

CSE 143: Computer Programming II

Homework Setup Tutorial

Introduction

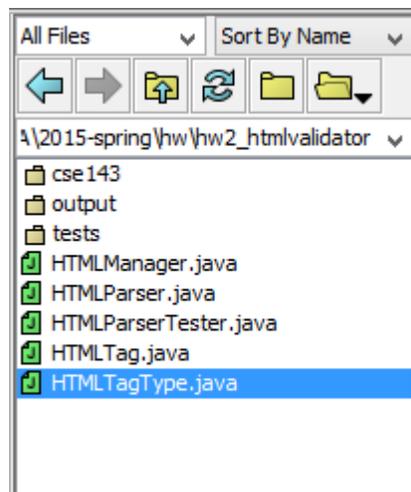
This quarter, rather than having you download a wide variety of different utility classes to use on your homework assignments, we are going to be providing a `cse143.util` package instead.

Because this may be the first time that you have needed to work with a custom package, this guide will cover how to properly set up your files and environment for jGrasp and Eclipse.

If you're curious on how to go about setting up your files using only the command line, or want to learn more about how exactly Java finds and loads classes and packages, please see the "Advanced Setup Tutorial" which can be found on the course website.

JGrasp

Our ultimate goal is to have your files organized similar to the image below.



More specifically, for each homework assignment, what you want to do is to have a separate folder per assignment. Each homework folder should contain:

- The `cse143.util` package folder
- A folder named `tests` that contains any sample input or log files that we've provided
- A folder named `output` that contains any output files that you've generated
- Any client files that we've provided
- Any files that you are meant to write and turn in

You should always try and organize your homework assignment in a similar fashion. At minimum, you should have a separate folder per homework assignment to avoid potential conflicts between files which can lead to subtle, hard-to-debug bugs.

We also strongly encourage you to have a separate `tests` and `output` folder to help your workspace from becoming cluttered, since some assignments will have many different tests and output. However, if you find that this is confusing, then it's ok to place files on the top level (though if you're confused, you should probably talk to a TA first before giving up).

Step-By-Step Instructions

- Create a folder for the current homework assignment. For example, if you're about to start homework 2, create a folder named hw2 or hw2_htmlmanager.
- Download all the zip files associated from a given assignment from the class website to your hw2 folder.
- Extract all zip files.
- Create a folder named output.

