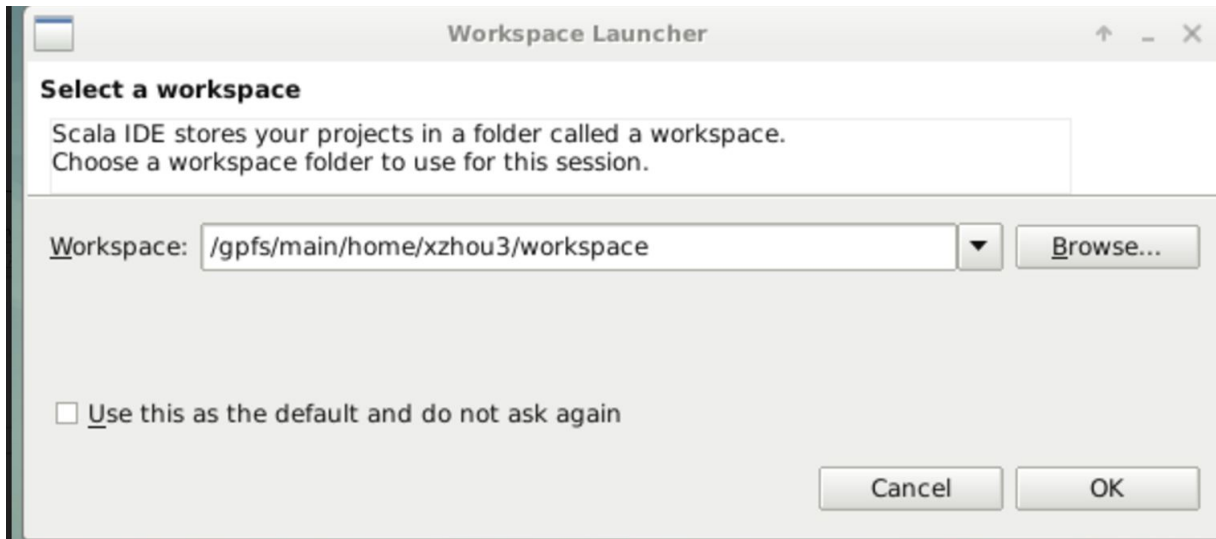


Eclipse Setup

Opening Eclipse

- Type `eclipse.photon &` into your terminal. (Don't open eclipse through a GUI - it may open a different version.)
- You will be asked where you want your workspace directory by a small window that says "Select a workspace" Simply use the default (it should be something like `home/<your-login>/workspace`).



Note: *If this is not your first time using eclipse, this may not pop-up*

- You may get multiple popups, click "Yes/Okay" when you see them.

Hint: *You may want to check the box "Use this as the default and do not ask again"*

- You should now see a welcome screen. Close it by clicking the "x" on the tab. Again, you may not see this if you've used eclipse before.

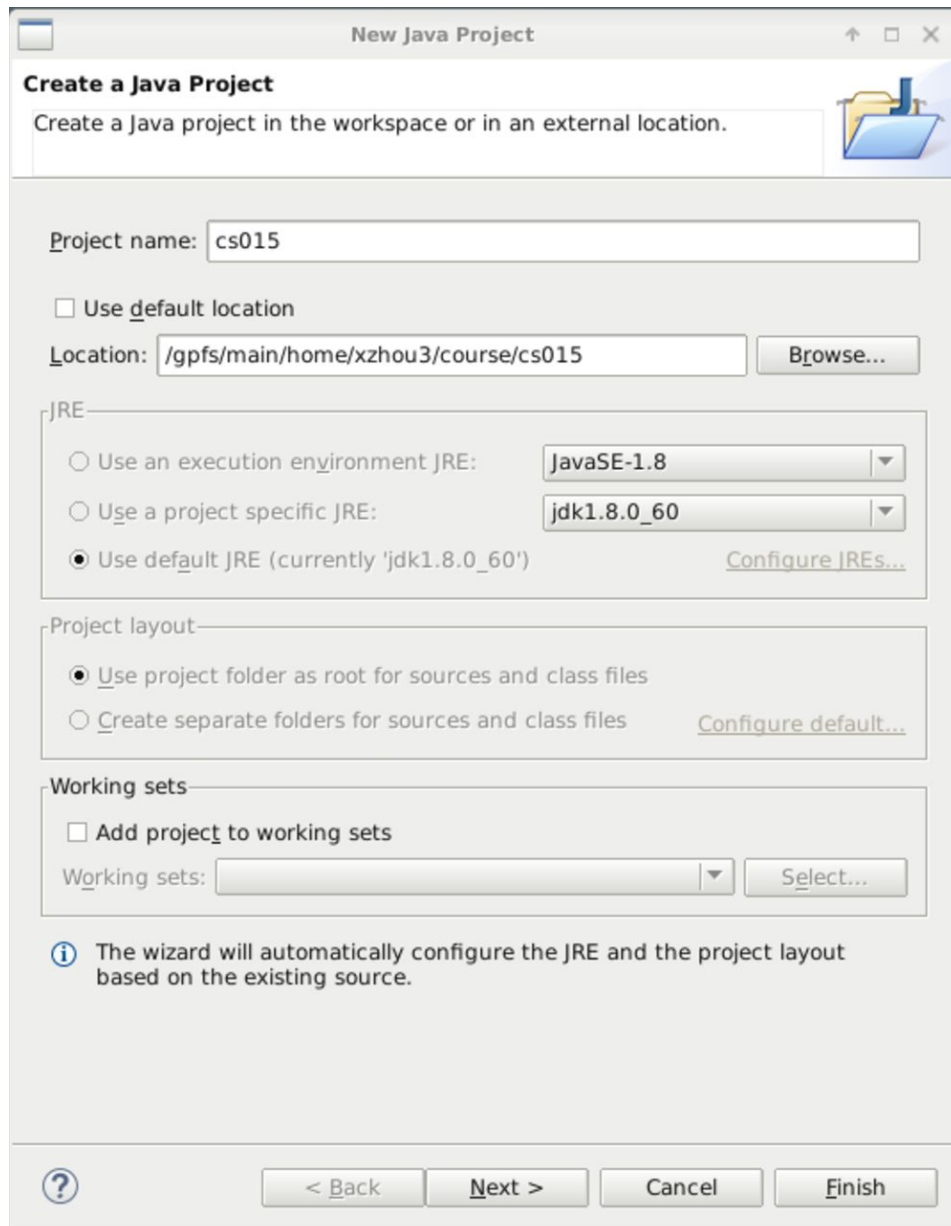
Welcome to Eclipse! Now let's get your code loaded onto Eclipse.

