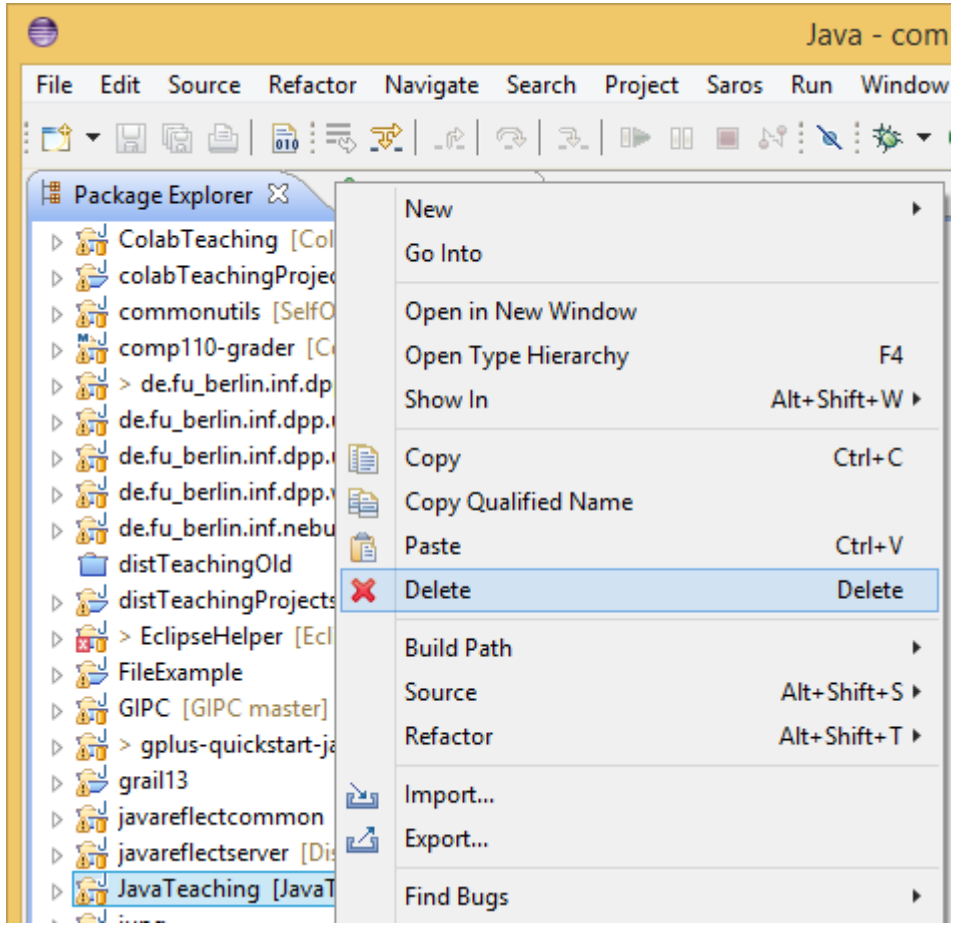
JavaTeaching and Importing a github repository

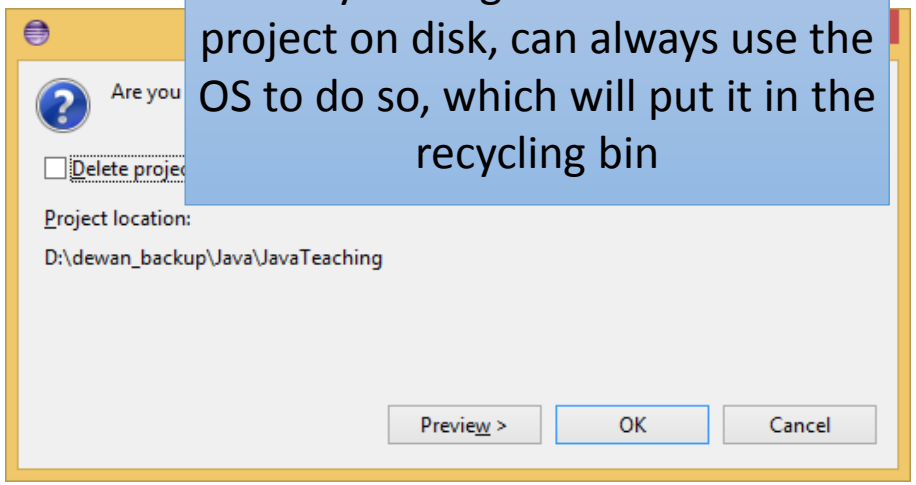
Dong Nie

Example used: JavaTeaching

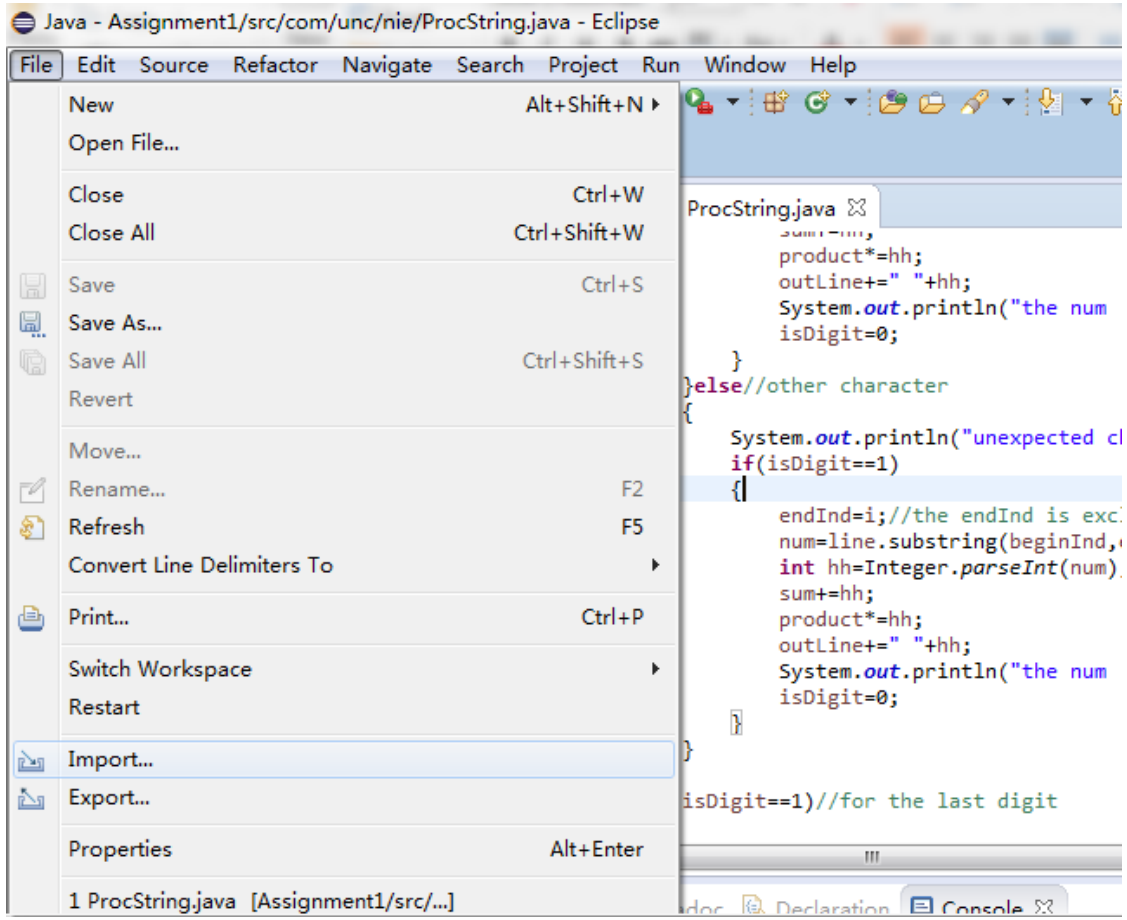


If you have already loaded JavaTeahcing from zip file, you should delete it before starting on this exercise

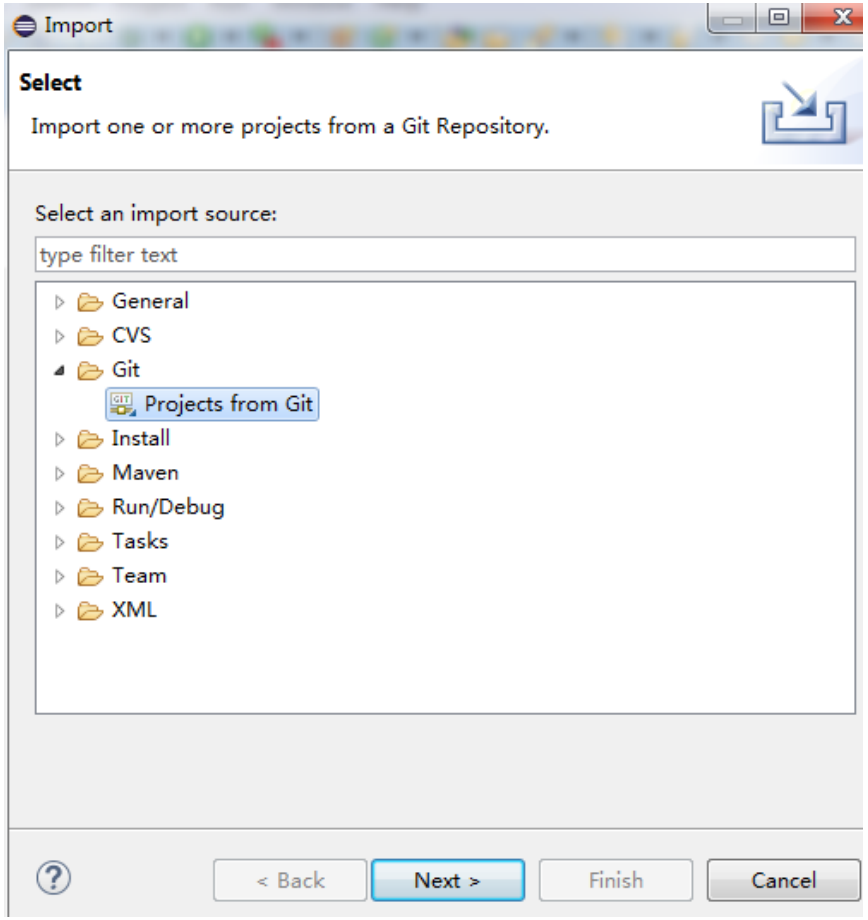
Right click project and delete. Usually not a good idea to delete project on disk, can always use the OS to do so, which will put it in the recycling bin



