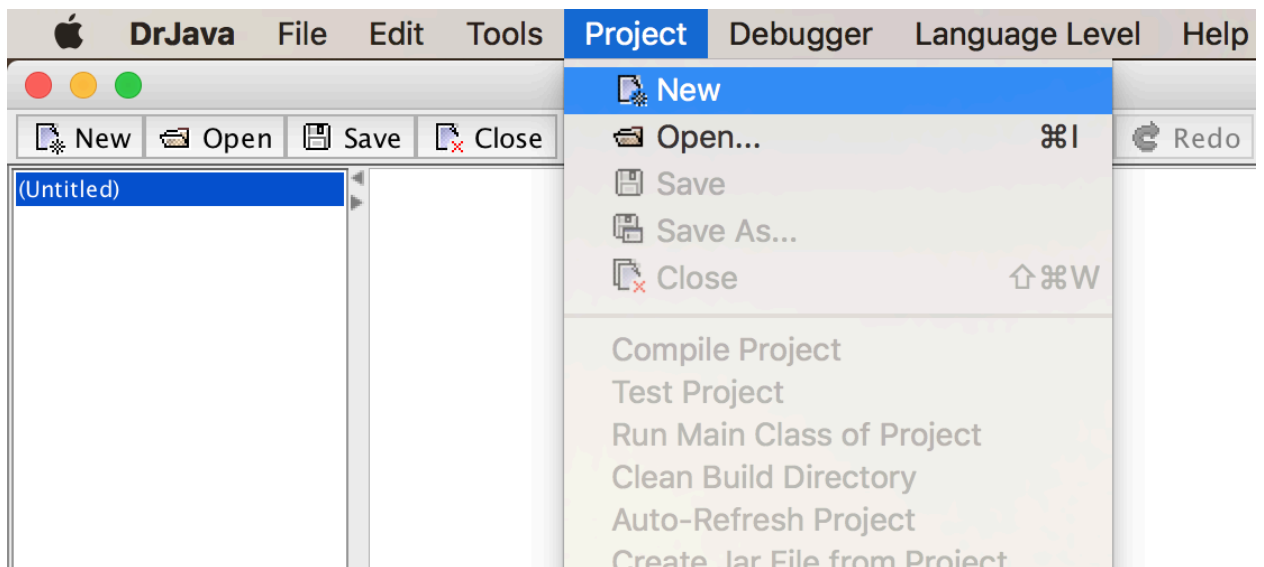


# Getting DrJava installed

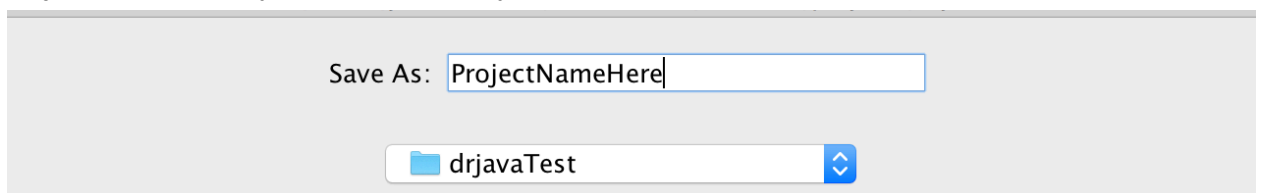
1. Navigate to <http://www.drjava.org>
2. Download either the Windows app or the JAR file. Note that the Mac version will not work for the course and Mac or Linux users will have to use the JAR version.
3. Run the .exe or JAR to launch DrJava. You can run these by simply double clicking on them in File Explorer/Finder.

## Creating a new Java project

1. In order to create a new Java project in DrJava, go to Project > New

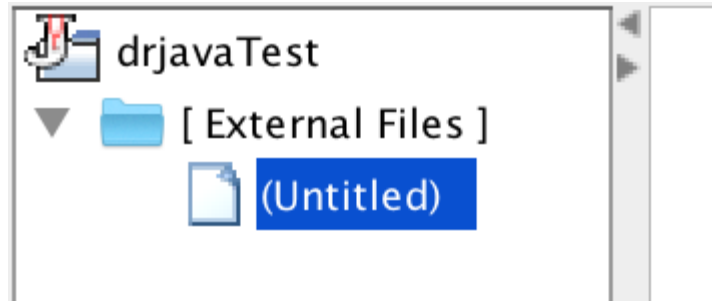


2. On the next screen, you can name your project and select what folder to save your project in. We recommend that you create a folder to keep all of your projects for CS2102 in. We also recommend that each project is a folder within that. Name your project and choose your folder. When you're satisfied, hit Save.



