Testing with Junit Lab 1

- Getting started with JUnit

Testing with Junit Lab 1

- Create a new project called **Lab1 JUnit** and click Next

- Right click on the new project file, hover over Build Path
  - Press **Add Library...** and choose JUnit; click Next
  - Choose JUnit 5 and click Finish
  - Now our project has JUnit on its classpath

- Hovering over Build Path again
  - Click **Create new source folder**, and name it **test**
  - Click Finish

- Click Finish to complete the process

Testing with Junit Lab 1

- Right-click on the ***test*** source folder and choose **New Package**
  - Name it `com.testing.lang`

- Right-click on the package and choose **New JUnit Test Case**
  - Name the test class `StringTest`
  - Leave checkboxes unchecked
  - Leave "Class under test" field blank since we are testing String

- Click Finish

Testing with Junit Lab 1

- Write a `testLength` method
  - What are the requirements for a test method?
    - @Test
    - Remember signature and return type requirements

- Test `String.length()`
  - Create a new `String` object (you pick the value)
  - Write an `assertTrue()` that compares the expected length of the string to the result of `String.length()`

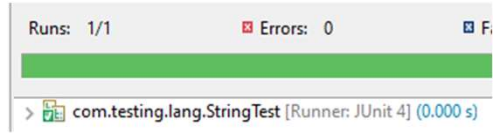
```
@Test
public void testLength(){
    String s = "JUnit Rules";
    assertTrue(11 == s.length());
}
```

Testing with Junit Lab 1

- Right-click on the test class in Package Explorer and choose **Run As JUnit Test**

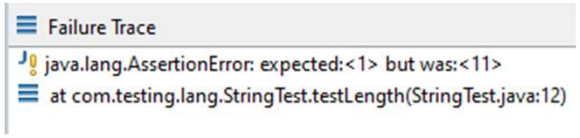
  OR

- Right-click in the editor itself and do the same thing



- Make it artificially fail
  - Make assertion incorrect
  - Rerun test

- Examine the reason for failure



Testing with Junit Lab 1(end of lab)

- Write a test method for `String.substring()`
  - Follow the same steps we did with the first one
  - Again, after you see it pass, make it fail and look at the reason

- Print a message to the console in each test method
  - Notice the execution order
  - Is it always the same?
  - Switch the order of the test methods in the class and repeat
    - What do you notice?

- Remember, test execution order cannot be guaranteed or relied upon
  - If you see a pattern, it may not happen like that every time