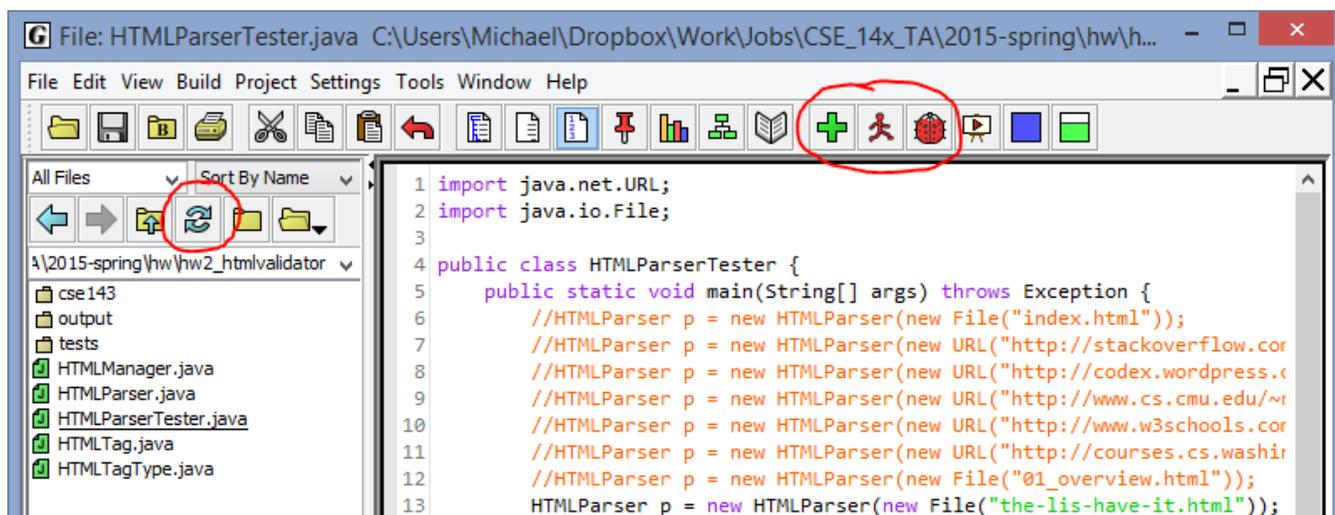
More Information

You should not have to rearrange any of your files – simply extracting the zip files should automatically place the files in the correct location in your hw2 folder.

Note that the cse143.util package is simply a regular folder named cse143 containing a single folder named util. This util folder will contain a wide variety of different utility classes that we will be using throughout this quarter, including our Stack and Queue interfaces and implementations.

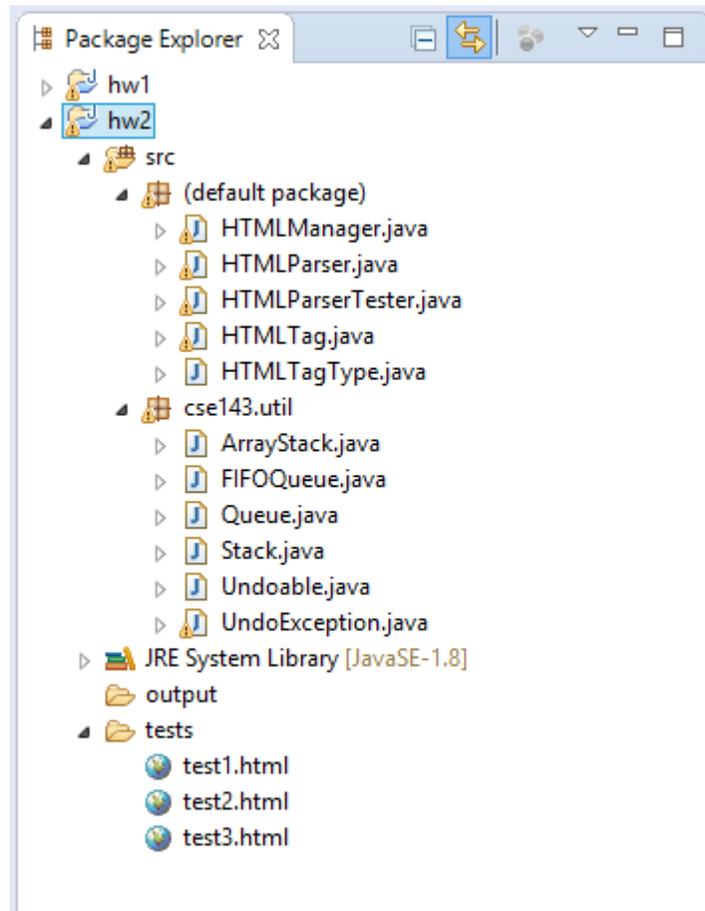
When running client files that ask you to provide a path to an input file or an output file, simply type in tests/ or output/ followed by the filename. For example, you would do something like test/test1.html and output/test1.fixed.html, not test1.html and test1.fixed.html.

In order to add a new Java file, click "File > New > Java". To run a client or test file, double-click it to open the file in the editing pane and click the "Run" button. Be sure to manually recompile your implementor class (the green "plus" button) before running the client to make sure the client is always using the latest .class file possible. To refresh the list of files on the left, click the "Refresh" button.