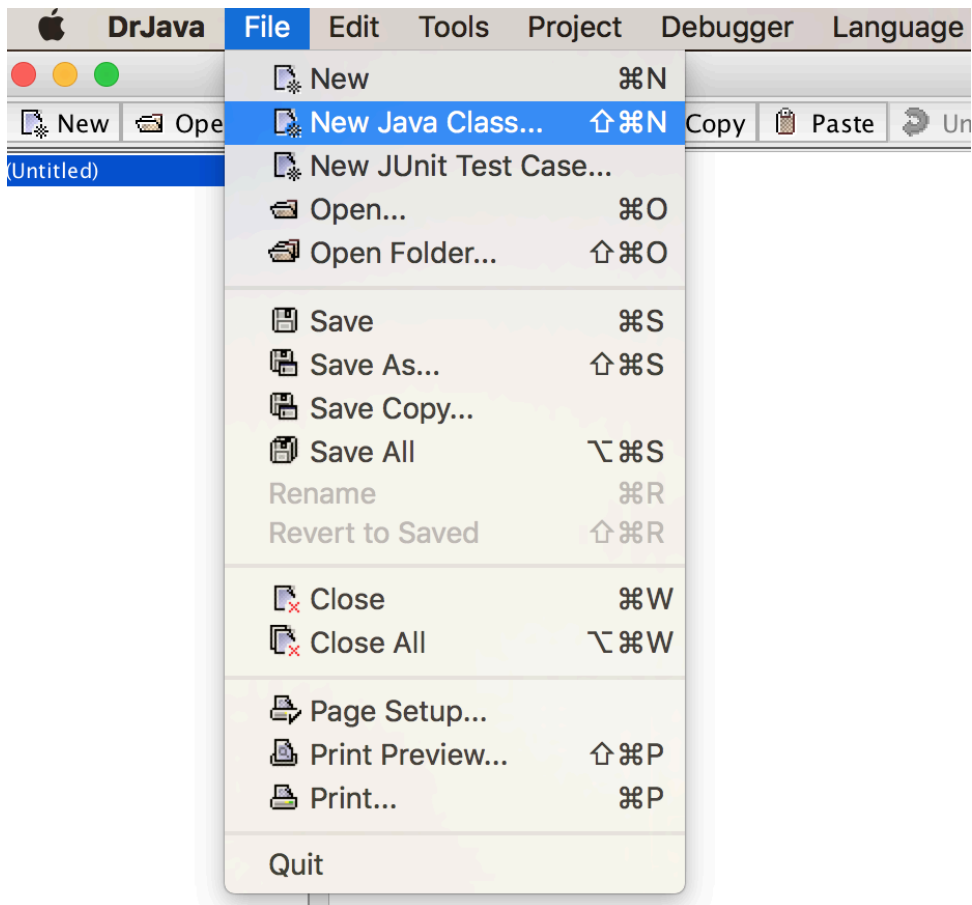
3. On the next screen, just make sure that the Project Root and Working directory are pointing to the folder where you want the project store and hit Ok. You'll notice you now

have a project in the left panel.