Eclipse: File->import

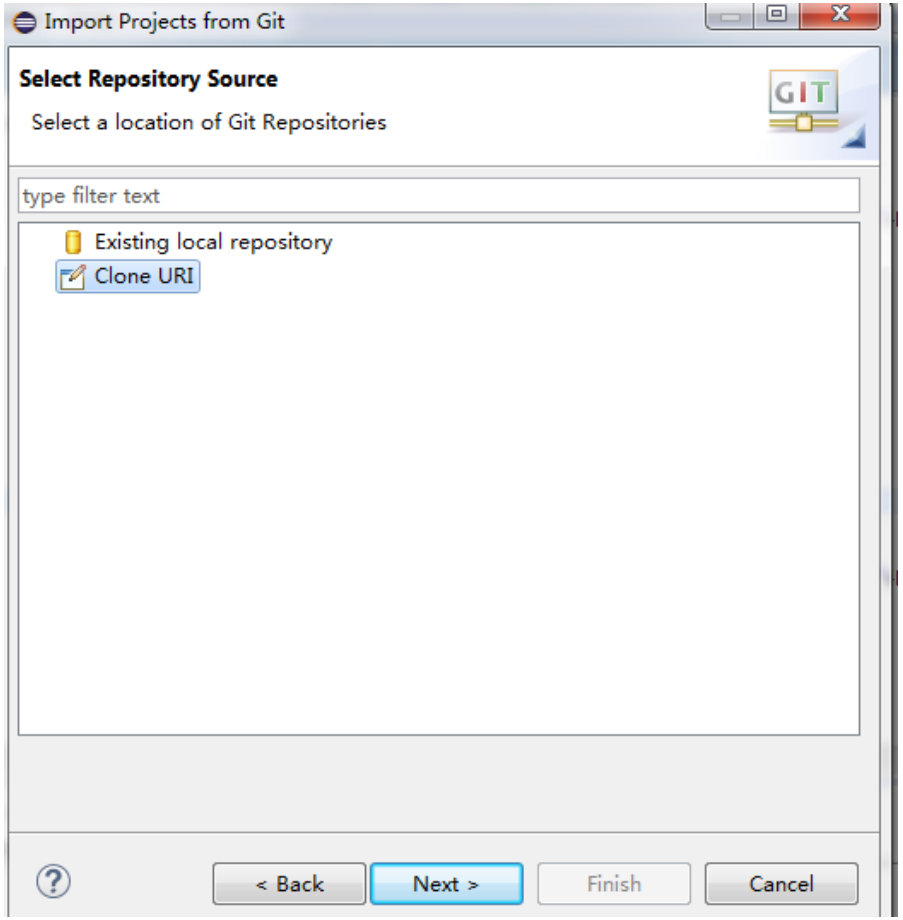


Choose Git: Projects from Git, and click “next”

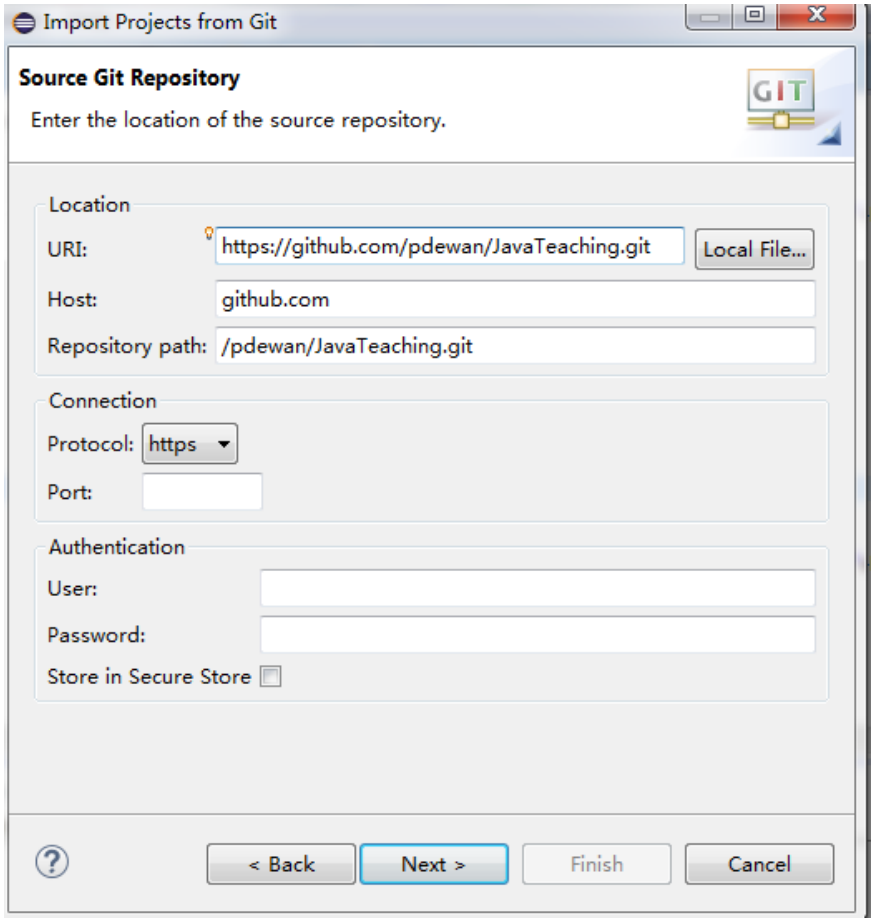


If you do not see the Git choice, see PPT on importing Egit in reference material or install most recent version of Eclipse

Choose Clone URI, and click “next”

