TCSS 305 Programming Practicum, Autumn 2019
Assignment 0 – Part B
Value: 2% of the course grade
Due: Sunday, 6 October 2019, 23:59:00

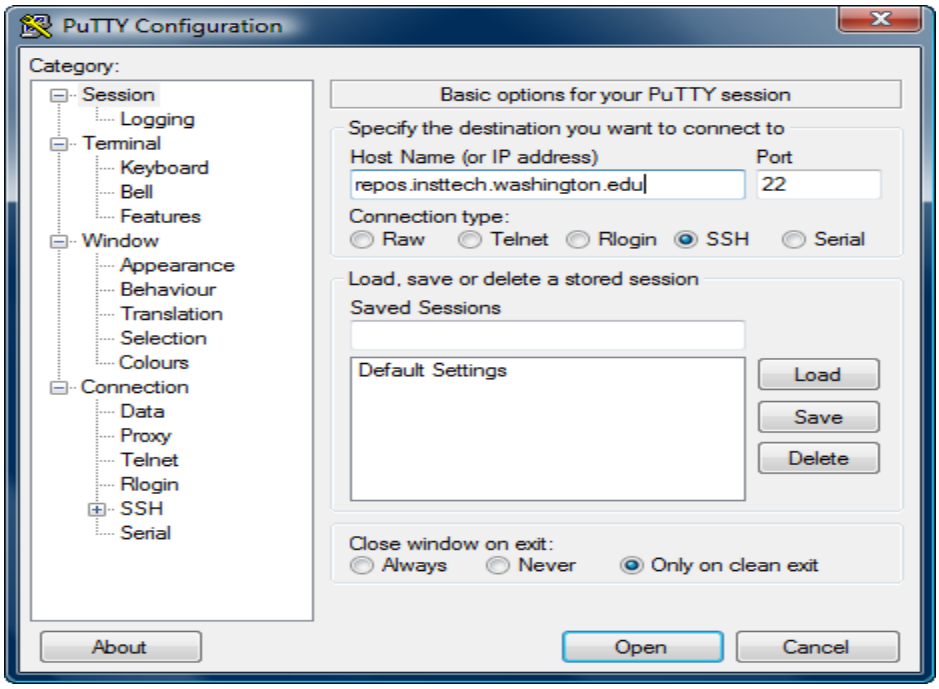
Assignment Description:

This assignment provides instructions to help you set up your Subversion repository. If you are using your own computer, you should have completed Part A of assignment 0.

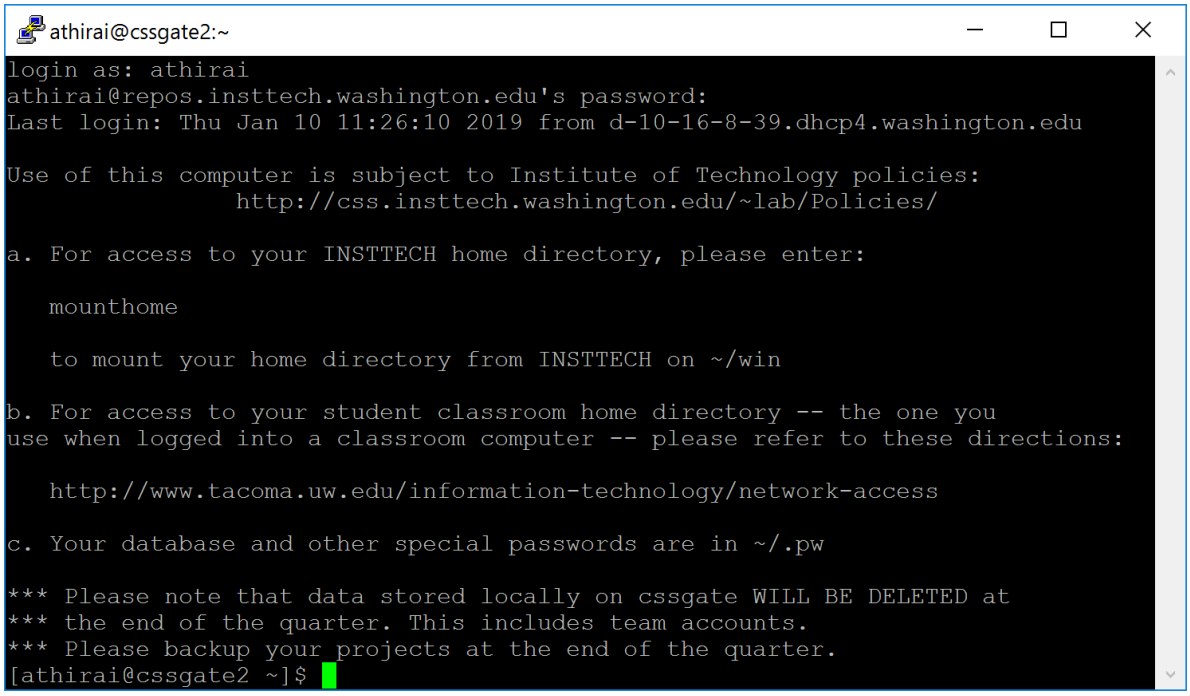
Subversion Repository Creation:

