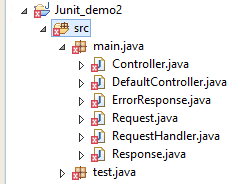
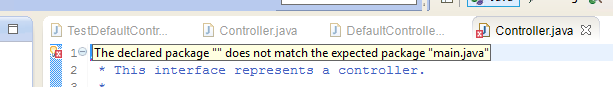
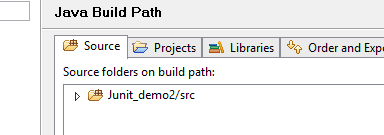
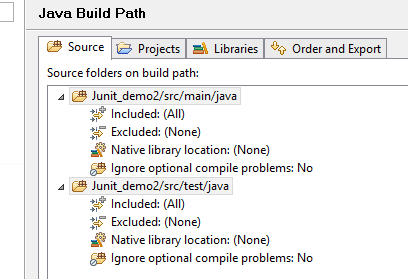
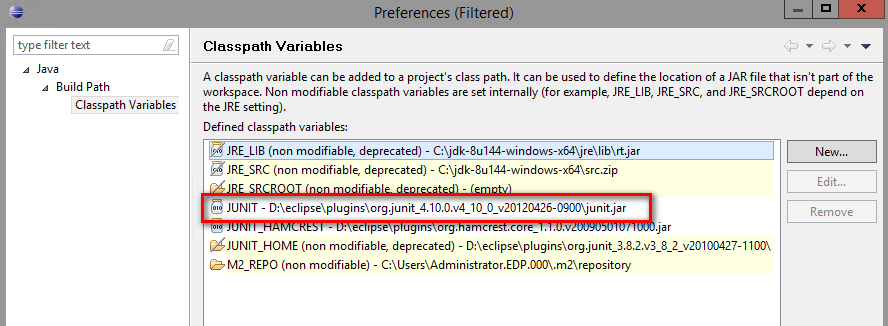
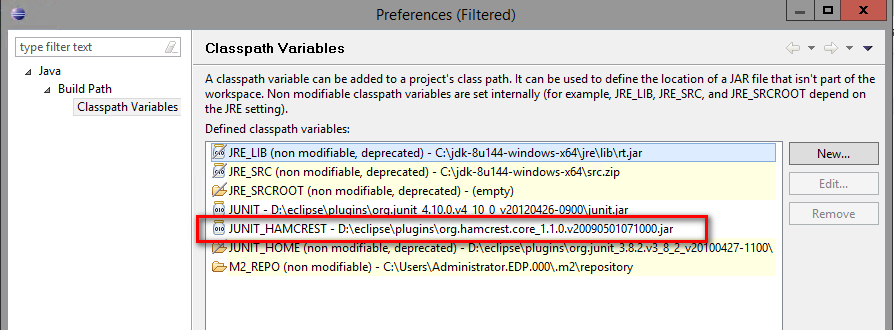
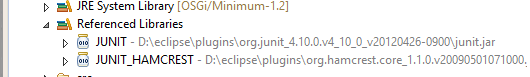
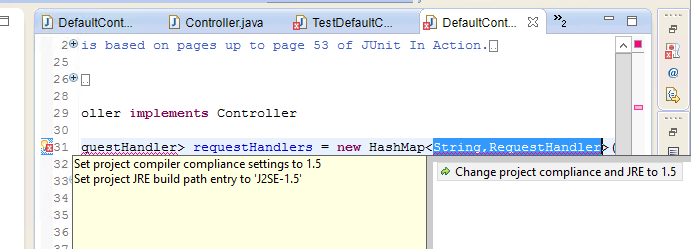
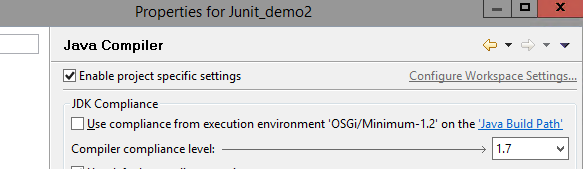
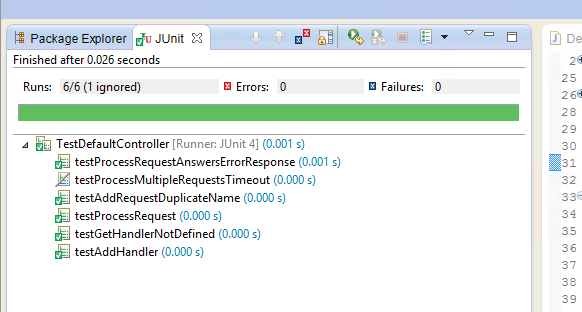
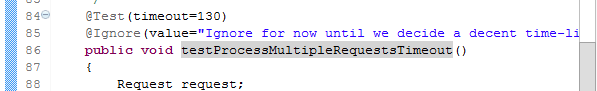
**Running JUnit tests from inside Eclipse**

1. Create a Java project, if don’t already have one.   
File > New > Java Project  
  
2. Copy our src files into the “src” directory of the project, in the user’s “workspace” directory.  
  
3. Right-click on the “src” folder in Project Explorer and select “refresh”.  
The source files should now be listed:  
  
**NOTE:** Eclipse reports the following error for each of my source files:  
  
This happens because the directory structure where I am keeping my files does not match my package declarations, and Eclipse is expecting them to match. I’m doing this is because I intentionally want to keep my JUnit sources files away from the main source files. To fix this Eclipse error, we need to go to “Properties > Java Build Path”, select the “Source” tab, remove the existing “src” entry…  
  
..and click the “Add Folder…” button to add two new entries, one for each of my directory trees “src/main/java” and “src/test/java”:  
  
**After a restart of Eclipse,** these errors have now gone.  
But we still have errors because we need to add the JUnit jar files to our project.  
  
  
4. Add the JUnit.jar and hamcrest.jar files to our project.  
  
**NOTE:** Because these files are located in the Eclipse “plugins” directory, we will add these as classpath variables.

- Right-click on the project name and select “Properties” from the context menu.   
- In the Properties dialog that appears, select “Java Build Path” and select the “Libraries” tab.  
- Click “Add Variable…”.  
- Click “Configure Variables…”.  
- Add a “JUNIT” variable that points to the jar file:  
  
- Also add a “JUNIT\_HAMCREST” variable that points to the hamcrest.jar file:  
  
- Back on the “Libraries” tab, click “Add Variable…” and double-click each of the new variables to add them to the project. The two libraries will now be listed by the Package Explorer:  
  
  
  
5. Changing the Java level used by the project.   
If you see errors like this…  
  
…then right-click on the project name, select “Properties”, select “Java Compiler”, uncheck “Use compliance from execution environment…”, and change the compliance level to a more recent Java version:  
  
  
  
6. Run the JUnit tests.  
Method 1: Right-click on each JUnit source file and select “Run As > JUnit Test”  
Method 2: Right-click on the project name and select “Run As > JUnit Test”  
  
The results should look like this:  
  
  
**NOTE:** That second test was disabled intentionally by me, using the JUnit “@Ignore”:  
(This is a better method than commenting-out the “@Test” annotation, which had to be done before JUnit v4 to disable a test).  
  
  
*JeremyC 17-12-2018***END**