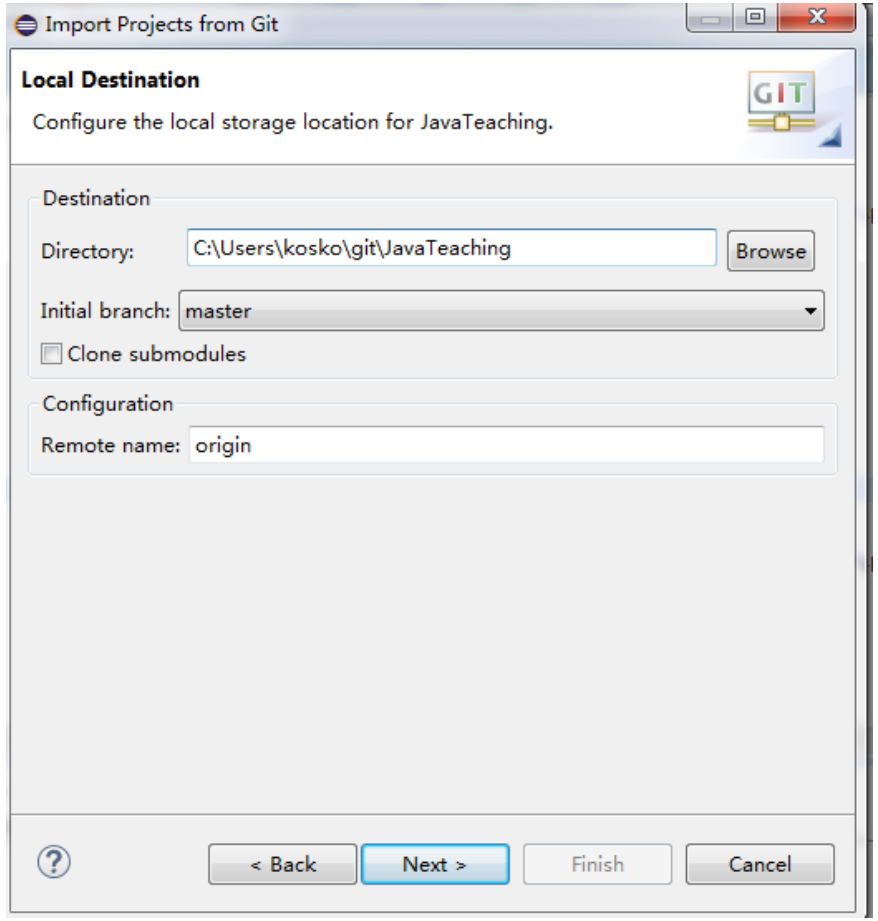


Use the url: https://github.com/pdewan/JavaTeaching.git

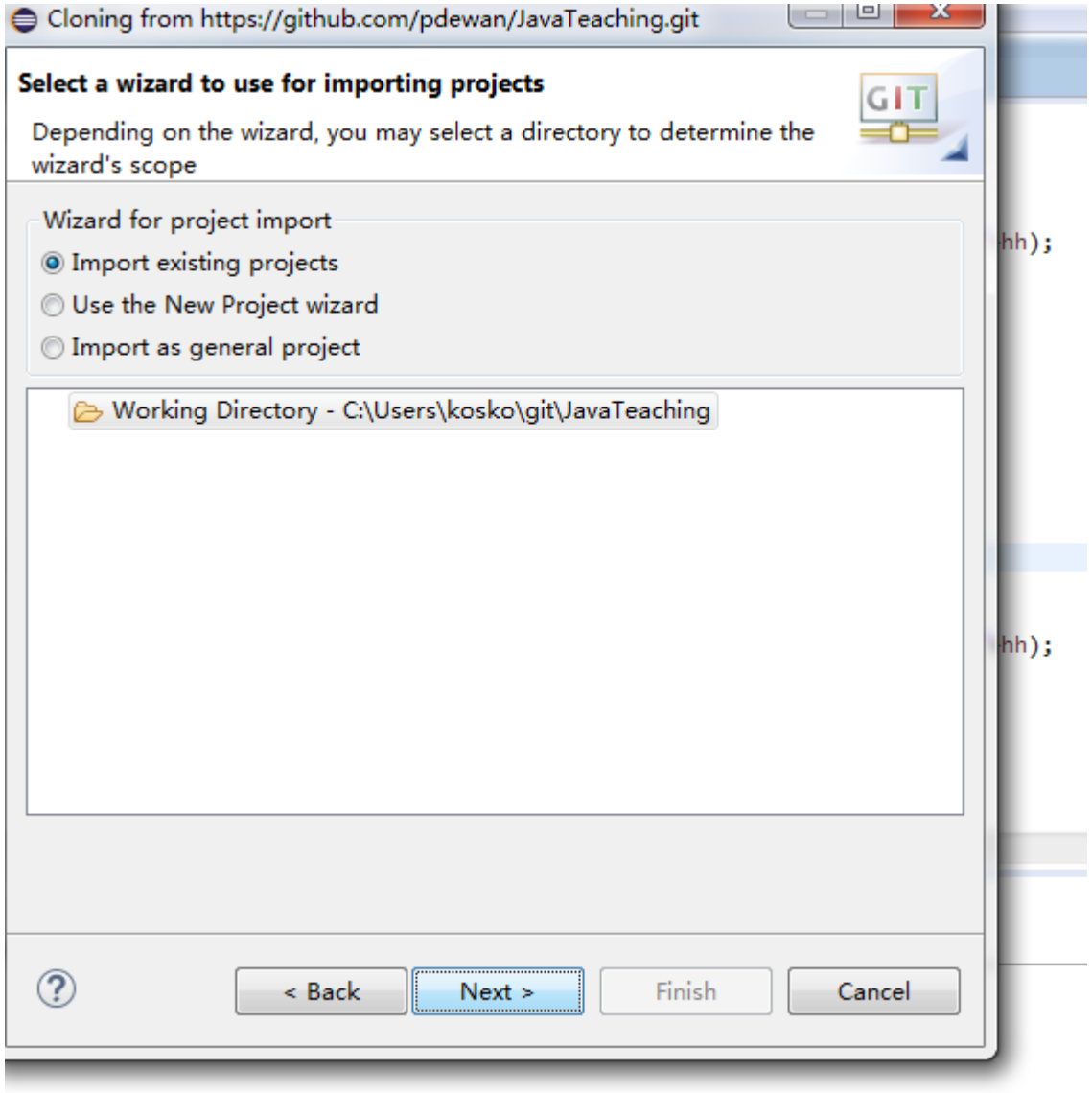


Only the URI field needs to be filled, the others will be set automatically

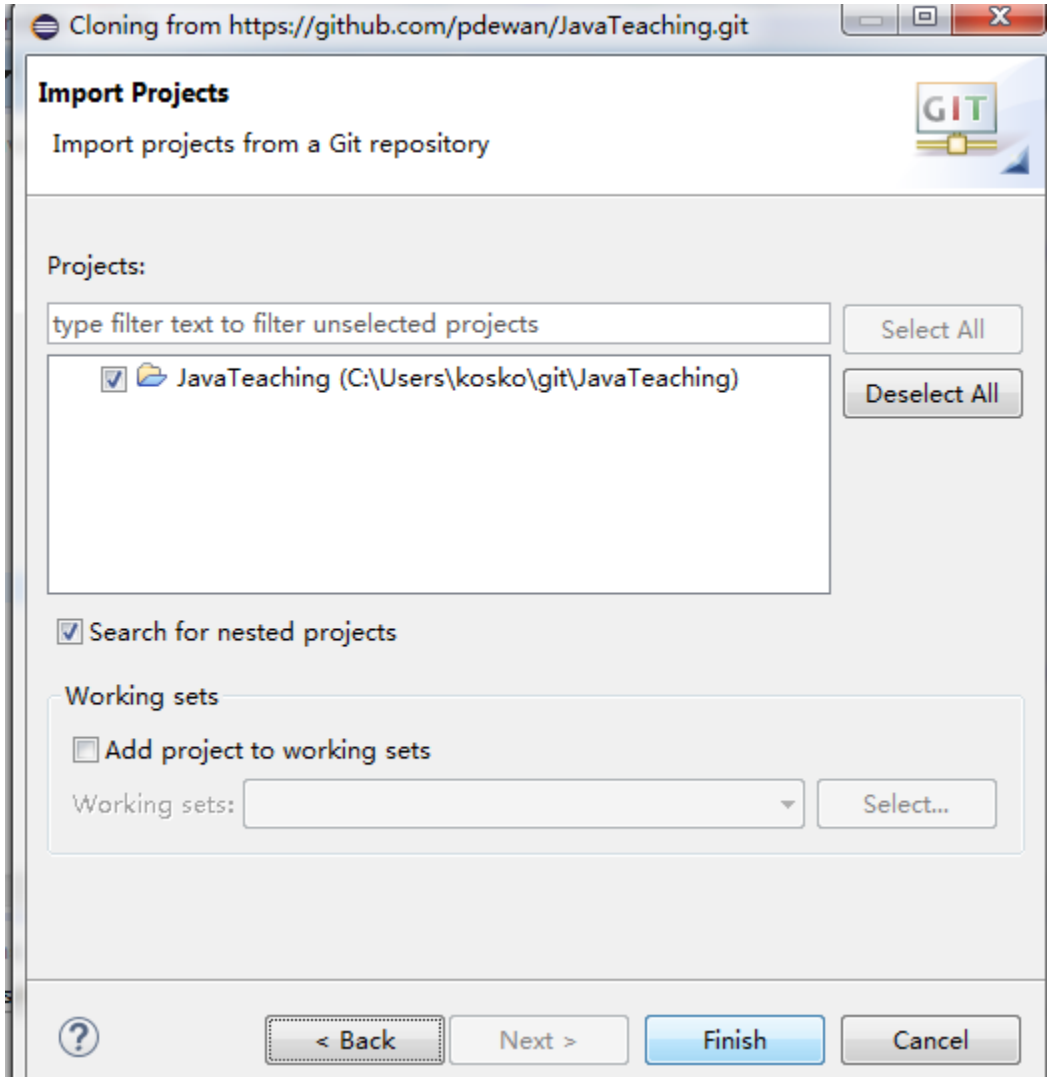
Choose Destination Directory (you can set default, and click “next”



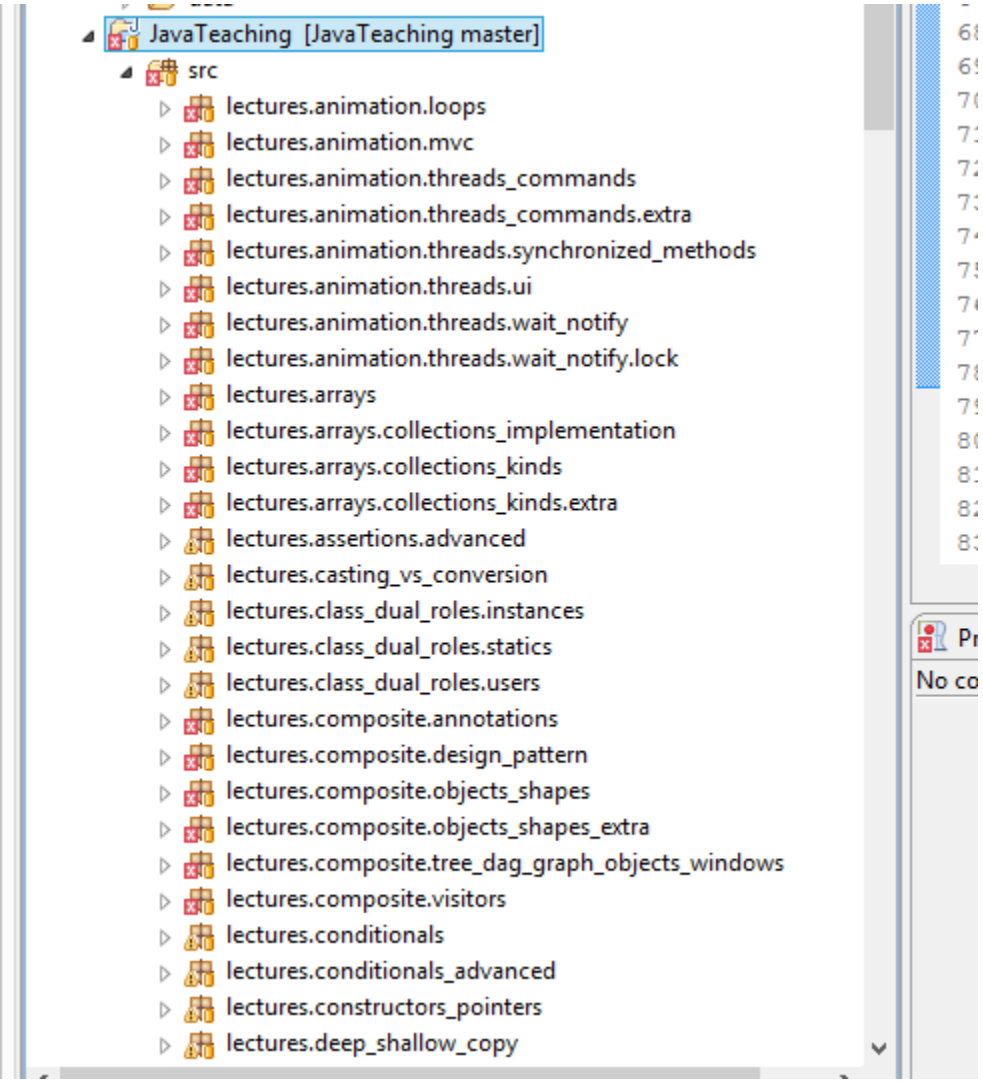
Click “next”


Cloning from https://github.com/pdewan/JavaTeaching.git
Select a wizard to use for importing projects
Depending on the wizard, you may select a directory to determine the wizard's scope
GIT
Wizard for project import
Import existing projects
Use the New Project wizard
Import as general project
Working Directory - C:\Users\kosko\git\JavaTeaching
< Back
Next >
Finish
Cancel

Finish

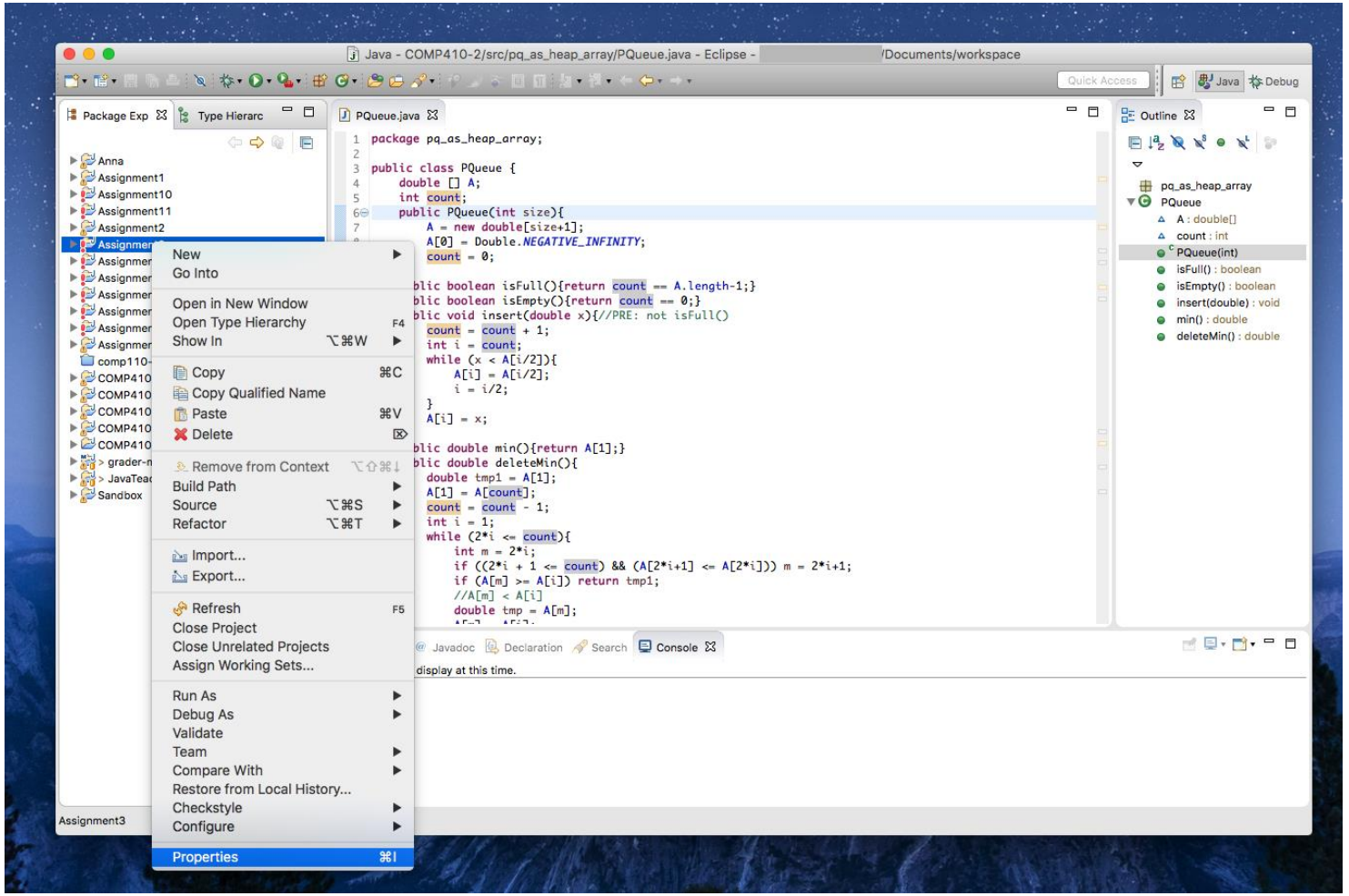

Cloning from https://github.com/pdewan/JavaTeaching.git
Import Projects
Import projects from a Git repository
GIT
Projects:
type filter text to filter unselected projects
Select All
JavaTeaching (C:\Users\kosko\git\JavaTeaching)
Deselect All
Search for nested projects
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