To set up your Subversion repository, perform the following steps:

1. Use an SSH client to remotely log in to `repos.insttech.washington.edu`, using your UWNetID as the username and your UW password.
    a. If you're on a Mac, or UNIX/Linux, you can use the terminal (Terminal.app, xterm, etc.).
        i. `ssh repos.insttech.washington.edu`
        ii. enter your UWNetID and password as prompted
    b. If you're on Windows, and you don't have an SHH client already, you can get PuTTY (A free client program) at: https://www.chiark.greenend.org.uk/~sgtatham/putty/
        i. The host name you will enter is: `repos.insttech.washington.edu`



        ii. After opening the client, enter your UWNetID and password as prompted. You will then see a screen as shown below



c. If you are using a home computer and want to connect to `repos.insttech.washington.edu`, you will first need to connect to the UW Network. Please see https://www.lib.washington.edu/help/connect/husky-onnet

2. Execute the following commands:
   a. `mkdir -p svn`
   b. `svnadmin create svn/css305a19`
3. Execute the command `ls svn/css305a19`;
   a. if your repository was created correctly, you should see something like the following. If you don't see something like this, try the above procedure again. If it still doesn't work, contact the instructor for help.

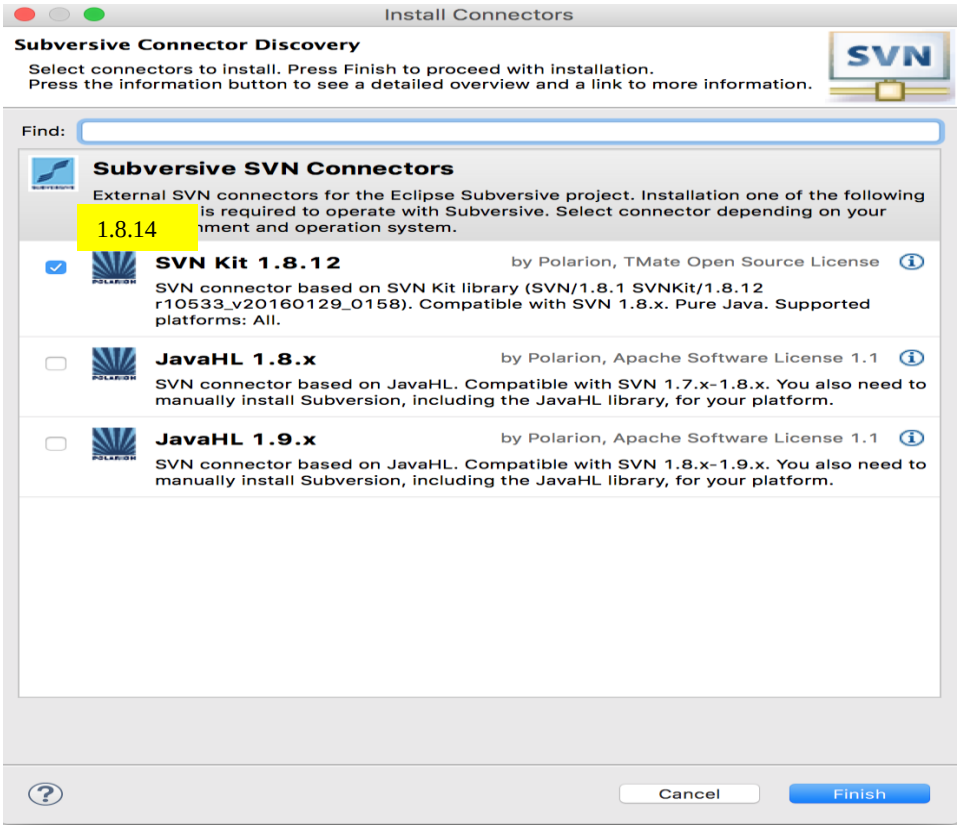
```
conf  db  format  hooks  locks  README.txt
```

Connecting an Eclipse workspace to your SVN repository:

After you have an SVN repository on the repos server and have an Eclipse workspace, you will need to connect your workspace to the server. This must be done one time for each workspace that you use.

