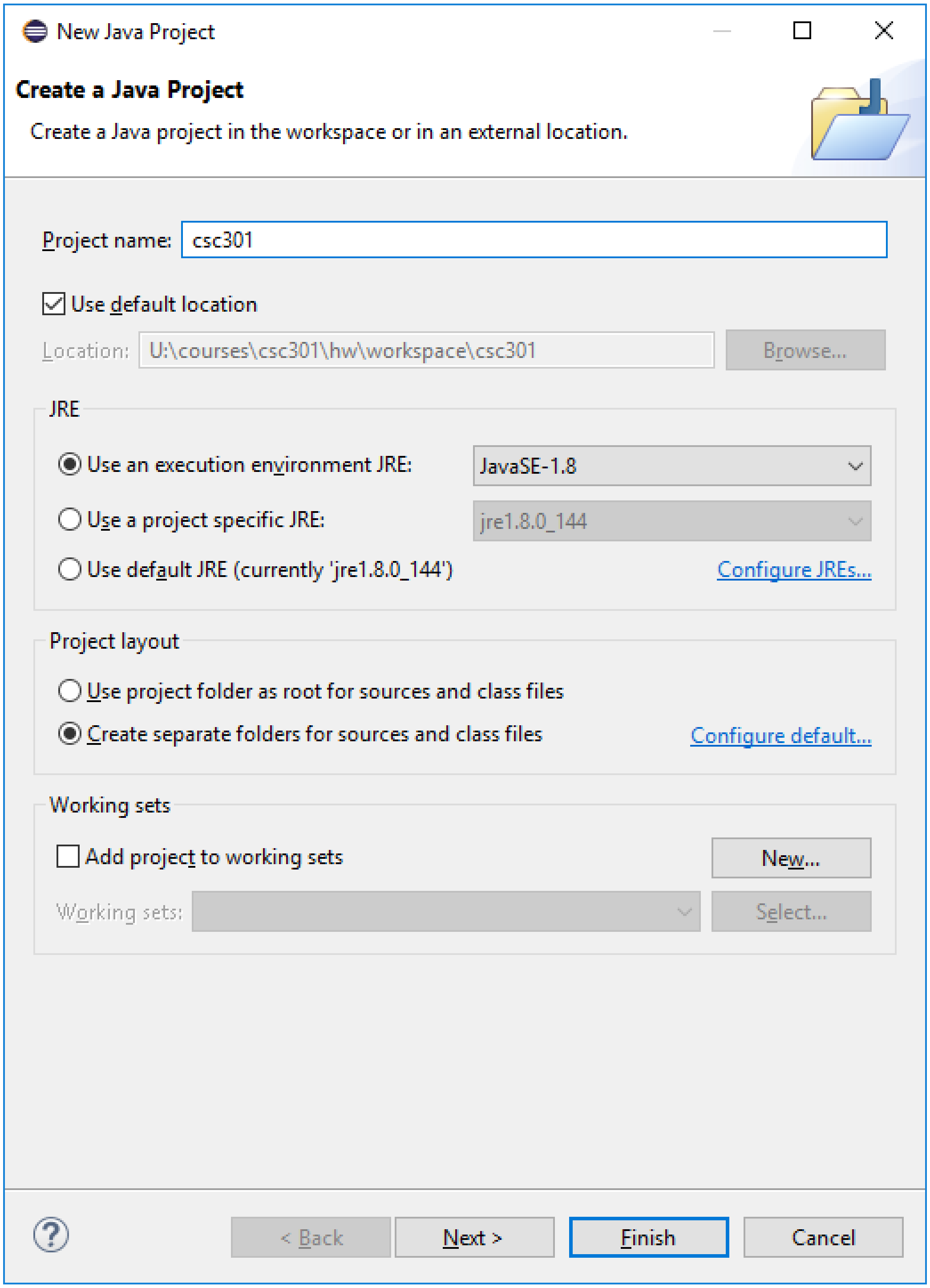
CSC301  HW1

**Instructions:**

1. Download the files SequentialSearchST.java, RSequentialSearchST.java, HW1Test.java, and RHW1Test.java from the dropbox for HW1.
2. Start up Eclipse

3. Create a new Java project called csc301 by navigating to File → New → Java Project from the menu bar and using csc301 as the project name.