Setting Up Eclipse for CS15

- Click `File » New » Project » Java Project`.
- Type `cs0150` as the Project Name
- In the JRE box click the box next to `Use default JRE` (it may be autoselected).
- In the Project layout box, check the first button says `Use project folder as`

root for sources and class files.

- Look back to the top of the window, uncheck Use default location.
- Browse to /home/<your-login>/course/cs0150 and click OK.
- Your screen should now look very similar to this (except the cs015 should be replaced with cs0150):



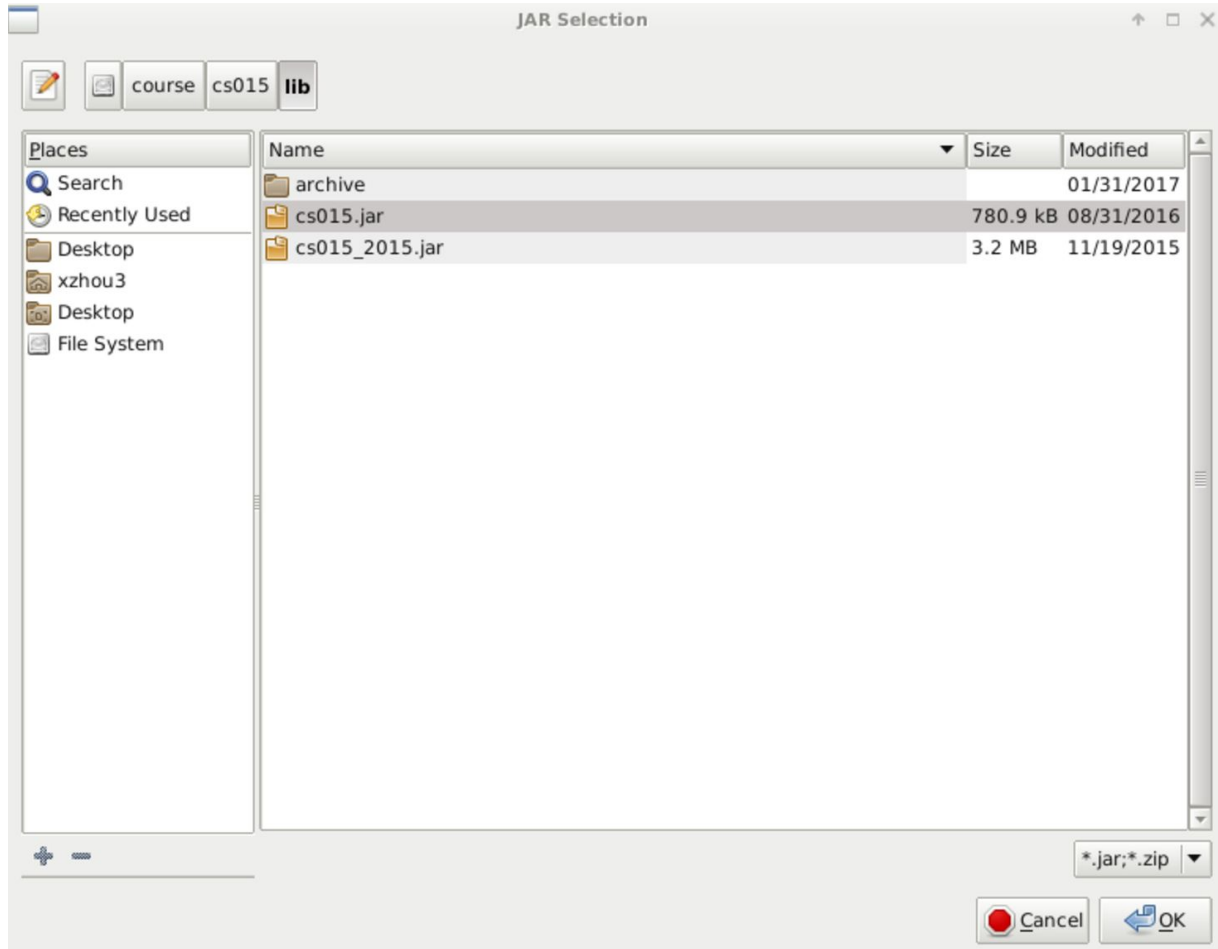
- Click Next

-
- In the Source tab, open the cs0150 project folder, and if you see the src folder,

right-click the `src` folder, and select `Remove from Build Path`.

- Right-click the `cs0150` project folder.
 - If the `Use as Source Folder` option comes up, select that. If not, do not worry! Your CS15 folder is already the source folder, and you can ignore this.

-
- Go to the `Library` tab
 - Click `Add External JARs...`
 - Navigate to `/course/cs015/lib`
Note: You must to go to the *department cs0150 folder (/course/cs0150)*, not your own (`course/cs0150`). Start by clicking `File System` on the left hand side column
 - `File System » course » cs015 » lib`
 - Select `cs015.jar`



- Click `Finish`


Now Eclipse is linked to your cs0150 folder. When you install a new project, click `File » Refresh`, and it will appear in Eclipse.

Note: Please be careful about deleting files/folders from Eclipse. You can remove them from Eclipse if you would like, but make sure to not also delete the files themselves (unless you want to).

Editing Code in Eclipse


To get some practice with editing and running code from Eclipse, as well as to familiarize you with its features, we will be working with some of your code from lab 2 in Eclipse. **If you did not complete lab 2, you should be able to complete this portion using partially-complete lab2 code or code from one of your projects!**