Here comes the Project with Compile Errors



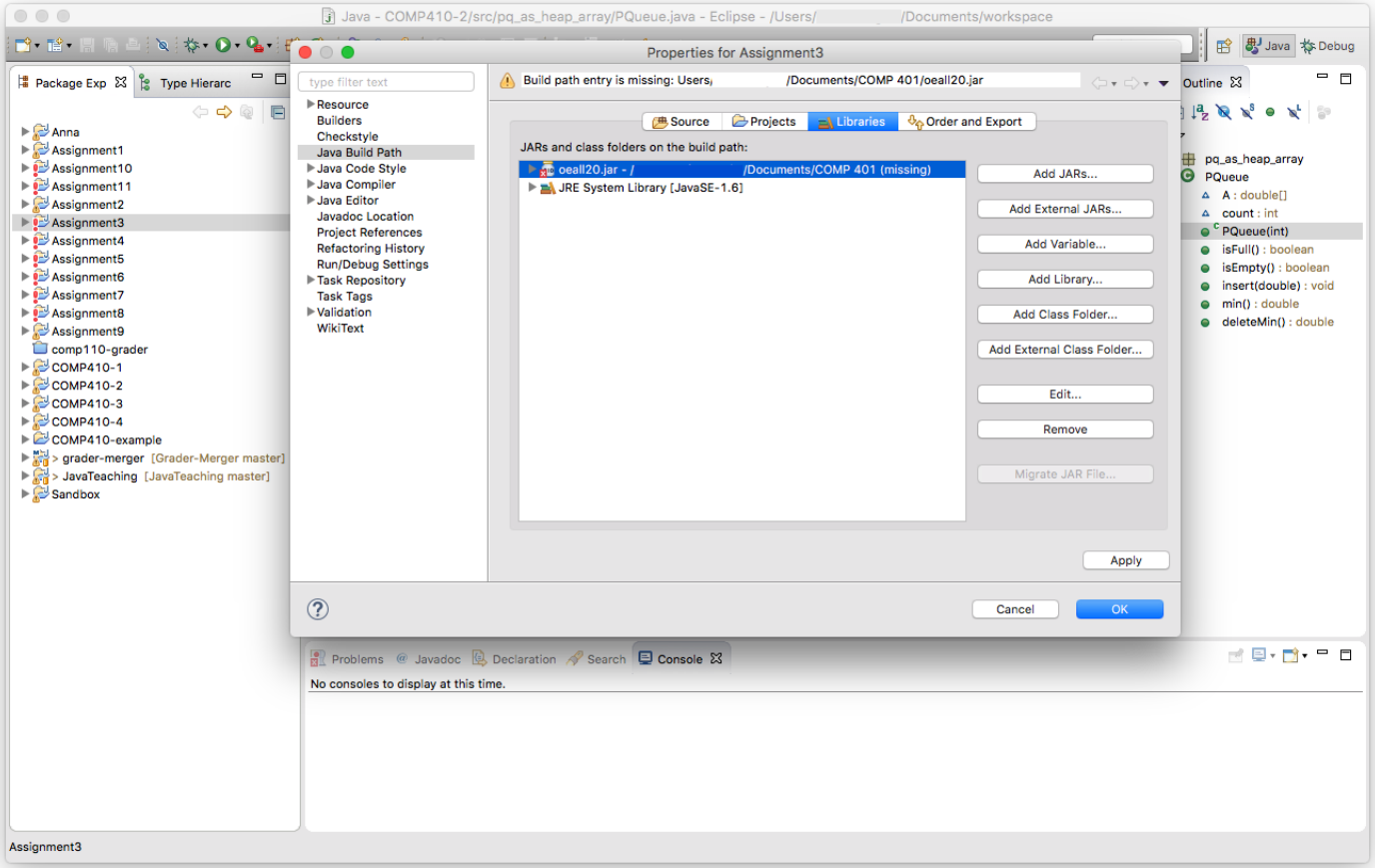
You will have errors, be sure to delete the oeall22 reference (if it exists in the class path) and add your oeall22 library (See eclipse install and objecteditor library sections and the next two slides) Until you need ObjectEditor, you can ignore the errors

Replacing/Adding oeall22 (or any other JAR)



Right-click on the project you want to modify → Properties

Replacing/Adding oeall22 (or any other JAR)

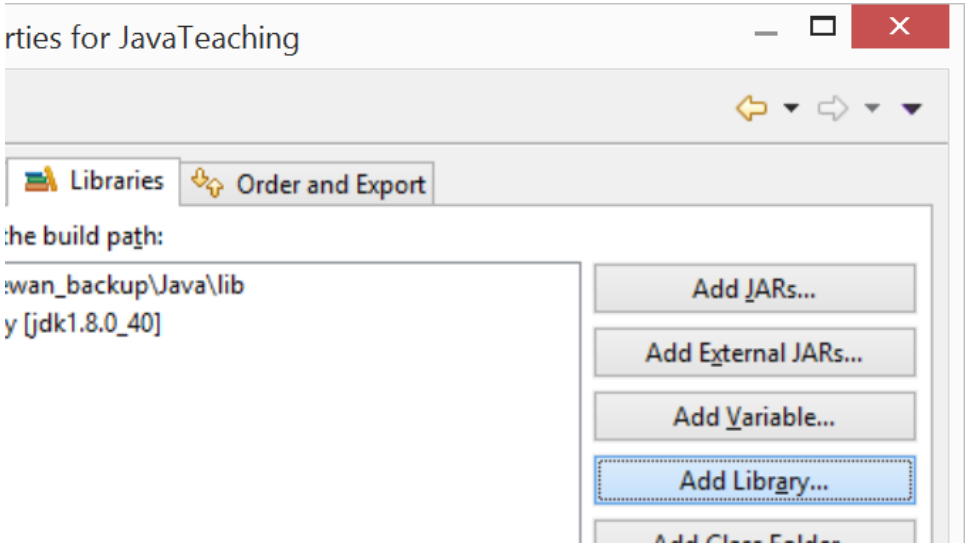


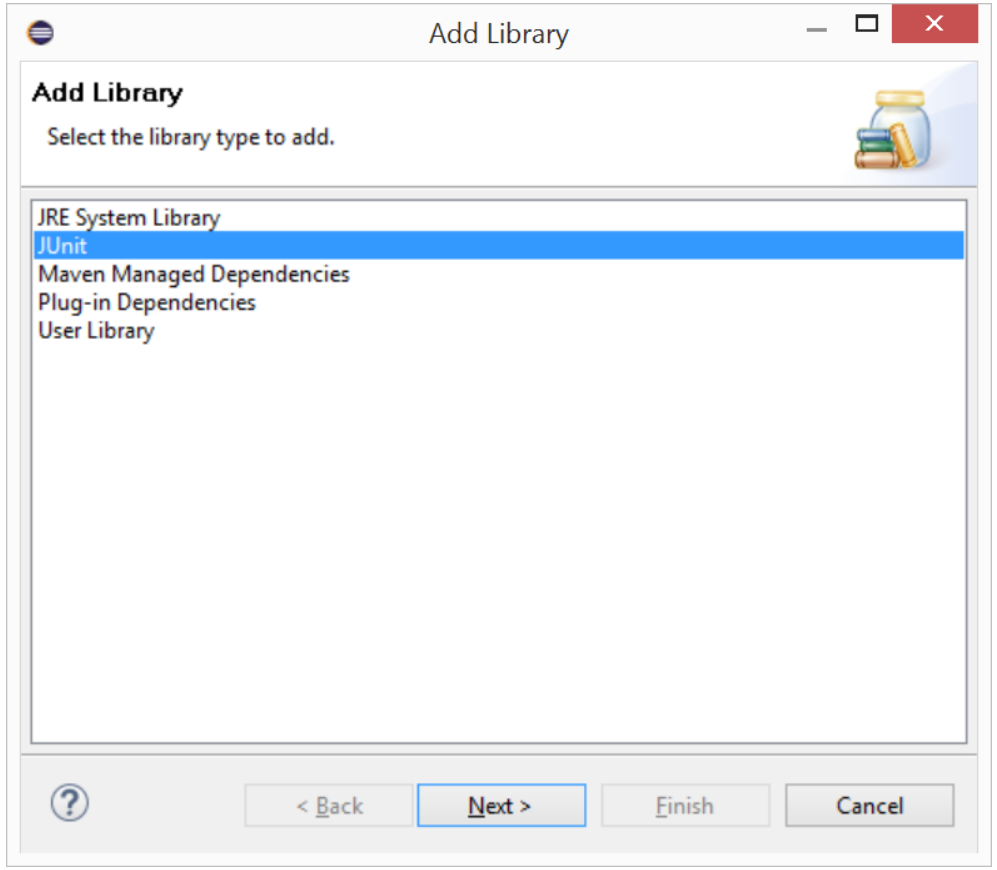
Go to Java Build Path → Libraries tab. Remove the old JAR (if it exists), and choose **Add External JARs**. Click Apply→OK

If you see modulepath or classpath as options, select classpath

You need not add external jar if you can live with the compiler errors in code you will not access in the first few praxes. However, if you have time, look at objecteditor PPT on how to download oeall from the web page and then use add external jar to add it. At this point there should be no errors.