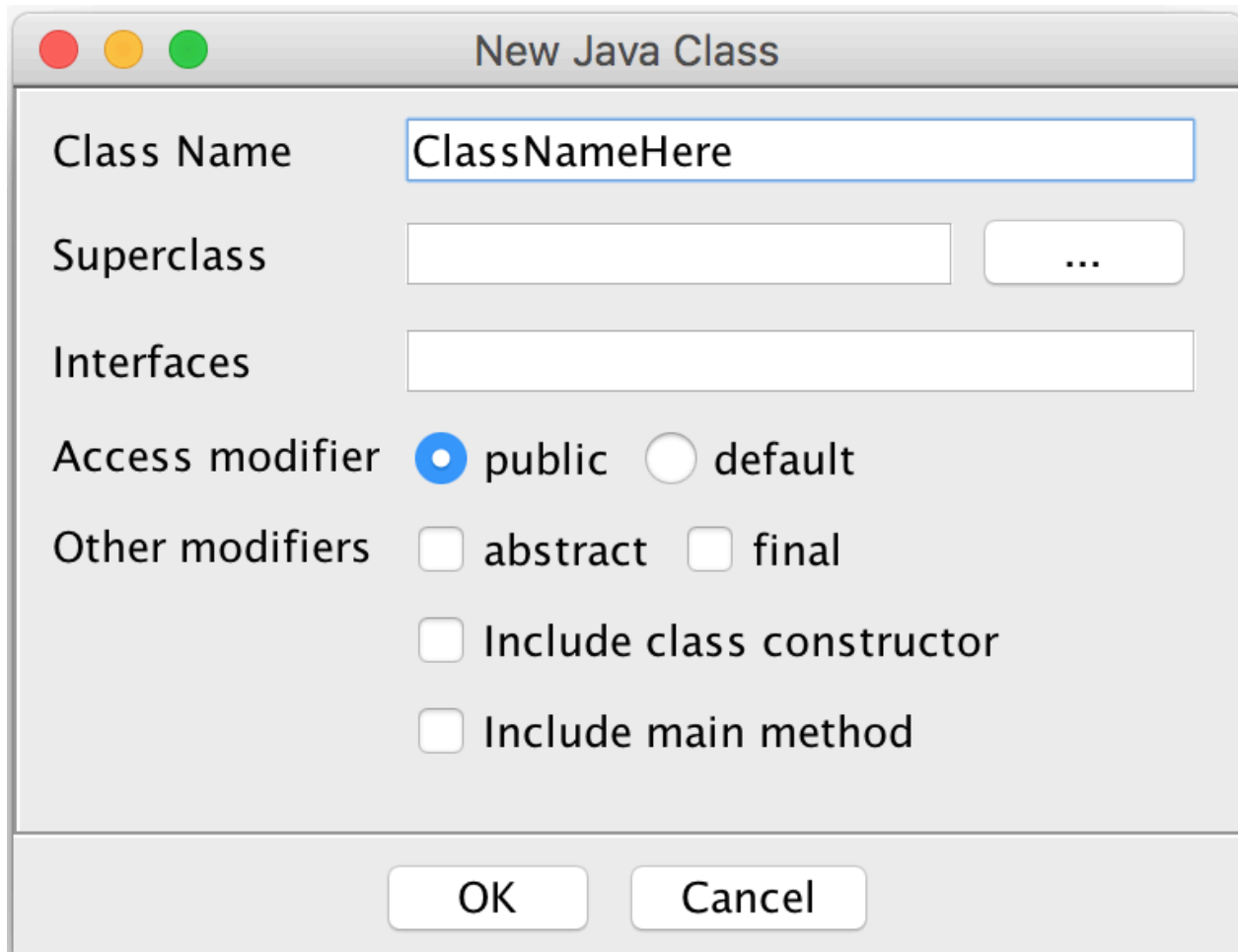


## Creating new classes, interfaces, etc.

1. In order to create a new class, click File > New Java Class



2. In the new window, you can name your class and more. For now we will just name the class. Once you have a name, click OK



The image shows a 'New Java Class' dialog box with a title bar containing three colored buttons (red, yellow, green) and the text 'New Java Class'. The dialog has several fields and options:

- Class Name:** A text field containing 'ClassNameHere'.
- Superclass:** A text field and a button with three dots '...'.
- Interfaces:** A text field.
- Access modifier:** Two radio buttons, 'public' (selected) and 'default'.
- Other modifiers:** Three checkboxes: 'abstract', 'final', and 'Include class constructor'.
- Include main method:** A checkbox.

At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

3. You will have some generate code in the right panel now.

```
/**
 * Auto Generated Java Class.
 */
public class ClassNameHere {

    /* ADD YOUR CODE HERE */
    |
}
```

4. You can now save your file by click File > Save. Name your file the same thing as your class by convention. Remember to keep all your files for a project or assignment within the same directory on your computer.
5. Congratulations, you can now make classes in DrJava.