Working with the Eclipse Editor


- Open up your cs0150 project by clicking the triangle arrow next to it.
- Open up your lab2 package (last week's lab).
 - To the left of lab2, you should see this icon: . If that isn't there, something went wrong with your Eclipse setup, so call over a TA.
- Click on each .java file in lab2 to open the files in the editor. You can navigate between the files using the tabs in the editor region.
- To make line numbers appear on the left hand side of your files, right click on the left-most side of the **Editor Region** and check **Show Line Numbers**. These lines will be helpful when debugging.
- Go to your VideoPlayer class. Try deleting the semicolon from the first line of your VideoPlayer constructor. What happens? Hover the mouse over the red X to view the error message, then fix the problem.
- Copy the following code into your VideoPlayer constructor in VideoPlayer class:

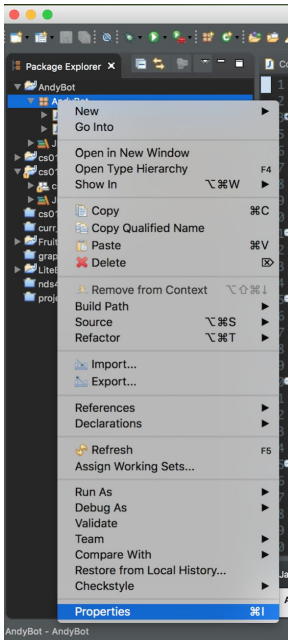
```
notAVariable = new CartoonNetworkProvider();
```

- Save the file and check out the error message. Fix the line of code so that it properly declares and initializes a lab2 variable of type LooneyTunesProvider - you can use the Eclipse "quick fix", like you did for importing, if you want!
- Hit **Ctrl + Shift + F**, which is the Eclipse command to auto-format your code. You might see some of your code change (for example, indentation levels could change and long lines could be wrapped). If not, try messing up some indentation and doing it again.
- Eclipse will import classes for you! Example:
 - Instead of using `javafx.scene.paint.Color` as parameter you can simply put `Color` instead.