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: csc301

☑ Use default location

Location: U:\courses\csc301\hw\workspace\csc301

JRE

● Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE: jre1.8.0_144

○ Use default JRE (currently 'jre1.8.0_144')

Project layout

○ Use project folder as root for sources and class files

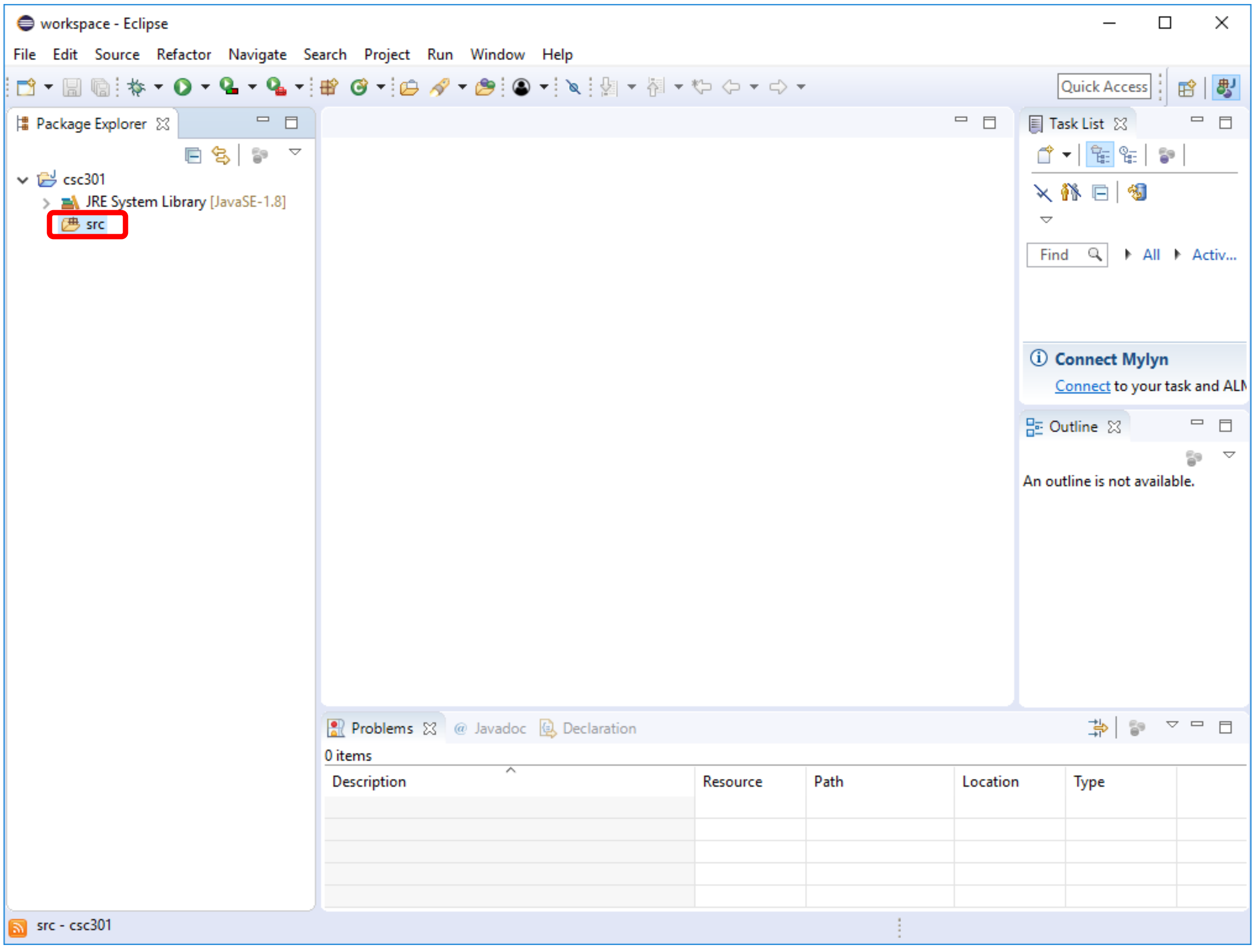
● Create separate folders for sources and class files

Working sets

☐ Add project to working sets

Working sets:

4. Inside the src folder, create a hw1 package.  Do this by right-clicking the src folder in the explorer window and selecting New → Package and using hw1 as the name.