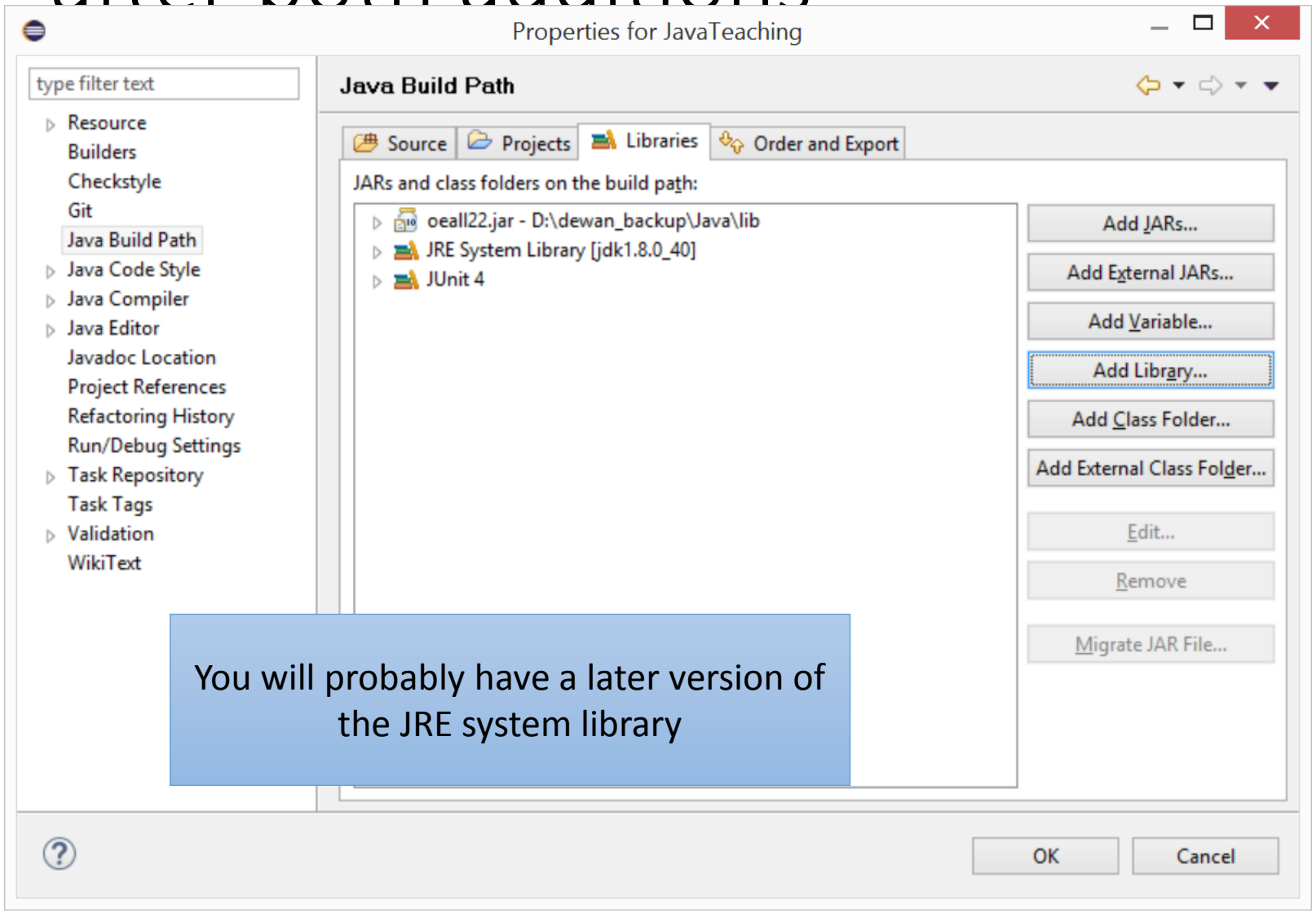
Add junit library to your classpath





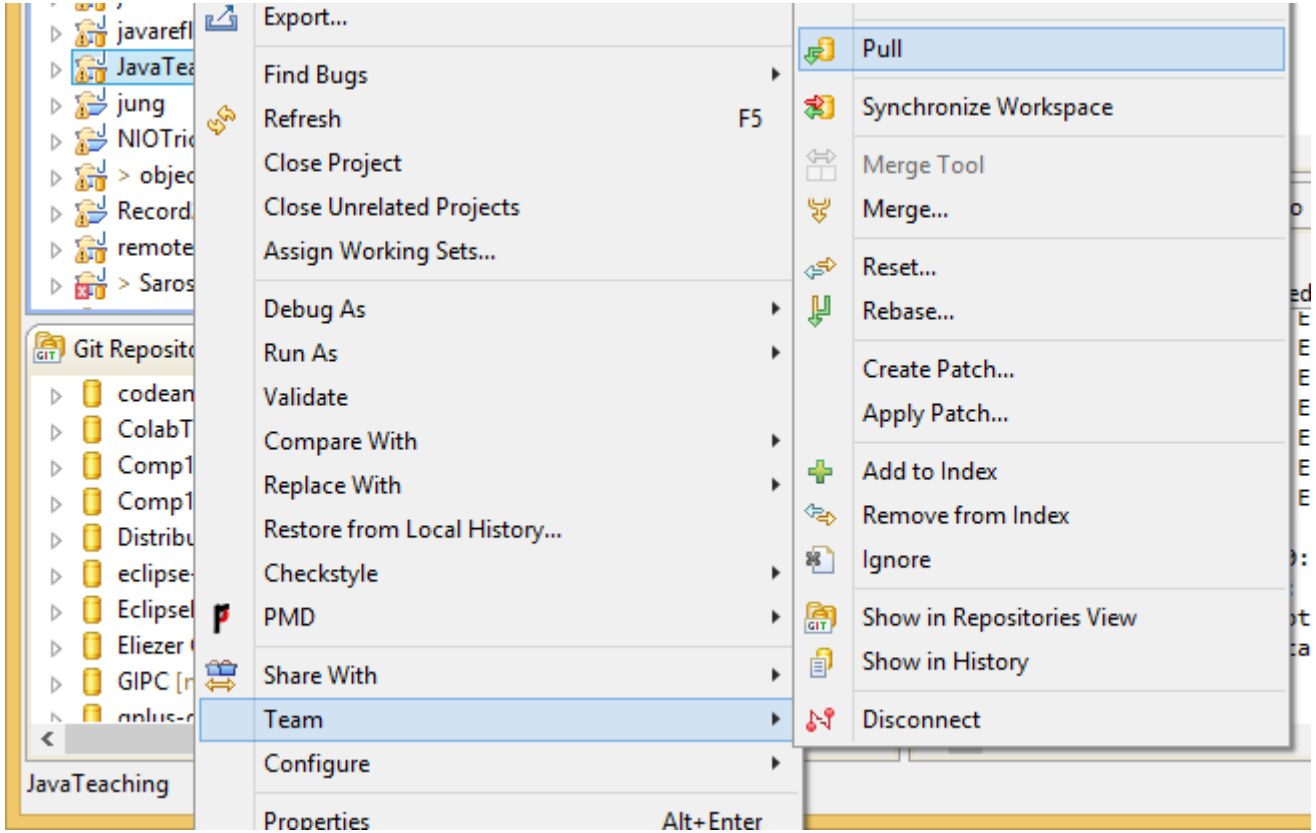
Select Add Library instead of Add External Jar and then select Junit

What your libraries in build path should look after both additions



You will probably have a later version of the JRE system library

Pulling (Updating) the Project

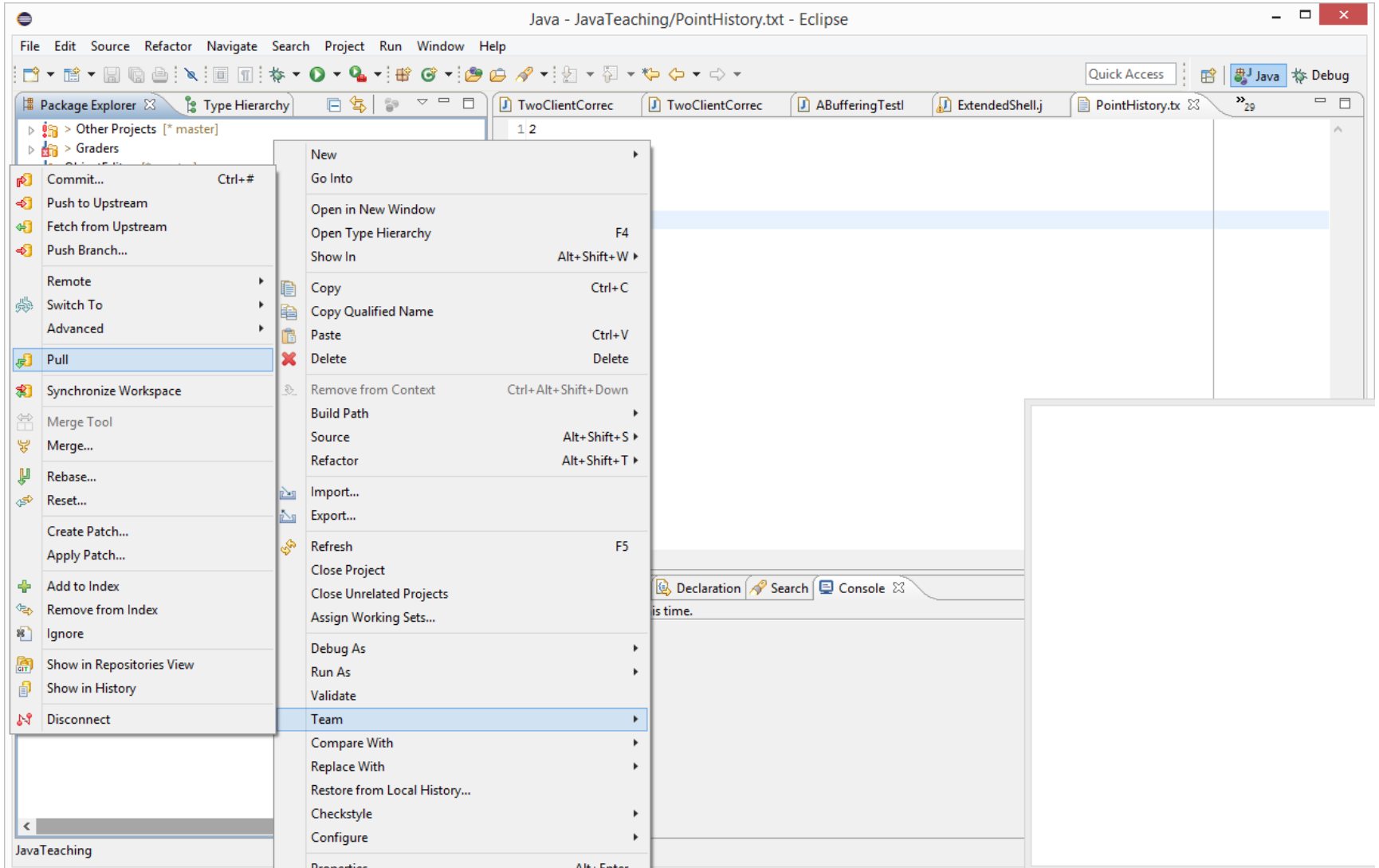


Right click project, Team>Pull (not Pull…)

