Recitation 2

Testing

Writing Correct Code

Code should implement a specification

Define specification in Javadoc comment

Test code to make sure it is correct

Unit Testing

Break your program up into the smallest testable parts or units. Units should be independent.

These are often one method or a few methods

Test units as you go! Fix bugs before implementing the next unit

JUnit

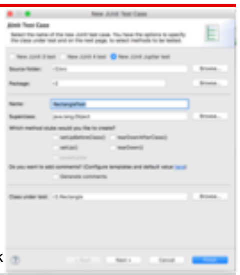
A framework for writing tests for Java programs

Use JUnit to create a Test Class

Use Eclipse GUI to run the tests

To Create a JUnit Test Class

1. Select the src folder for the project in the Package Explorer pane.
2. Use menu item File --> New --> JUnit Testing Class.
3. The window that opens looks like the image to the right. Type in a name for the class, something like R3Test, if the class whose methods are to be tested is named R3
4. Click Finish.
5. If a window opens that says, "JUnit 5 is not on build path. Do you want to add it?", then click the *OK* button. Library JUnit 5 *must* be on the build path for the JUnit testing class to work



JUnit demo

How to run tests

How to tell whether a test failed

How to check that all test ran & completed

White-box testing

Test each statement of a unit
Test each branch of a unit
Test each expression thoroughly
Test extreme or corner cases

Programming Assignment

Consider buggy class Rectangle

Identify the units you should test

Write white-box test-cases to thoroughly test (and fix) this class following best practices