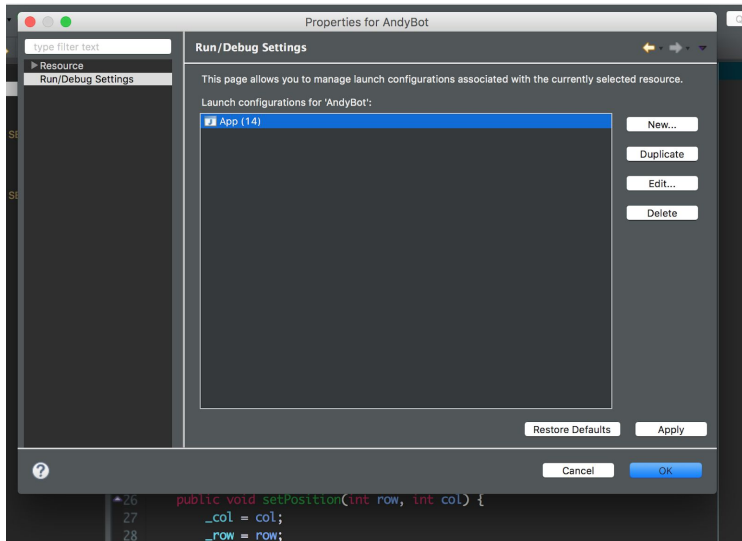
- From doing this, you would get red squiggly lines underlining some of your code. On the left in the Package Explorer, your class would also have a red 'x' next to it, indicating that there are compiler errors in the file.
- If you were to hover over the red line underneath "Color". This brings up a list of possible quick fixes - click the first option to import the Color (under `javafx.scene.paint`)
- With the class imported, you do not have to type the entire path of the class anymore!
- Fix any remaining compiler errors, and then **run your code** by right-clicking `App.java` in the `lab2` part of the Package Explorer and selecting `Run As... >> Java Application`
 - You can re-run the last application you ran by clicking the green play button () at the top of the screen!
- Your lab2 application should run!

Remote Setup

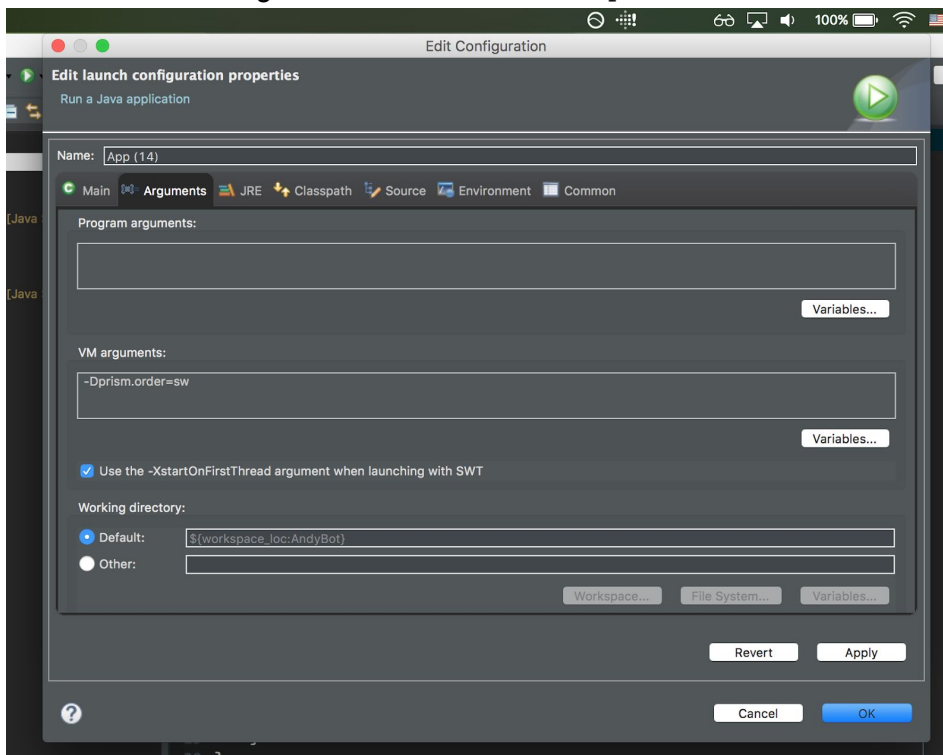
- ***You will have to do this for each cs0150 lab or project that you install, in order to run it in Eclipse over FastX or SSH.***
- Right-click on the package icon  next to the project or lab name. Go to Properties:



- Go to Run/Debug Settings, select in the main window `App` and click `Edit`:



- Go to the Arguments tab, and enter '-Dprism.order=sw' in the VM arguments box:



- Hit 'Apply' and 'ok'
- Hit 'Apply' and 'ok'
- You should now be all set to work on this project remotely in Eclipse! Make sure to do this for each new project.