Eclipse

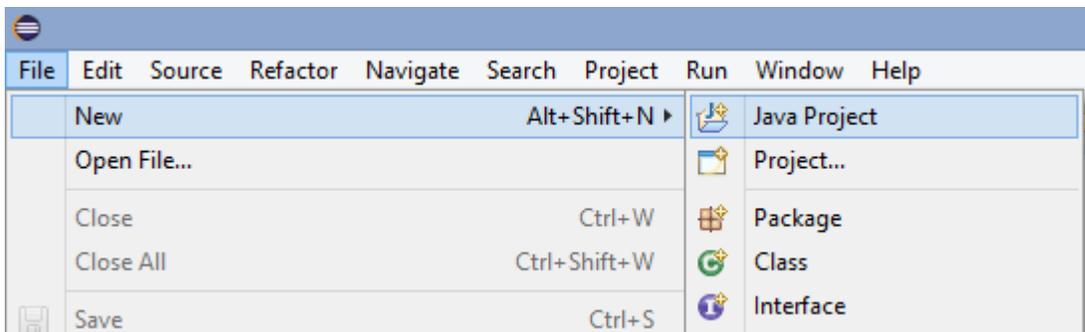
Setting up Eclipse is slightly more complicated. Our ultimate goal is to have our folder structure looking something like the below image.



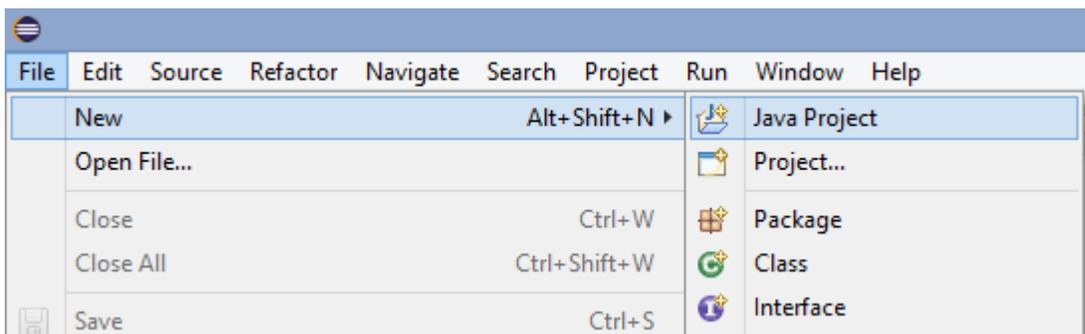
More specifically, we want a separate project per homework assignment, which should be organized as follows:

- src: contains any Java files
 - (default package): contains any client files that we've provided, and any files that you are meant to write and turn in.
 - cse143.util
- JRE System Library: Eclipse will provide this by default. It's safe to ignore this.
- tests: contains any test files or sample input
- output: contains any output that you generate
- contains any other files which are not Java files but a particular client may need

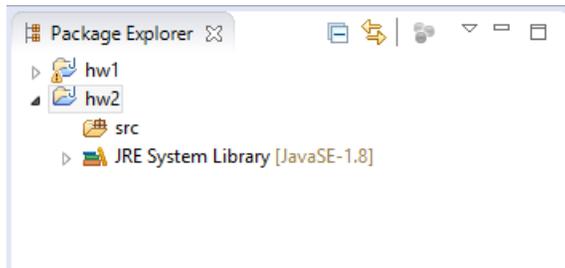
To start a new project, click “File > New > Java Project”.



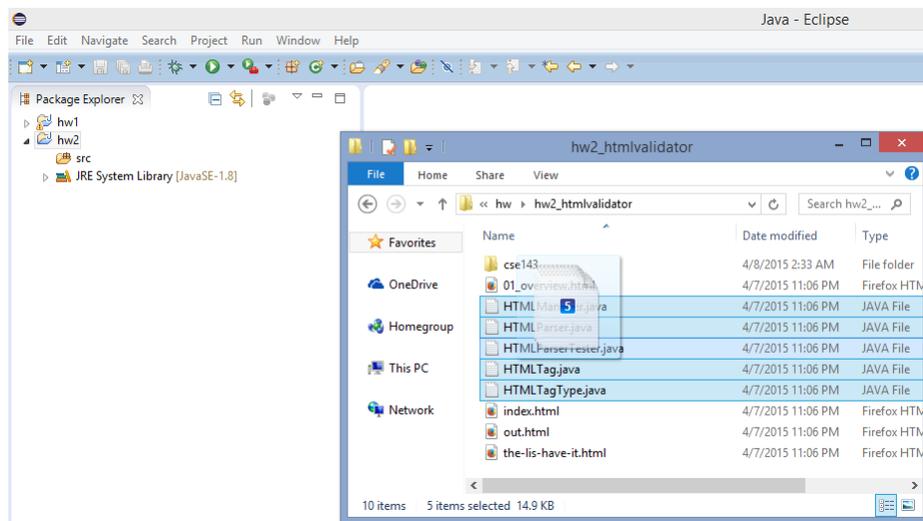
Give your project a name in the “Project name” textbox – something like “hw2” or “hw2-htmlmanager”. It’s safe to keep everything else at its default value. Hit “Finish” once you’ve given your project a name.