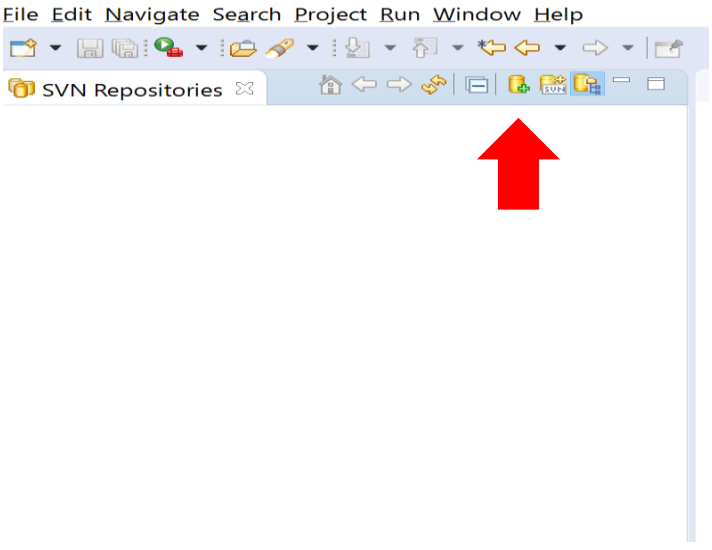
1. Start Eclipse and open the workspace you want to connect to the server.
2. Goto `Window->Perspective->Open Perspective->Other` … and select "`SVN Repository Exploring`"

   NOTE: If SVN connectors have not been installed you may see a dialog similar to the image below. If so, select SVN Kit 1.8.14 and click "`Finish`".



You will then see two more install dialogs related to SVN connectors. Click "`Next>`" in each of the dialogs. Accept the license and click "`Finish`". If you see a warning about installing unsigned content click "`OK`" to continue. When prompted restart Eclipse and then return to step 1 above.