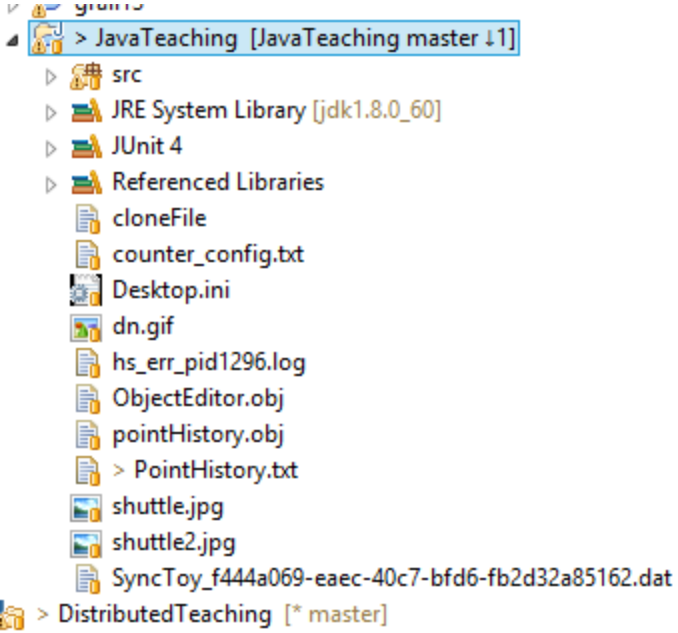
You may get conflicts if you changed files that were updated in the master version you are pulling

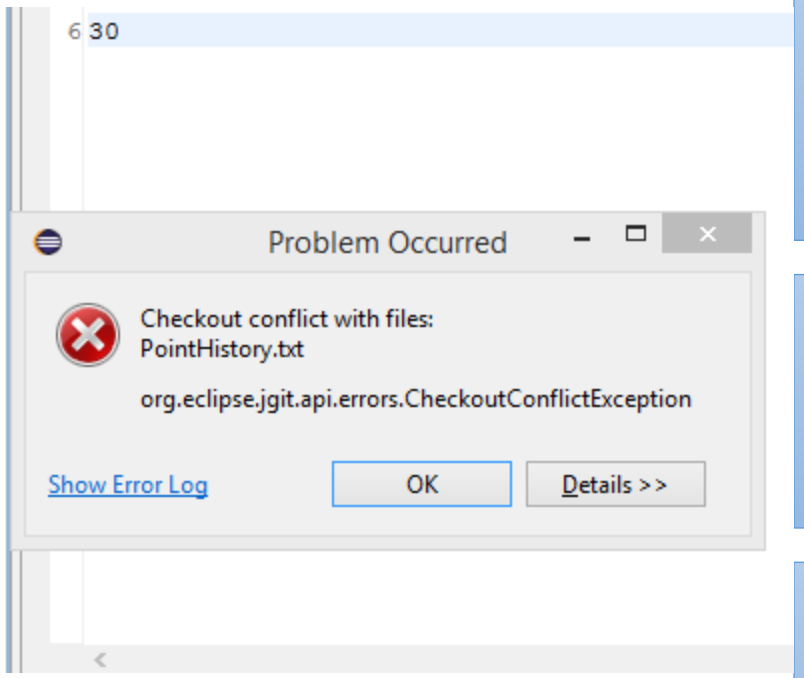
In this case, you should save the folders you changed (they will have a > next to them) and then reset the project and then pull again

Pulling the project



Conflict





This happens if you (accidentally) changed a file that was also changed by some one else – in this case the instructor

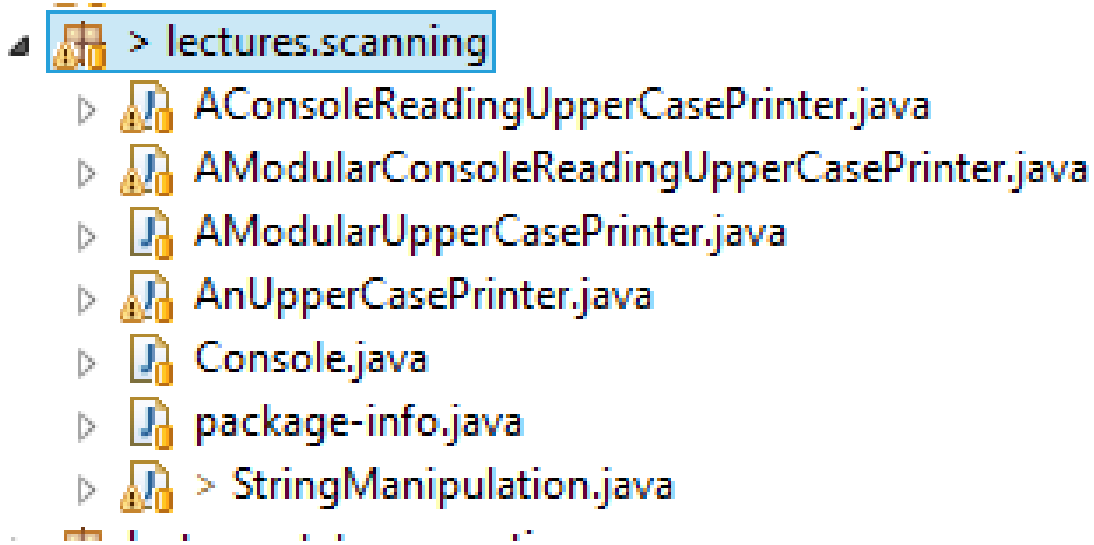
The name(s) of the changed files are given in the message, and next to the project name you see you are behind the version in the repository

If your changed files have important information, save them, or save the whole project

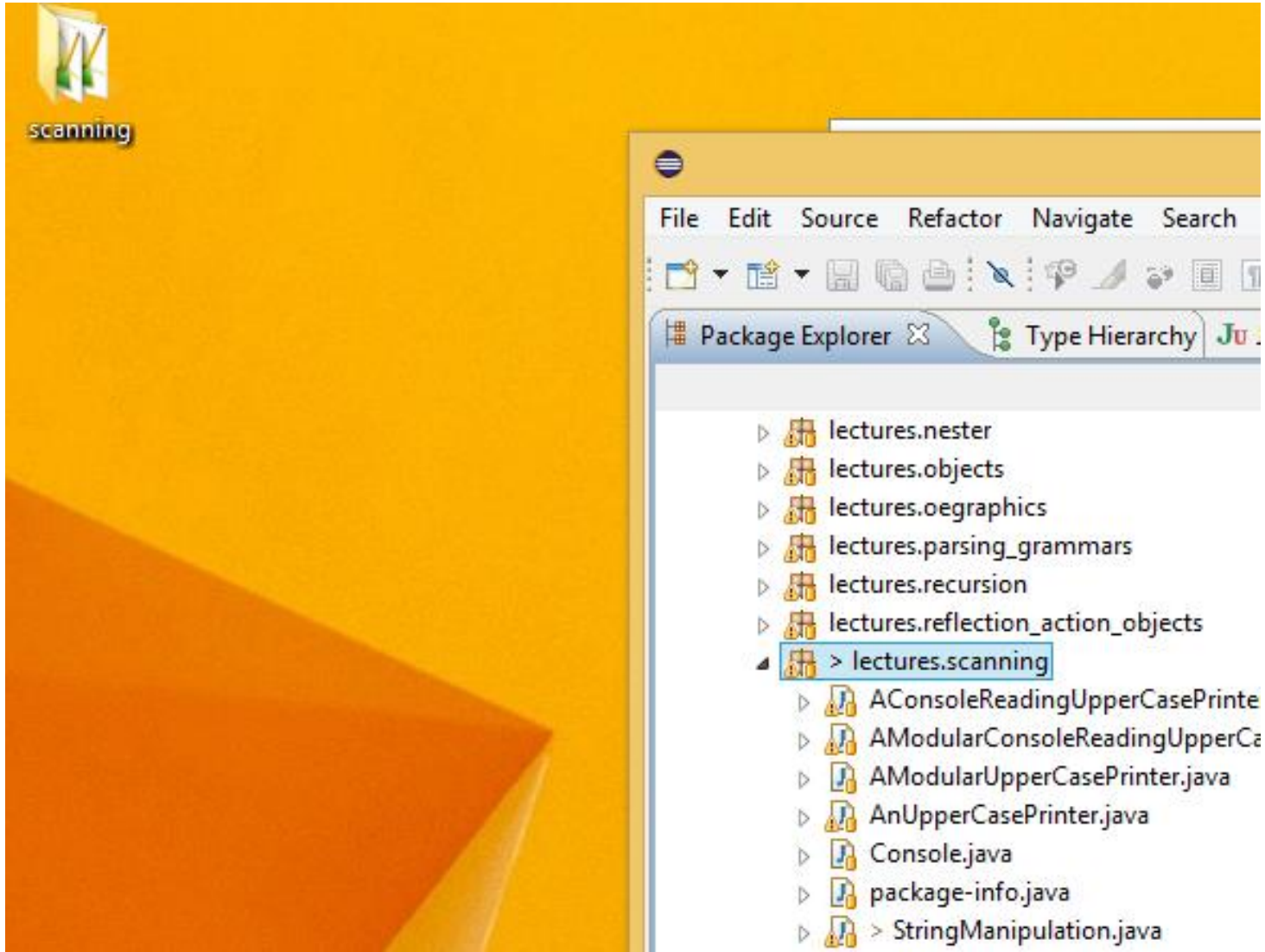
Then reset (hard) the project

Saving folders changed

Click on each changed folder/file and execute CTRL-C (copy)