## Setting up JUnit

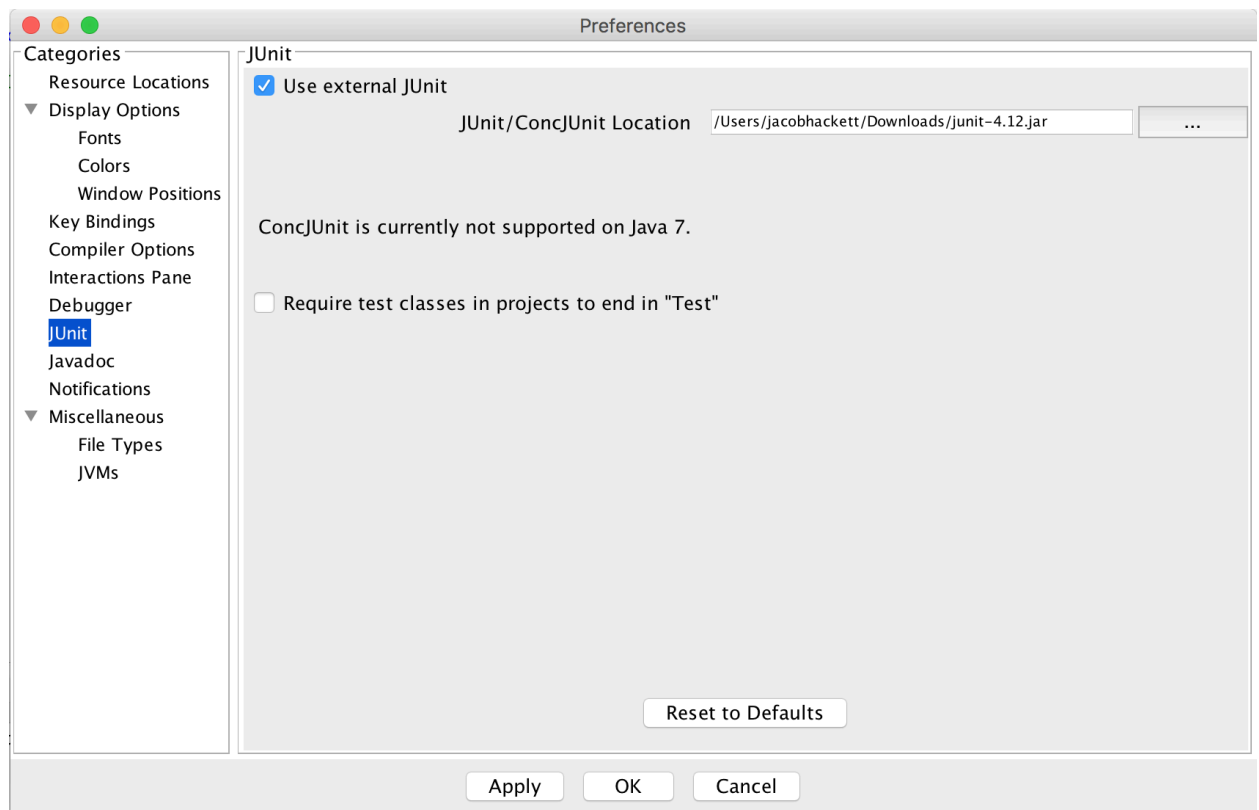
1. JUnit is a popular testing library. It is widely used within industry and the rest of the Java community. The first step to setting up JUnit is to download a JAR file [here](#). That link should be direct downloads recent versions of JUnit.