3. Start the "`New Repository Location`" wizard

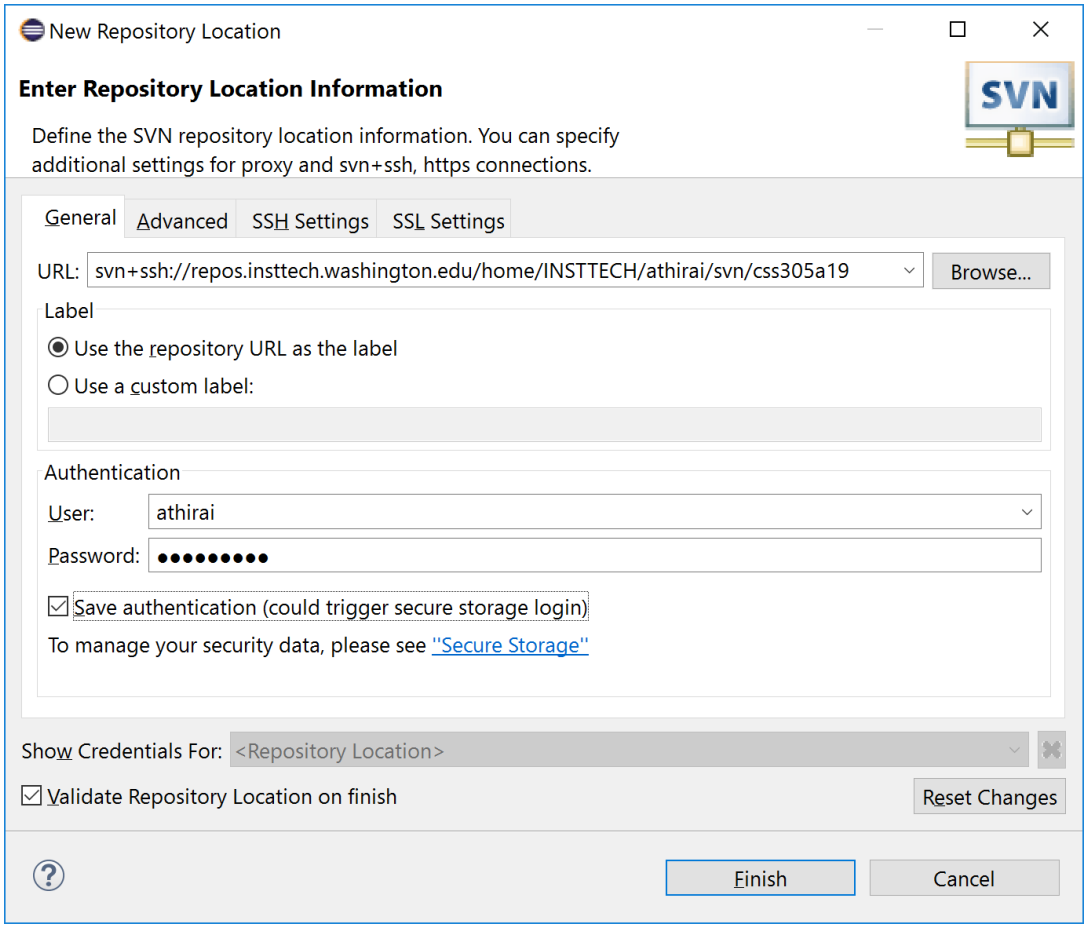


4. Fill in the window that appears as shown below.

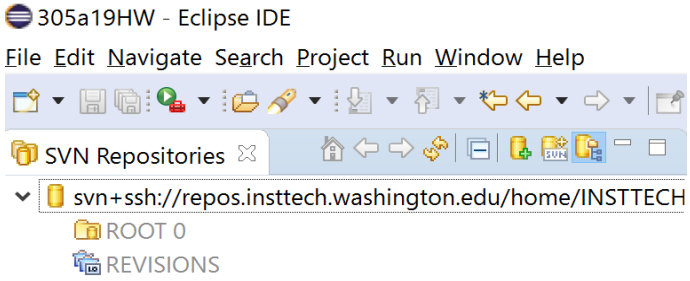
    a. The "`URL`" to be typed in the box is:
    svn+ssh://repos.insttech.washington.edu/home/INSTTECH/your_username/svn/css305a19
    (replace ***your_username*** by your UWNetID in the above URL),

b. Under "`authentication`", `enter` your UWNetID as well as your INSTTECH domain password. The INSTTECH domain password is the same as UWNetID password.

c. Click "`Save authentication`" if needed and click "`Finish`".



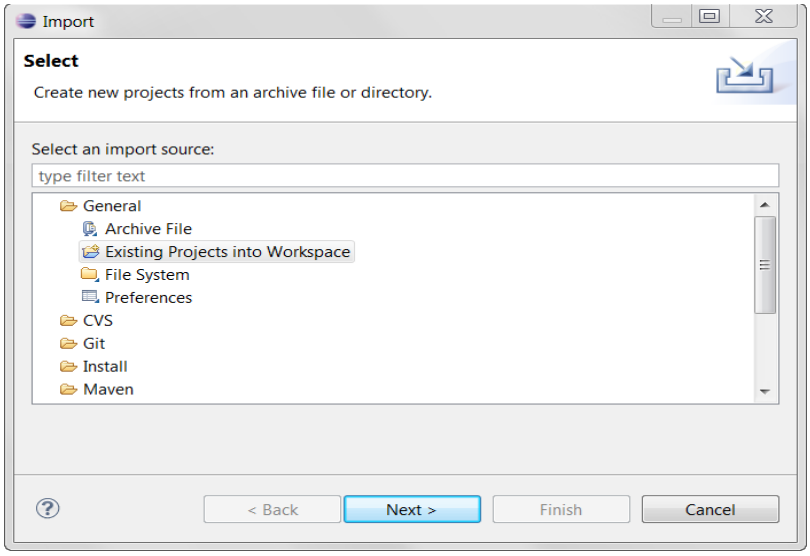
5. Your Eclipse workspace is now connected to the repos server and will see a screen similar to the one shown below. You will not need to repeat this process unless you choose to work from a different workspace.



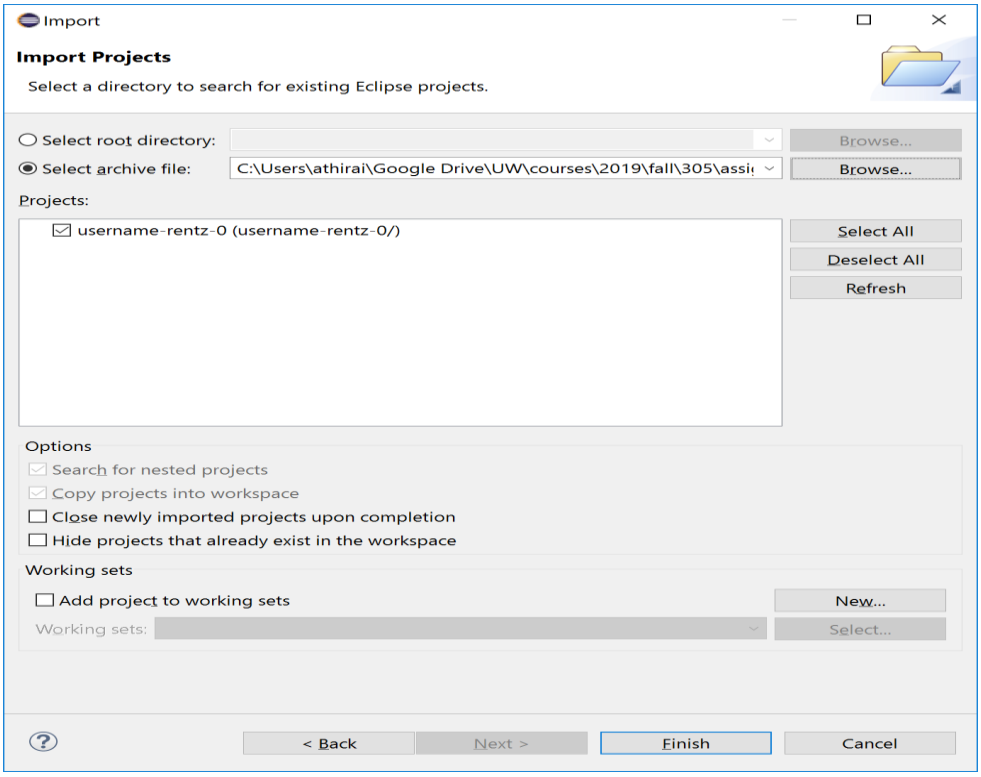
6. Go back to the JAVA perspective: `Window->Perspective->Open Perspective-> Java`