Connect Mylyn

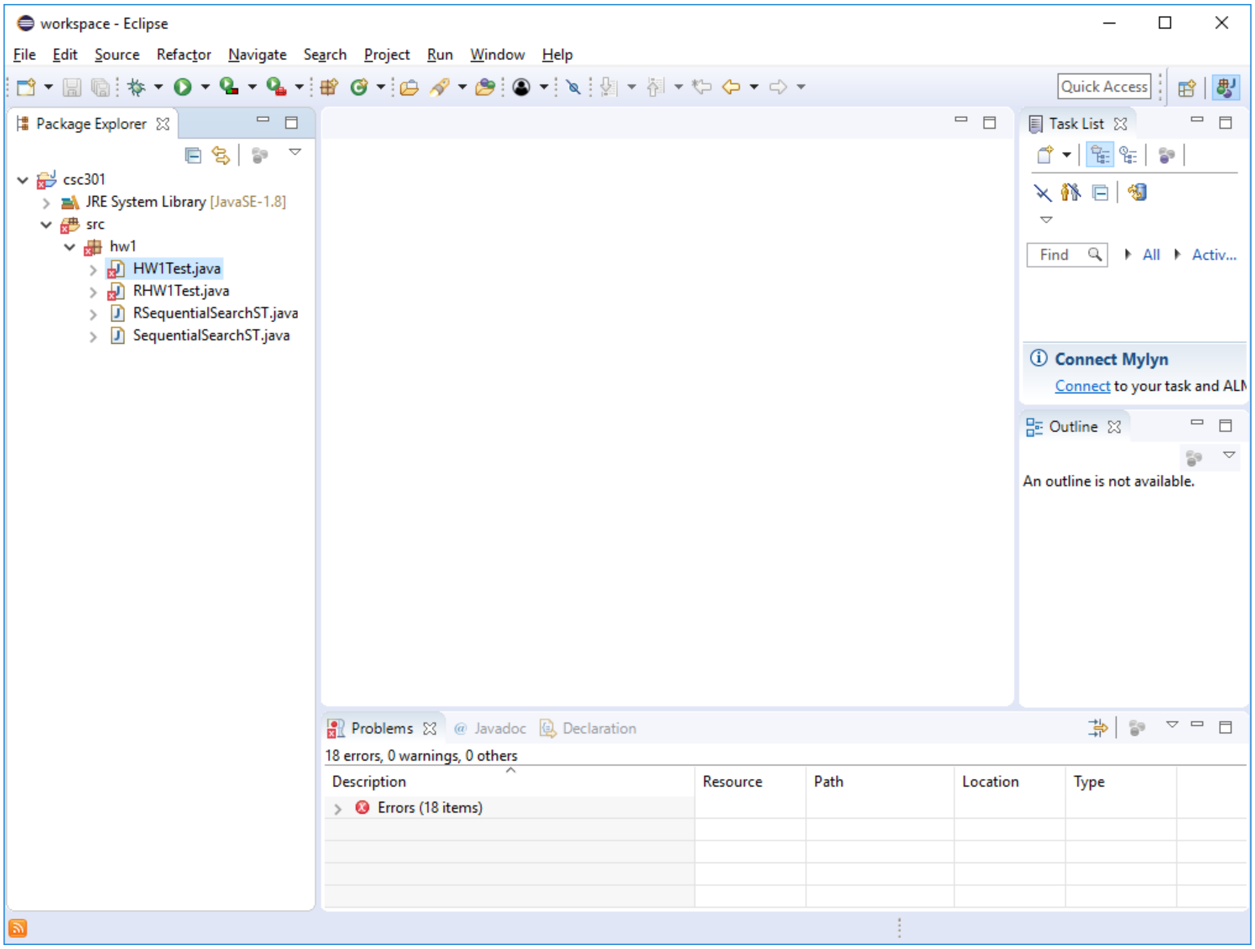
Connect to your task and ALI

An outline is not available.