2. Go to the DrJava > Preferences or Edit > Preferences

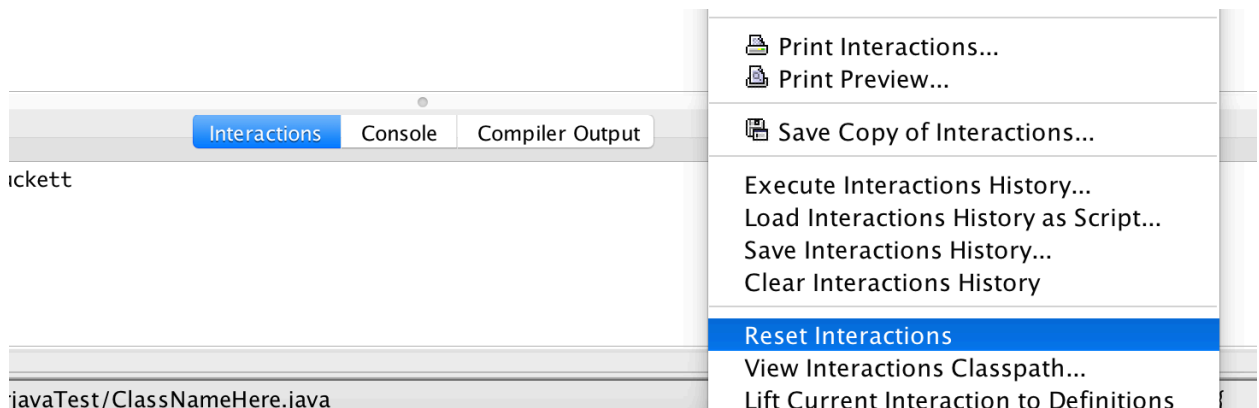


3. Click on JUnit in the left side panel. We need to check the Use external JUnit box and then click on the three dots and find the JUnit JAR we downloaded. Select that JAR and

hit open. Once your screen looks similar to this, click OK.

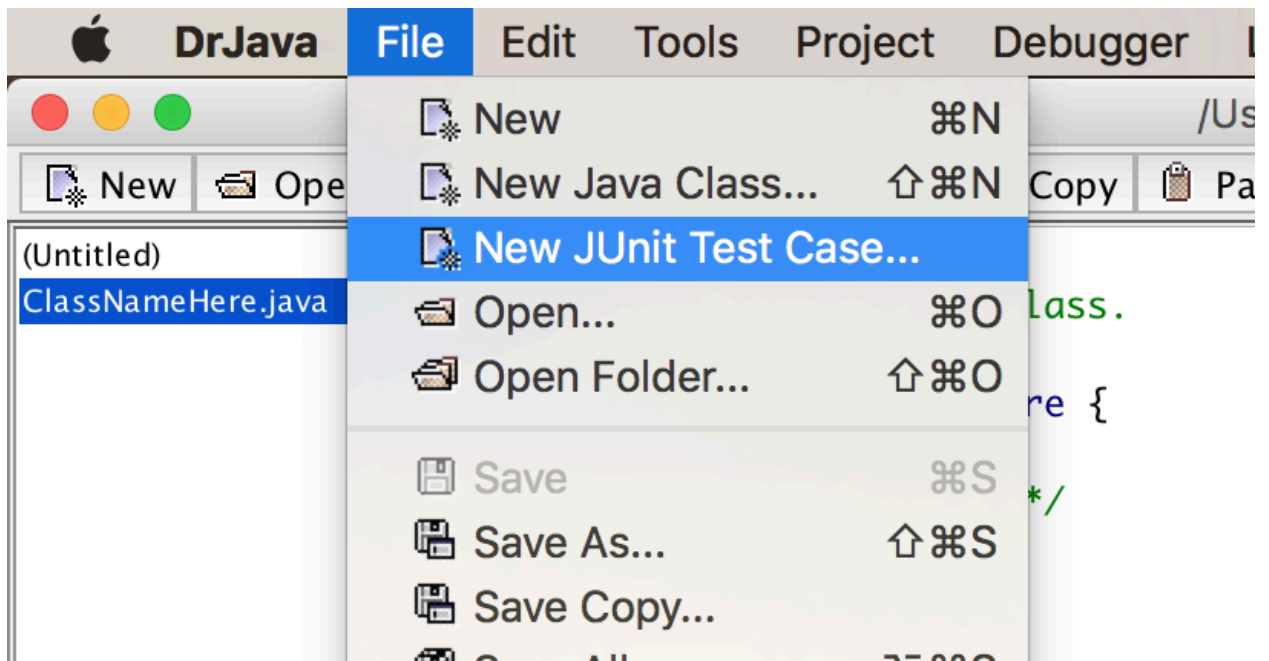


4. DrJava will tell you that you need to reset the interaction panel for this to take effect. In order to do that, just click on interactions above the bottom panel and then right click the bottom panel and click Reset Interactions.