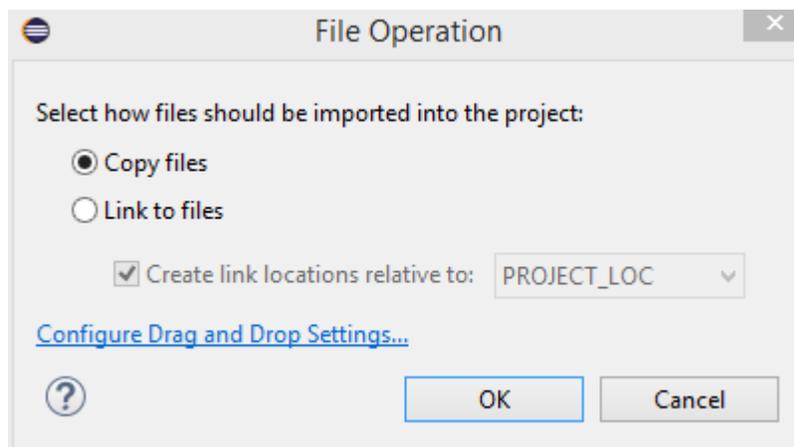
You should now see something similar to the following in the “Package Explorer” pane:



Now, download and extract all zip files associated with the current homework assignment from the course website. Drag and drop all Java files into the src folder in the “Package explorer” pane in Eclipse.

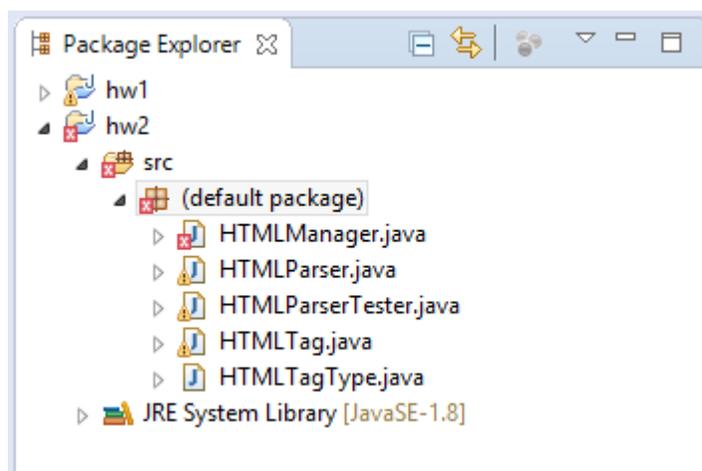


A dialog box should appear asking if you want to “Copy files” or “Link to files”. Stick with the default (“Copy Files”) and select ok.



The Package Explorer pane should now look something like the following. Eclipse may report that a few files have some errors – do not worry about those for now. The errors are most likely because some of the clients are attempting to import `cse143.util`, which we haven't added to the project yet.