0 items

| Description | Resource | Path | Location | Type |
|---|---|---|---|---|
|  |  |  |  |  |

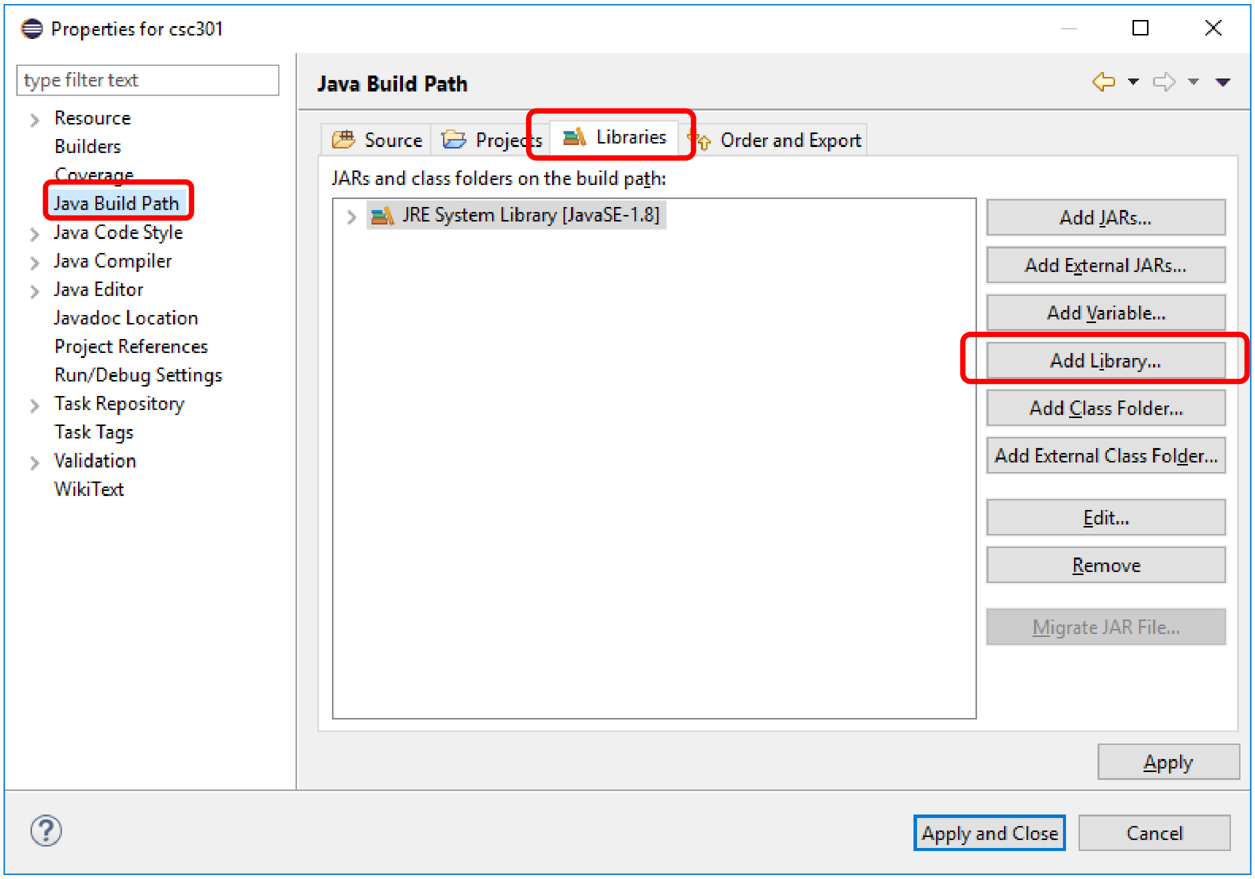
5. Select the files you downloaded and copy and paste them into the hw1 package in Eclipse. In Windows you can select all the downloaded files and hit Ctrl-C to copy. Then click on the hw1 package in the explorer window in Eclipse to select it and hit Ctrl-V to paste. It should now look like this:



An outline is not available.

18 errors, 0 warnings, 0 others

6. Right click the csc301 project and select properties, then Java Build Path and then the Libraries tab.



Properties for csc301

type filter text

- Resource
- Builders
- Coverage
- Java Build Path
- Java Code Style
- Java Compiler
- Java Editor
- Javadoc Location
- Project References
- Run/Debug Settings
- Task Repository
- Task Tags
- Validation
- WikiText