Project Setup:

1. Download the `hw0.zip` file from the Assignment 0 page on Canvas.
2. Open Eclipse with the workspace you created previously, if it is not already open.
3. From the "`File`" menu, select "`Import...`". You will (after clicking on "`General`") see a window :



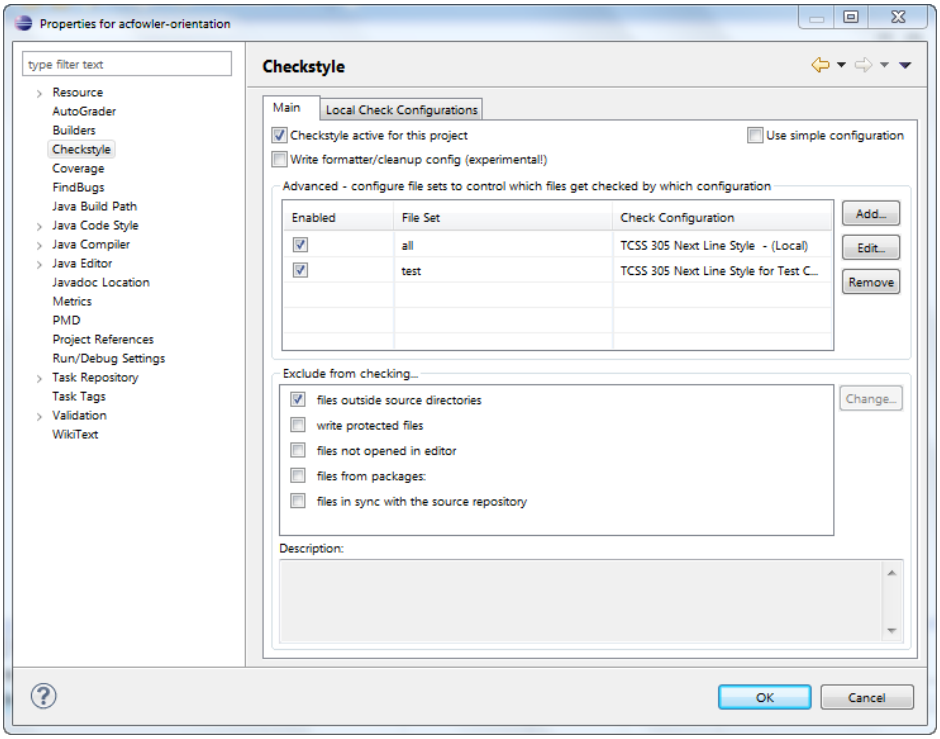
4. Select "`Existing Projects into Workspace`" (as shown in the window above) and click "`Next >`". You will then see an "`Import`" window. Activate the "`Select archive file`" radio button, and either use the "`Browse...`" button to locate the `hw0.zip` file that you downloaded in step 1 or type its full path name into the text box. Note that the listed project name is "`username-rentz-0`". Click "`Finish`", and the project will be imported into your workspace.



5. Rename the project: select the imported project "`username-rentz-0`" in the package explorer. Click on "`Refactor->Rename`" from the menu. Replace "username" with your UWNetID (for instance, your instructor would name his project "athirai-rentz-0"). It is extremely important that you name your projects correctly. Incorrectly named projects will receive a grade penalty. It is also extremely important that you rename your project ***before*** you check it into Subversion. The entire project name including your UWNetID should be in lower case (the server is a case sensitive Linux machine).