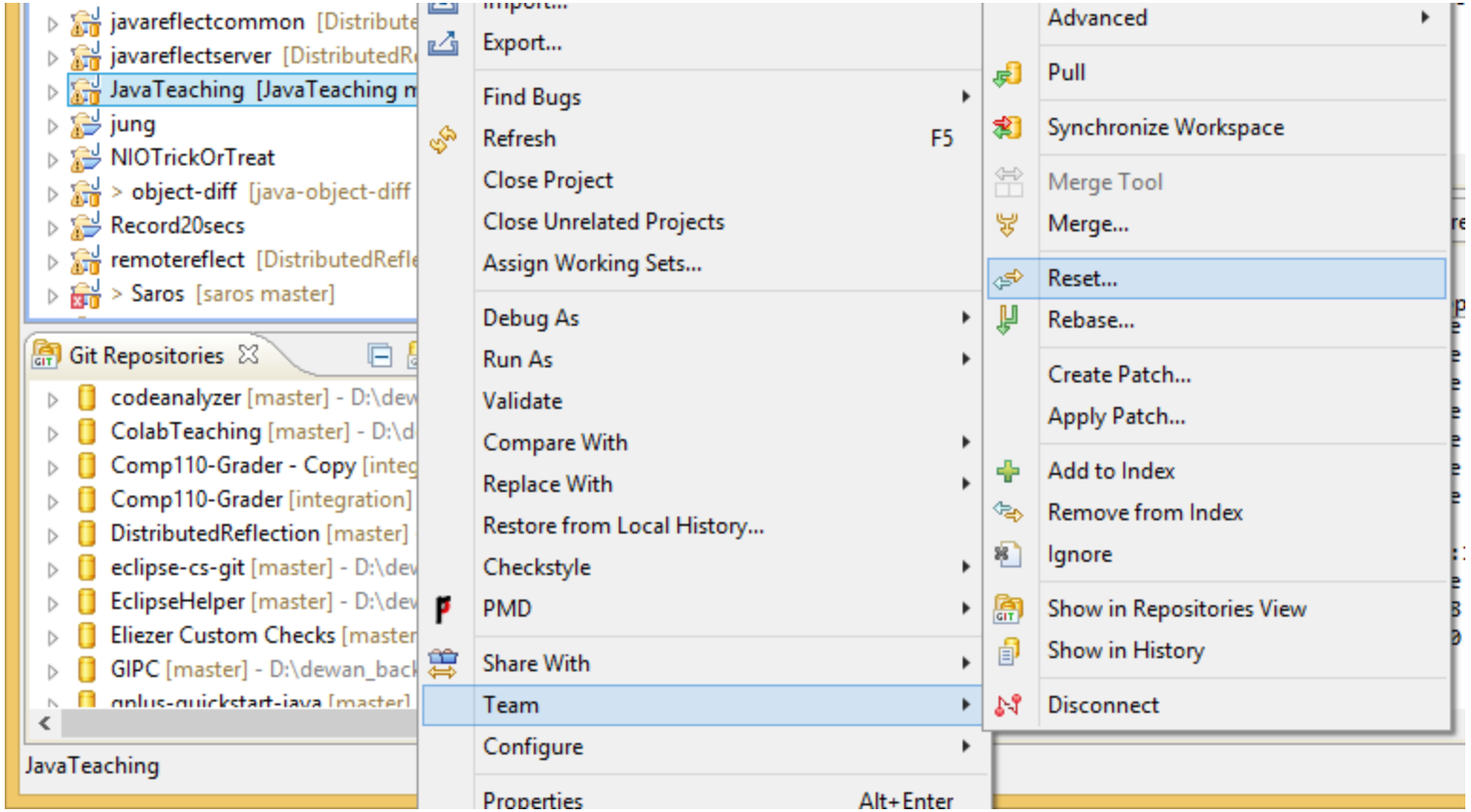


Saving folders changed



Go to desktop or some folder and hit CTRL-V (paste)

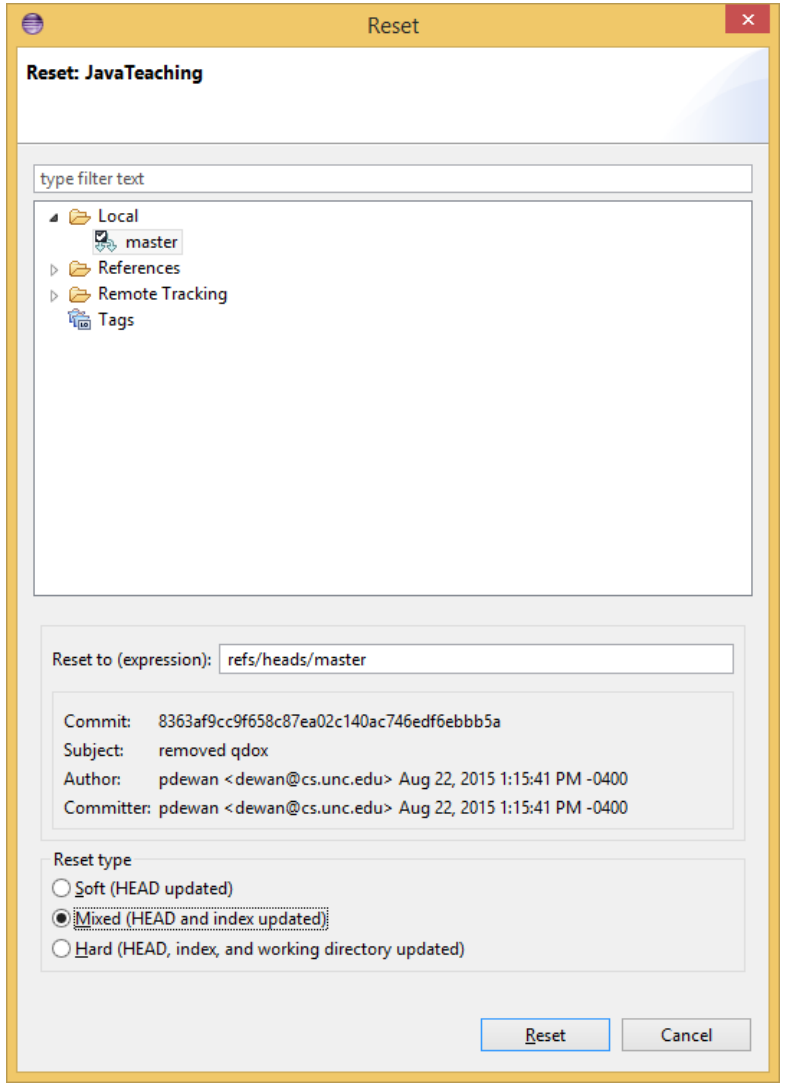
Resetting the Project



After doing an exercise with the code or when you get a conflict, you can reset the project