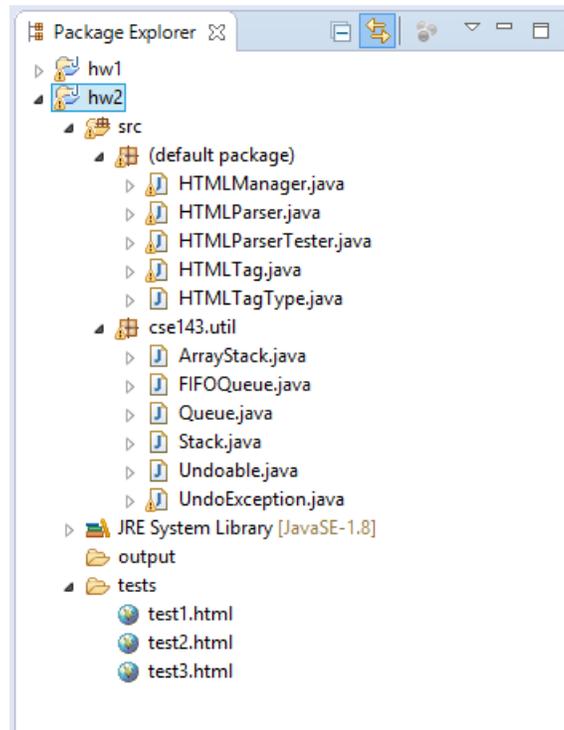
If this zip file contains any `.txt` files, you will need to move them and drag and drop them on the top level `hw2` folder. We do not have any files like this in this homework assignment, so we'll skip this step.



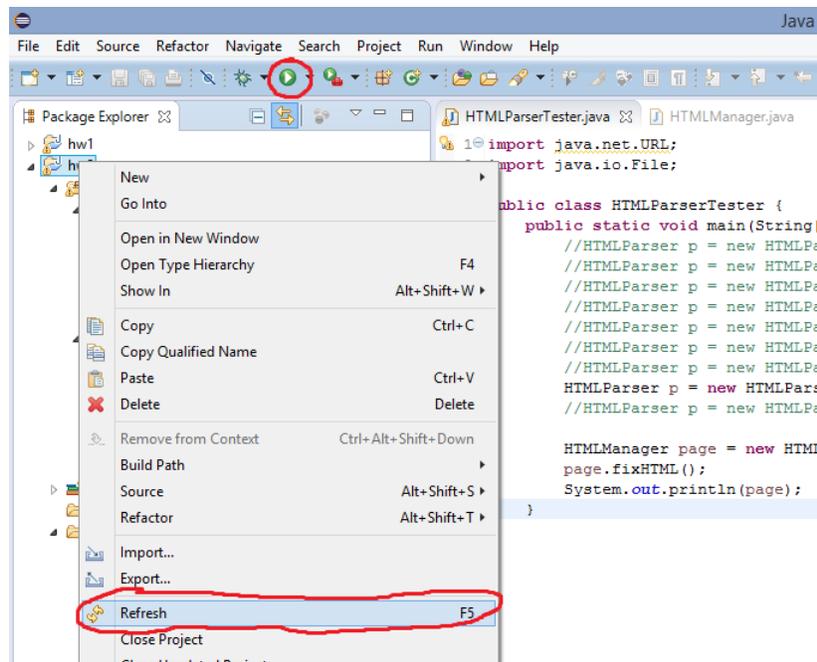
Now, we are going to repeat the same thing two more times, once to import the `cse143.util` package, and once to import any test that we need. Drag and drop the `cse143` folder on top of the `src` folder. Drag and drop the `tests` folder on top of the entire project folder – in this case, on top of `hw2`. If any dialog boxes appear, just go with the defaults and hit “ok”.

Finally, right-click on `hw2` and click “New > Folder” and create a new folder named `output`.

You should now see something like the following:



In order to add a new Java file, right-click on “(default package)” and select “New > Class”. In order to run a client file, double-click it to open it in the main editing window, and click the green “Run” button. If the client file creates any new files, either hit “F5” or right-click hw2 and click “Refresh” to update the Package Explorer pane to see the latest files.



Similar to jGrasp, be sure to specify the full path to any test or output files that you want to generate when running any client programs. For example, you would do something like test/test1.html and output/test1.fixed.html, not test1.html and test1.fixed.html.