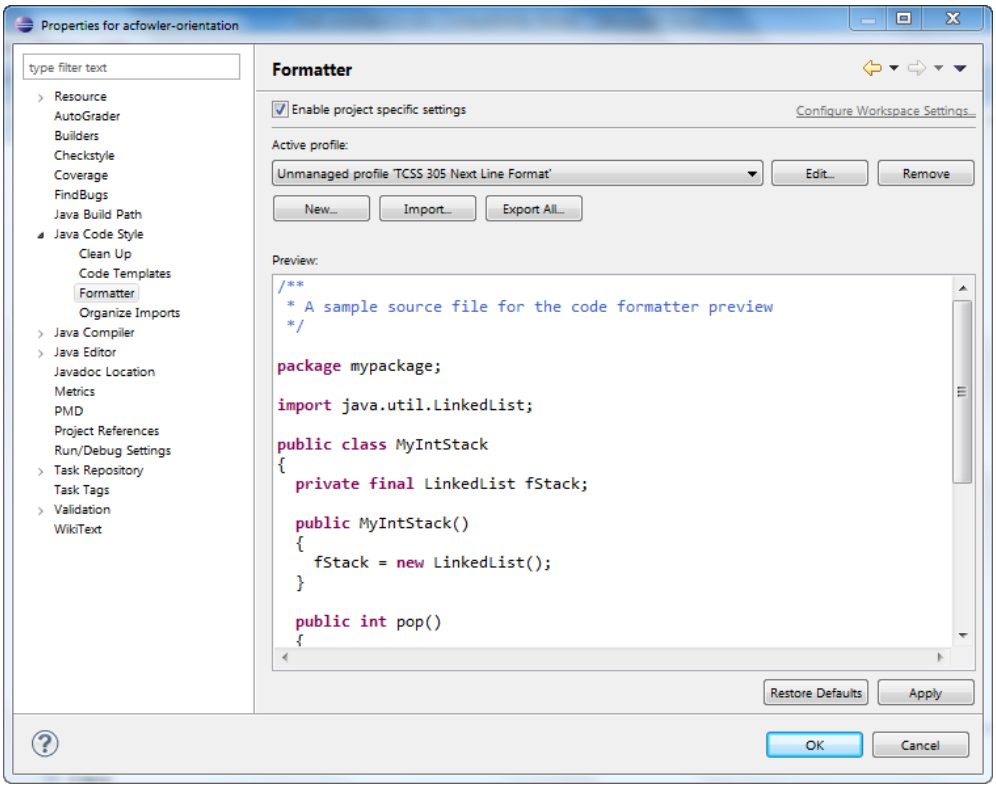
6. **[Optional]** By default, the project is configured to format code with opening braces on the same line as code. **If** you prefer to put your opening braces on the next line, rather than on the same line as code, right click on the project "`username-rentz-0`" and select "`Properties`".

    a. The Checkstyle configuration can be changed by selecting "`CheckStyle`". In the "`Main`" panel, change the configuration for "`all`" and "`test`" by clicking "`edit->check configuration`" and selecting the other code formats ("TCSS 305 Next Line Style – (Local)" for "all" and "TCSS 305 Next Line Style for Test Code – (Local)" for "test") from the dropdown menu.



    b. To change the code formatter settings, goto "`Java Code Style -> Formatter`". Click "`Import…`" and select the file `support_files/tcss305_format_nextline.xml` supplied with the project. This will give you the "TCSS 305 Next Line Format". Once you have performed this import operation, it should not be necessary on subsequent assignments within the same workspace ("TCSS 305 Next Line Format" should already be in the dropdown menu for you to select).

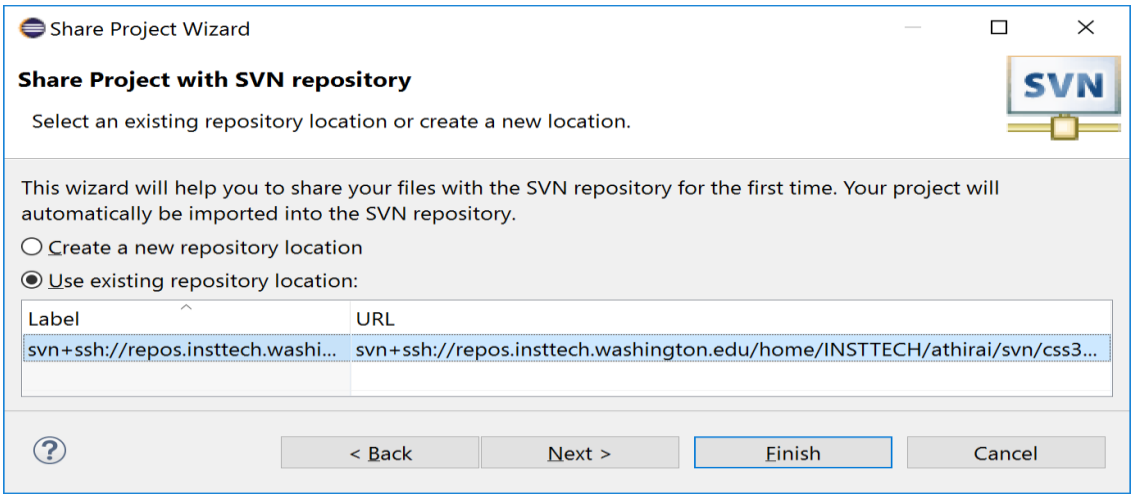
If you chose to change the Checkstyle and Formatter settings you may see a dialog requesting to rebuild the project, click "`Yes`".



