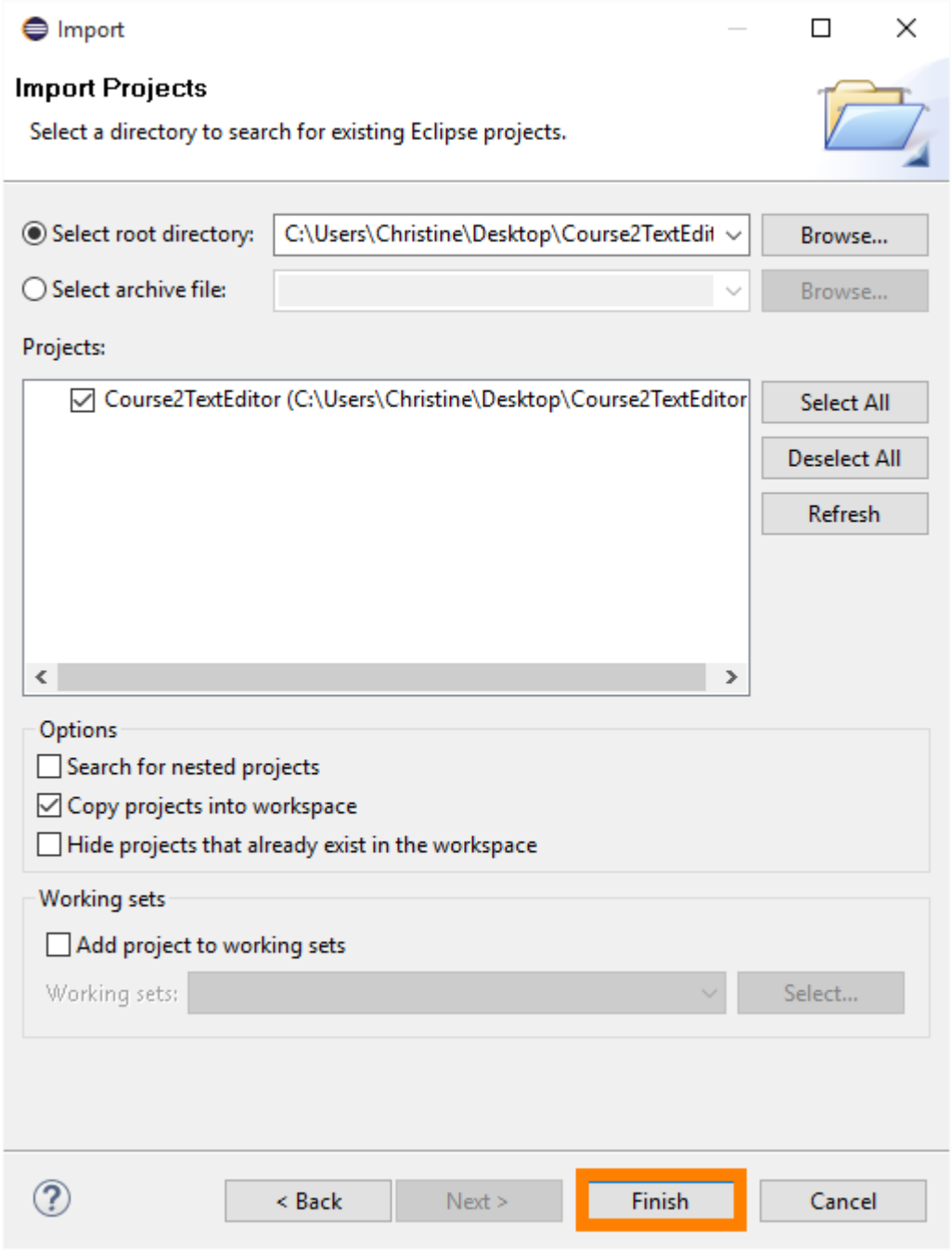
6. In the window that appears, click Browse next to "Select root directory (keep this option selected).



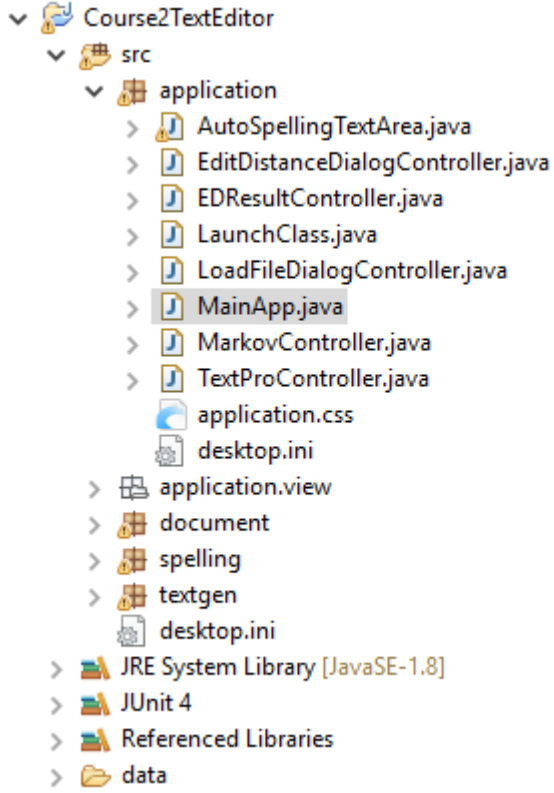
7. In the File Chooser, browse to where you uncompressed the starter code. It should be in a folder named MOOCTextEditor. Select this folder (NOT one of the subfolders) and click OK.



8. You should see the MOOCTextEditor project selected under Projects: Make sure the option to "Copy projects into workspace" is checked, then click Finish.



9. Now, you should see the project in your Eclipse workspace. In the Package Explorer you will see the MOOCTextEditor project. If you expand this project you will see several folders including src, data and some library folders. src is where all of your source (Java) code that we provide and that you will write is located. If you expand source you will see a number of "packages," which we will explain as you go through the programming assignments in the different modules. A package is just a collection of files that are organized into their own space so they won't interfere with other files of the same name in other packages. The "application" package (shown expanded in the screenshot below) contains the graphical user interface (GUI) front end for your text editor. This code is provided for you. You will be adding code to the other packages to implement the back end. The instructions for the programming assignment in this module give you more information about how to compile and run your code.



It's expected to see some warnings (the tiny yellow exclamation points next to the items in the Package Explorer) and generally you can ignore them for now, though sometimes they do give you hints about how to improve your code. You can see what they are in the window at the bottom. However you should not see any errors (which would be displayed in red).

10. **Add JUnit to your project.** To do this, go to Project->Properties. Select "Java Build Path". Select the "Libraries" tab and "Add Library". Select JUnit, then JUnit 4.

11. **Set up the workspace JRE.** Right click on the MOOCTextEditor project folder in the Package Explorer and select Build Path->Configure Build Path. Go on the Libraries tab and click on "Add Library". Select the "JRE System Library" and click Next. Select "Workspace default JRE" and click Finish. Then click OK.

12. Finally, notice that when you imported this project into Eclipse you got a COPY of all of the starter code files you downloaded into your workspace. You will find the actual files Eclipse is using (including the source (src) and data files) in a directory in your workspace folder called MOOCTextEditor. Go find these files so you know where they are and what Eclipse is doing behind the scenes.

That's it! You're all set up and ready to go.

✓ Complete