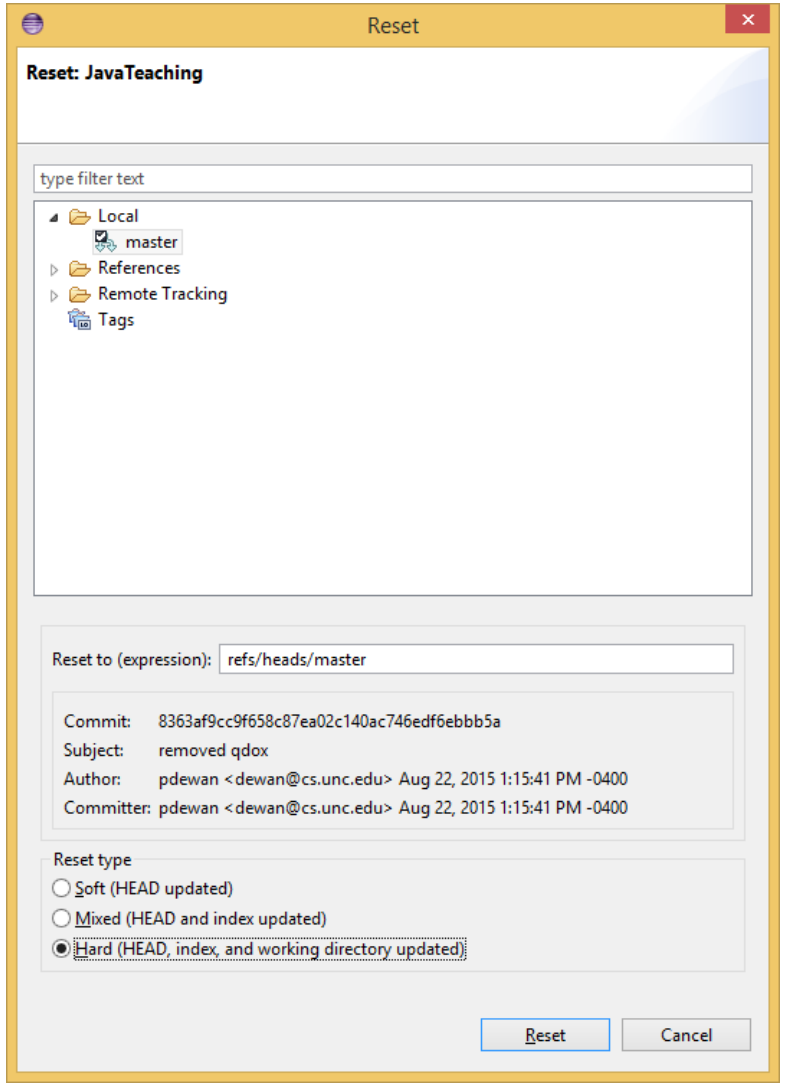
Right click project, select Team>reset

Reset Dialog to Pick Kind of Reset Option



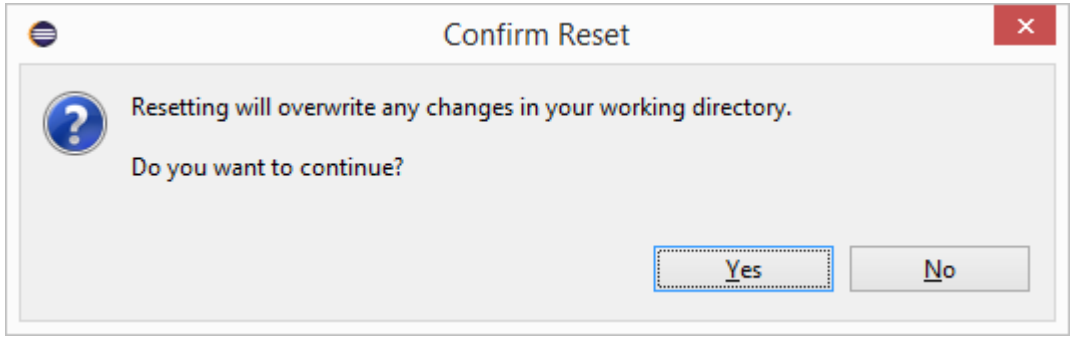
Default picks Mixed Option

Hard Reset Deletes all of Your Changes

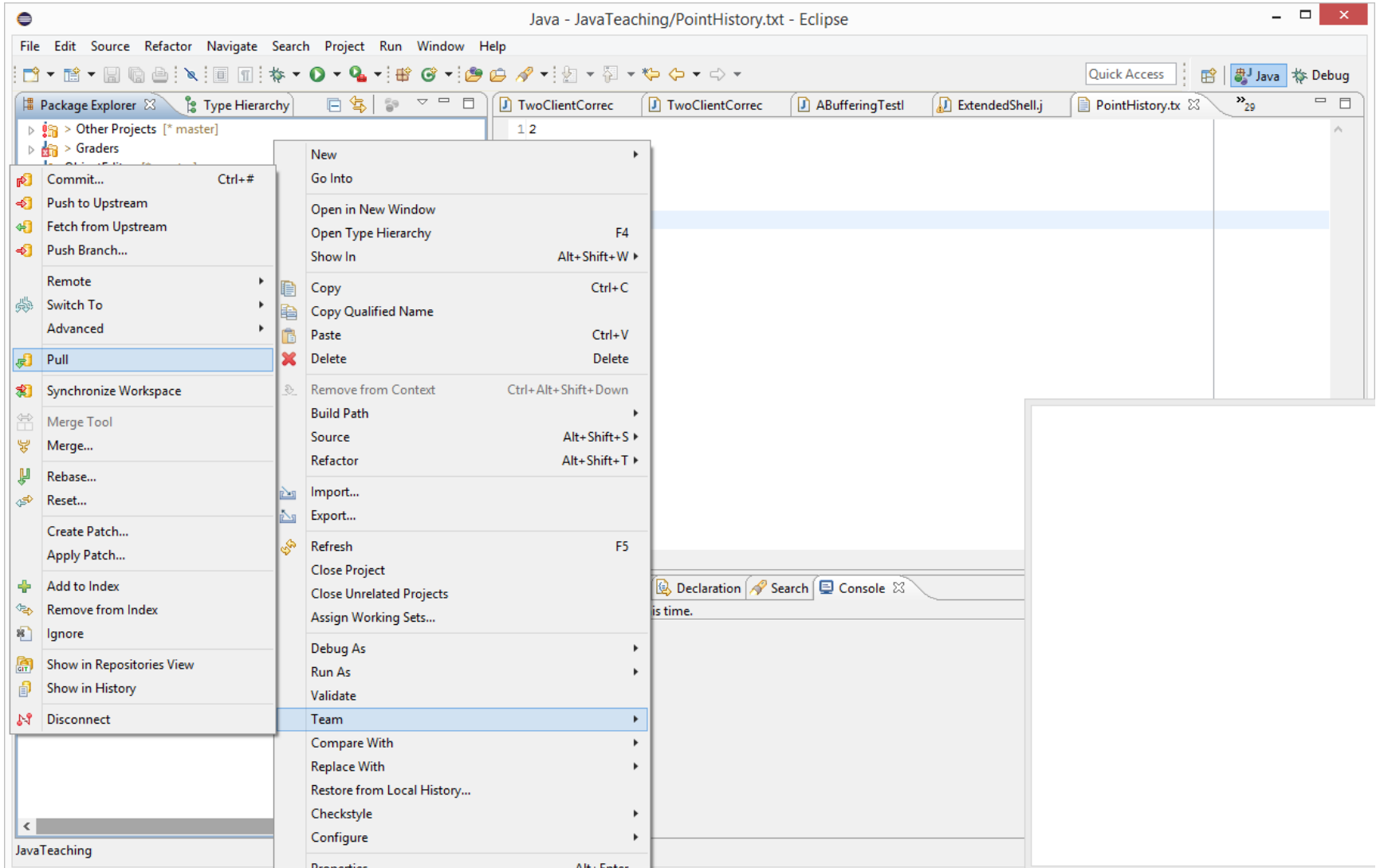


Pick hard and press reset

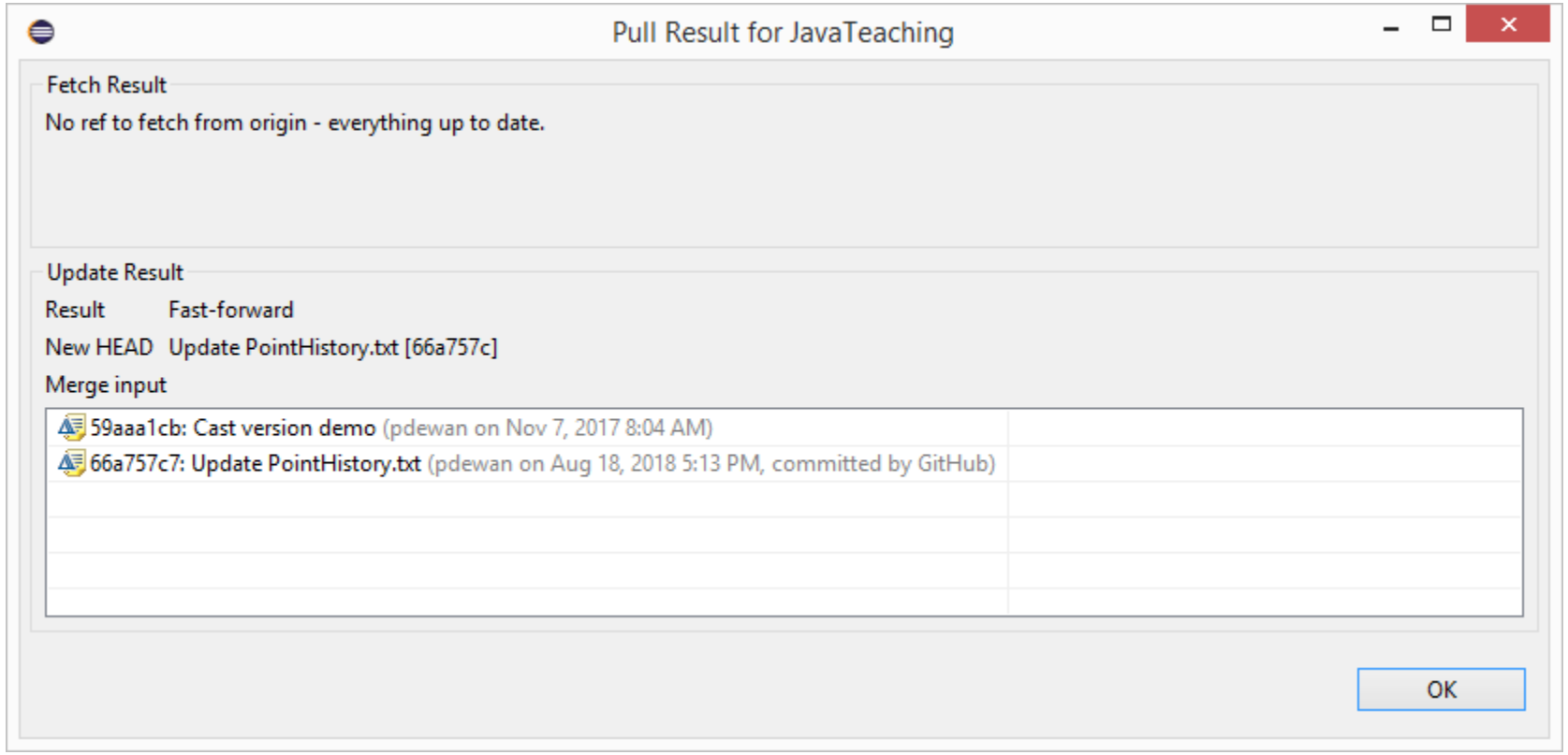
Hard Reset Deletes all of Your Changes



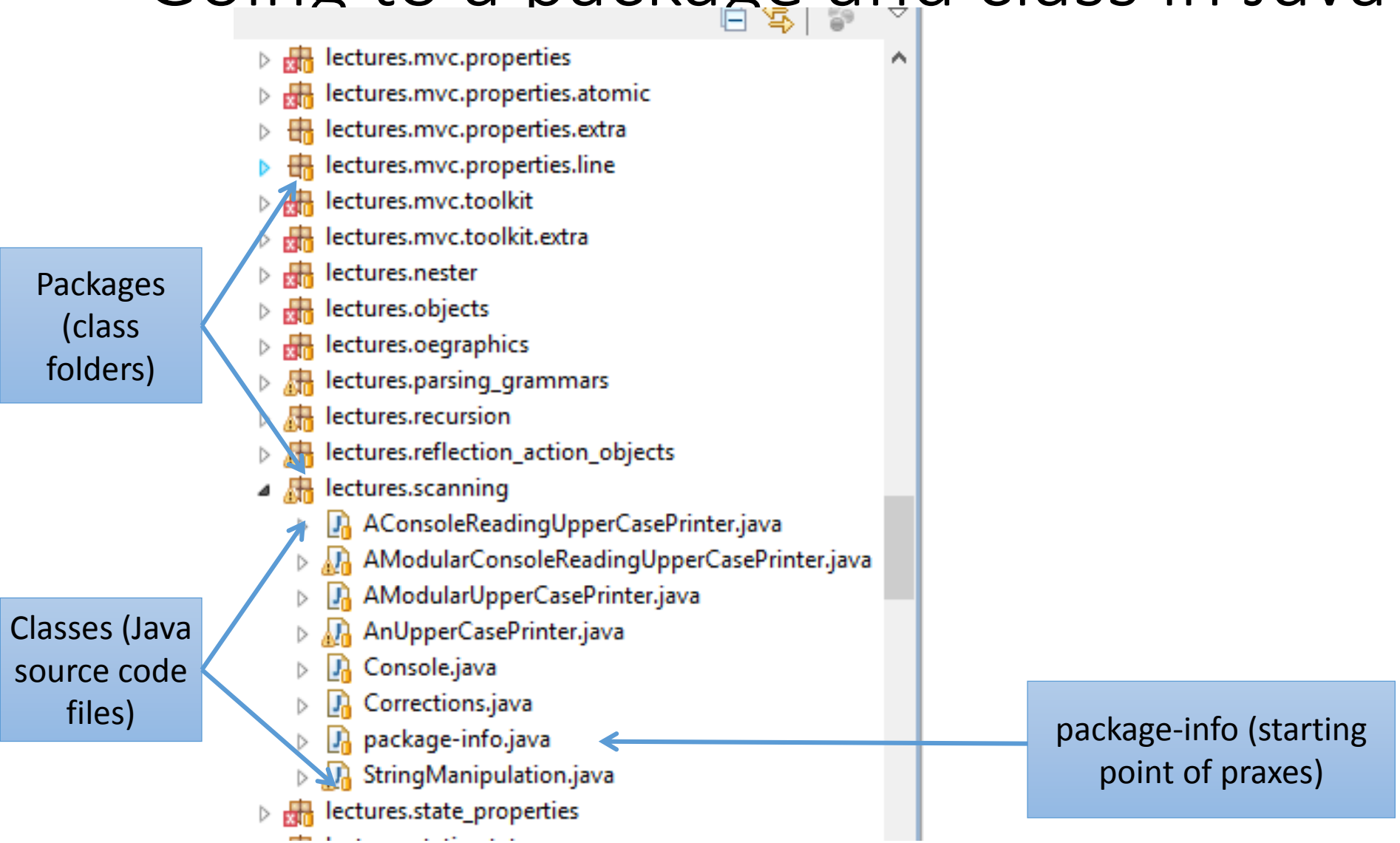
Pull the project again to get latest non conflicting changes



Successful Pull After Conflict



Going to a package and class in Java Teaching



Collapsed parts

```
 1
 3⊕ * RUNNING A PROGRAM..
56 package lectures.scanning;
57     /**
58      *
59      * IMPORTS
60      * T/F It makes sense for a class to import only those classes that are
61      * not in its package.
62      *
63      * If a class C1 refers a class C2 in a different (external) package
```

Open them

```
/**
 * RUNNING A PROGRAM
 * This is an example of a Java program called a class in Java.
 * There are many ways to execute a class,
 * If you are editing the class and your insertion point is in it,
 * Right click->Debug As->Java Application is one way.
 * Use it or some other way you know to run this program.
 *
 * If you do not see the Java Application option in Debug As, or your get a
 * console message:
 * "Cannot run or load class"
 * then your project is misconfigured and you to delete the
 * oeall.jar file from the class path.
 *
 * Look at the ObjectEditor PPT
 * http://www.cs.unc.edu/~dewan/comp401/current/Lectures/ObjectEditorLib.pdf
```