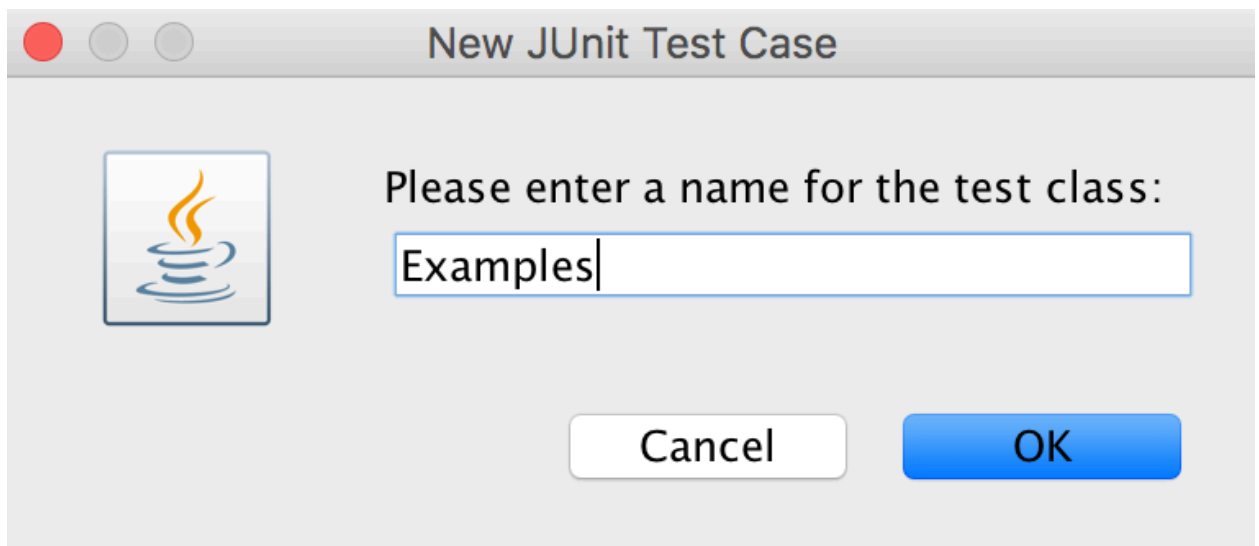


5. Now we have to make our Test file. We're going to call this file Examples.java for all of our projects. We will put all of our tests within Examples. In order to make this, go to File

> New JUnit Test Case.



6. Name your test case Examples and hit OK



7. You should see a new file that DrJava generated. There's a good amount of stuff here but we'll focus on the important stuff for now. Make sure to save this as Examples.java before moving on.
8. You should see a method called testX in the new file. This is an example of a test. Within this we will use JUnit constructs like assertEquals to test that two things are equal and

our program is behaving as intended. For now, let's write a dummy test just to see how it

```
public void testX() {  
    assertEquals(1, 1);  
}
```

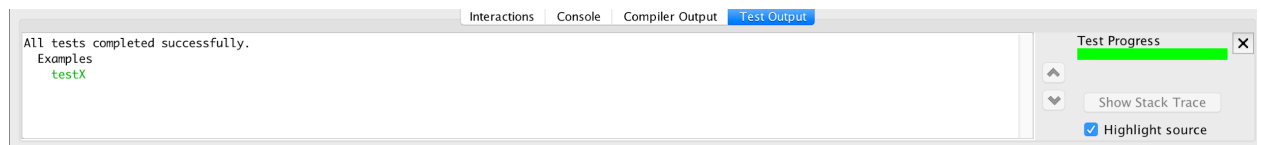
works.

9. This code says to assert that 1 and 1 are equal. We will be testing the output of other methods to make sure they are working. Instead of `assertEquals(1, 1);` we may have `assertEquals(dillo.length, 5);`
10. In order to run our tests, we first need to compile our code. Hit the compile button in the top. Next simply hit the button labeled Test to run all of the tests.

irsework/CS2102/projects/drjavaTest/Examples.java



11. Once you hit Test, you should see the bottom panel change. It should now show your test output. This will contain a summary of all of your tests included which one passed and which failed. Right now we have one test called testX that passed.





12. You can continue to create test cases that follow the pattern of testX within Examples.  
For example:

```
public void testX() {  
    assertEquals(1, 1);  
}
```

```
public void testY() {  
    assertEquals(2, 2);  
}
```

13. Congratulations, you now have JUnit and you can create test cases!