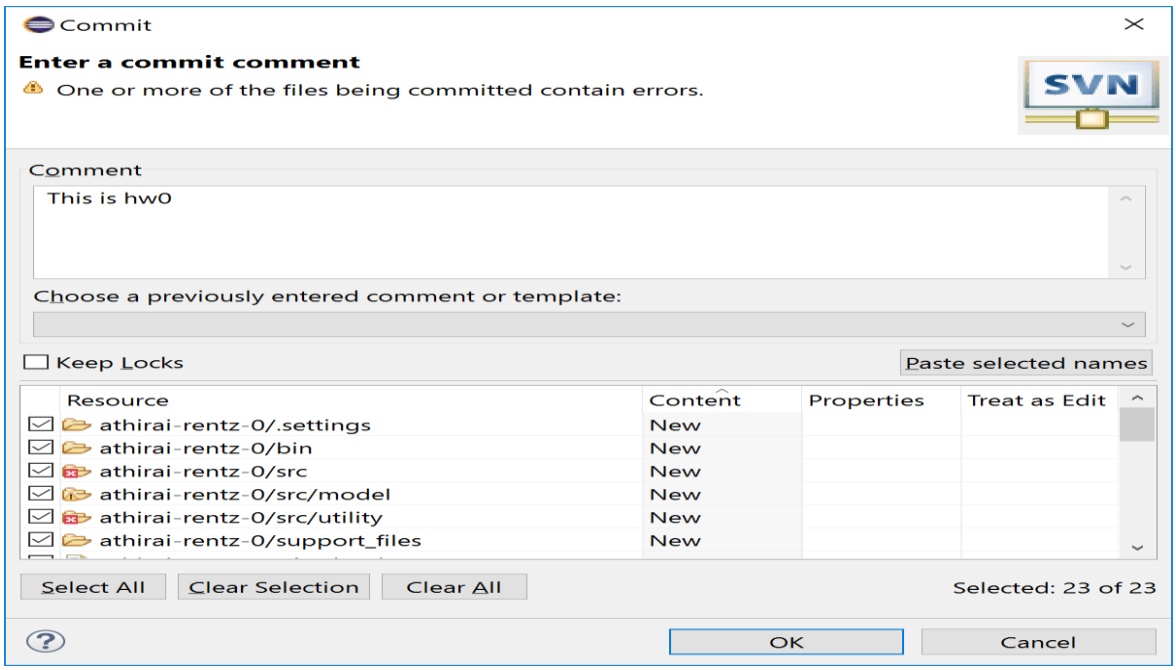
7. Format your source code. To use the formatter, right click on the project and select `source -> format (or use Ctrl-Shift-F)`. This should apply the new format to all code in the project.
8. The project "`username-rentz-0`" contains 4 errors and 3 warnings. There should be no other errors.

Submitting your project to the SVN server:

1. Right-click on the project name on Package Explorer (in this case, "`username-rentz-0`") and select "`Team / Share Project…`". You will be offered a choice between Git and SVN. Choose `SVN`, the click "`Next>`". The next dialog should show your repository information being selected. Click "Finish". You will do Step 1only when you "first" submit a project to the SVN repository. New changes to a project that has been previously submitted to SVN repository can be saved using Step 6.



2. You will be asked to enter a descriptive comment about the files you are committing, which you should enter in the `Comment` section.

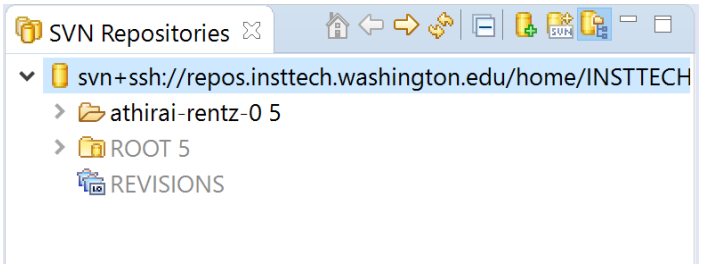


3. Remember the project "`username-rentz-0`" contains 4 errors and 3 warnings. There should not be any other warnings or errors other than this when you commit to the SVN repository. You will receive a penalty for additional errors or warnings.