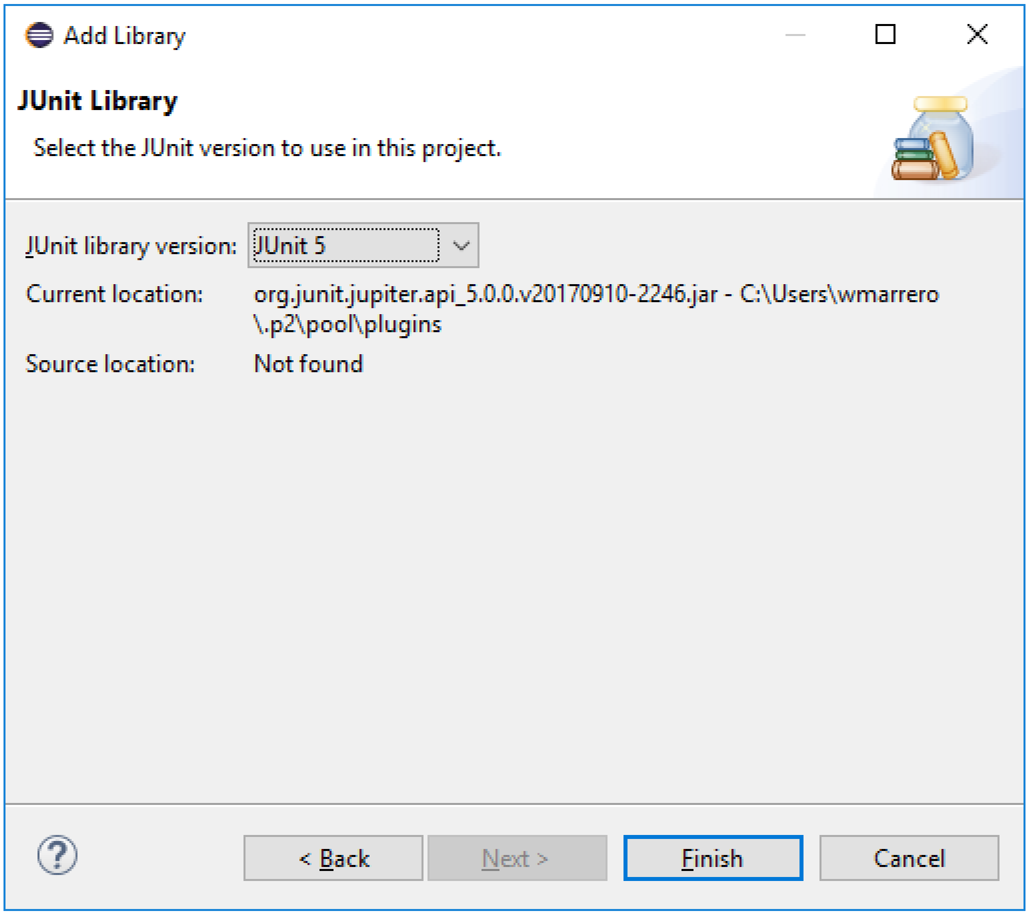
Java Build Path

Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

JRE System Library [JavaSE-1.8]