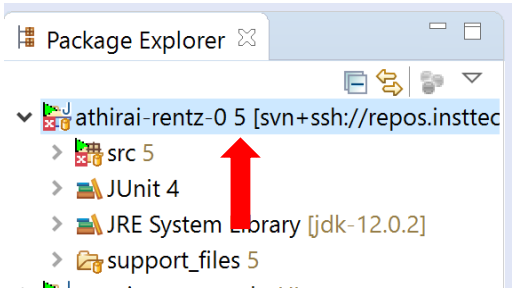
4. Goto SVN perspective `Window->Perspective->Open Perspective->Other` … and select "`SVN Repository Exploring`"

   a. Right Click on your repository and select "`Refresh`"

   b. You should see "`username-rentz-0`" under your repository



5. Go back to the JAVA perspective: `Window->Perspective->Open Perspective-> Java`. Right click on your project, select "`Team->Update`". When you do this, you will see a number listed besides your project name. This is the SVN revision number. Always perform "`Team->Update`" before you note the revision number as eclipse may not display the correct one.



You will need to report your revision number in the executive summary. The revision number shown above can differ from your revision number.

6. If you have made further changes to the project and want to save the new changes to the SVN repository. Right click on your project, select `"Team->Commit"`. The new changes to the project that has been previously submitted to the SVN repository will be saved to the SVN repository. Make sure to note the most recent SVN revision number using Step 5.

Submitting your executive summary:

The executive summary template will be provided to you in canvas. Download the file. Rename the executive summary "`executivesummary-username-rentz-0`", where `username` is your UWNetID. Fill and upload the *executive summary* to canvas in the same WORD format. The grader will grade your submission based on the executive summary as well as the project you submit to the SVN repository. If you do not submit either of them, you will receive no grades.

Grading Rubric (Total 100 points):

Task	Max Score Possible
Executive Summary -Submission on canvas	10
Executive Summary – Correct Naming	5
Executive Summary- Correct SVN revision number	20
Project – Submission to SVN repository	50
Project – Correct Naming	5
Project – No additional warnings/errors. There are already 4 errors and 3 warnings for HW0	10
Total	100

APPENDIX

Subversion/Version Control Best Practices (for TCSS 305):

To get the greatest benefit from version control systems like Subversion, you need to use them correctly and consistently. Here are some general guidelines.

1. When sitting down to work on your project, *always* perform an update ("Team / Update") on the entire project before starting to modify files. This is not too important for your work in this class, unless you work on your projects from multiple locations (with multiple Eclipse workspaces), but it's a good habit to develop because it ensures that you always get the latest changes that have been made to the code before you start working.
2. When you stop work – and any time you make a major change, such as a refactoring – *always* perform a commit ("Team / Commit…") on the project. It is generally preferable that you only commit code that compiles (and, in some software engineering situations, also passes all its tests), but this is not necessary if you're the only person working on the code. Since you will always be the only person working on your code, you should commit very frequently. It is far better to have high Subversion revision numbers than to realize that you need to see what the code looked like when it worked last night, but you can't because you didn't commit it before you broke it.
3. The revision number next to the project name may not always update to the current revision number when you commit new changes; performing an update on the entire project, even if it does not change any files, will cause the revision number next to the project name to update. You will need this number when you submit the assignment (as it will be used to check out your submission), so it is *very important* that you have the correct one. When in doubt, *always* perform an update.

If you follow these guidelines, you will never have to resolve Subversion conflicts (because you are the only person working on your code). If you do run into Subversion conflicts, or other Subversion-related issues, ask for help as soon as possible.

Subversion Damage Control (for TCSS 305):

It is possible, despite your best efforts, that something will at some point go wrong with your version control repository. Here are two ways to attempt to fix it. There are more subtle ways of fixing such problems, but we are not going to discuss version control to the extent necessary for you to learn them in this class. Do *not* ask for help with mysterious Subversion errors until after you have tried both methods. If you have any problems that cannot be fixed using one of these methods, ask for help as soon as possible.

1. The first thing to try when a commit or an update fails is always a "clean up" operation. Right-click on your project and go to "Team / Cleanup", then try the commit or update again. If it works, wonderful! If not, use method 2.
2. The surest way to fix a Subversion problem (conflict, *etc.*) is to delete your project and check it out again from Subversion. Of course, you may have modified some of the files, and you don't want to lose your modifications, so do the following:
    a. Make copies of the files in your Eclipse workspace that you need to keep (this will likely be Java source files that you've modified), by finding the Eclipse workspace in Windows Explorer, the Finder, or whatever file browser your system has and copying the files to somewhere *outside your Eclipse workspace*.

b. Inside Eclipse, right-click on your project and select “Delete”. Make sure to check the box that tells Eclipse to “Delete project contents on disk.”

c. Check a fresh copy of your project out from Subversion, either by creating a new project of the “Projects from SVN” type or by checking the project out from SVN using the “SVN Repository Exploring” perspective.

d. Copy the files you backed up from the old project over the files in the new project, by dragging them to the appropriate locations within the project in Eclipse’s Package Explorer.

e. Commit the project to Subversion.