7. Click Add Library (see previous figure) and select JUnit then JUnit5.



JUnit Library

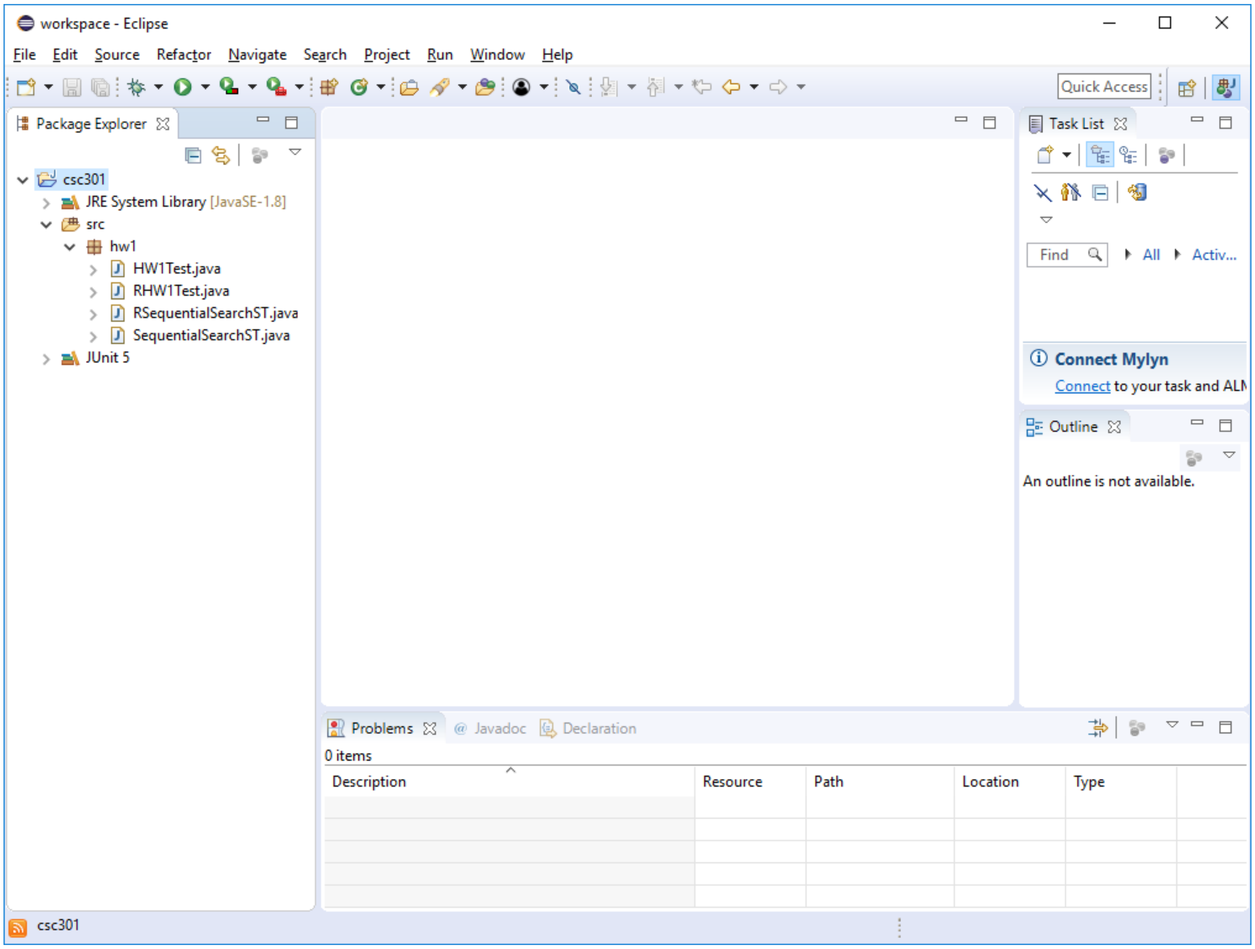
Select the JUnit version to use in this project.

JUnit library version: JUnit 5

Current location: org.junit.jupiter.api_5.0.0.v20170910-2246.jar - C:\Users\wmarrero\.p2\pool\plugins

Source location: Not found

8. The error icons that used to appear by the files HW1Test.java and RHW1Test.java should now be gone and you should now be able to edit the four files you downloaded.



9. Implement the 3 methods, put, get, and delete in SequentialSearchST. **When implementing put, make sure the list is maintained in increasing order of the keys. When implementing get and delete, make sure to take advantage of the fact that the list is in order to avoid looking at all the items in the list. None of the functions should use recursion. You also may not use the keys() method.**
10. In the HW1Test.java file, use the test functions I provided as templates for creating additional test functions for the SequentialSearchST class. In the comments, I indicated some of things you should test. Run your JUnits tests to check your work.
11. Implement the same 3 methods, put, get, and delete in RSequentialSearchST. **When implementing put, make sure the list is maintained in increasing order of the keys. When implementing get and delete, make sure to take advantage of the fact that the list is in order to avoid looking at all the items in the list. All of functions must use recursion. This means each one will need a private helper function that takes a node (the front of the list) as an additional argument. Your code cannot contain any loops except in the keys() method which you are not changing. You also may not use the keys() method.**
12. In the RHW1Test.java file, use the test functions I provided as templates for creating additional test functions for the RSequentialSearchST class. In the comments, I indicated some of things you should

test. (**HINT:** You should be able to use the same exact tests you created in HW1Test, just replace all occurrences of SequentialSearchST with RSequentialSearchST.)

13. Once you are satisfied with your solution, submit all 4 files on D2L in the appropriate dropbox. When you are prompted to drag a file into the dropbox, you should be able to drag it directly from inside Eclipse.
14. Double check your submission. Download it and make sure it is the file you intend to submit!