JUnit Testing with Eclipse

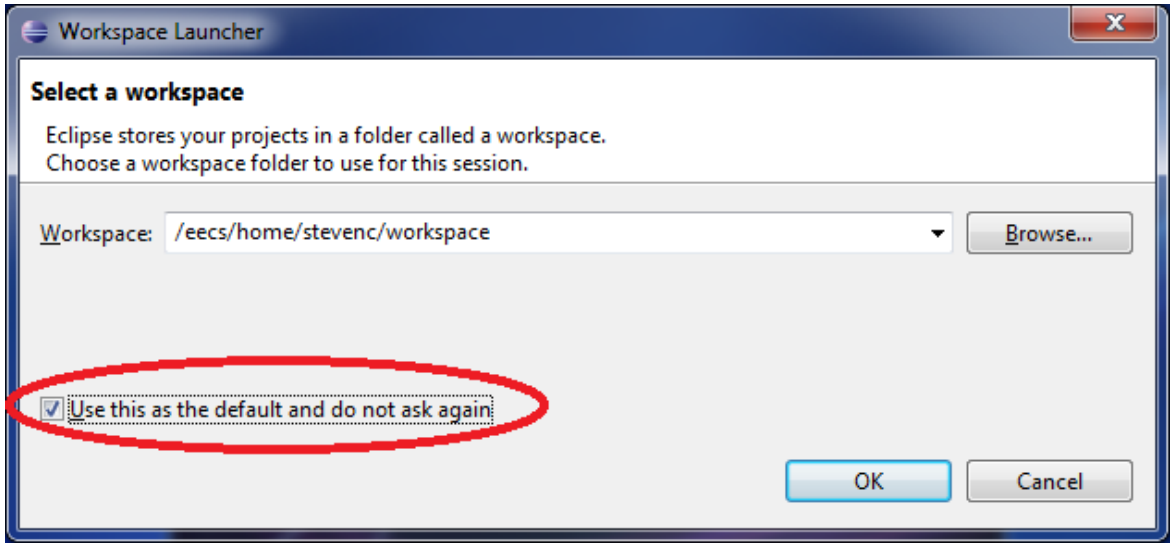
Plan

- Write a code to output body mass index (BMI)
- Perform JUnit testing
- Use type.lib.ToolBox.getBMI as oracle
- Use type.lib.ToolBox.launch to run your code

Start Eclipse

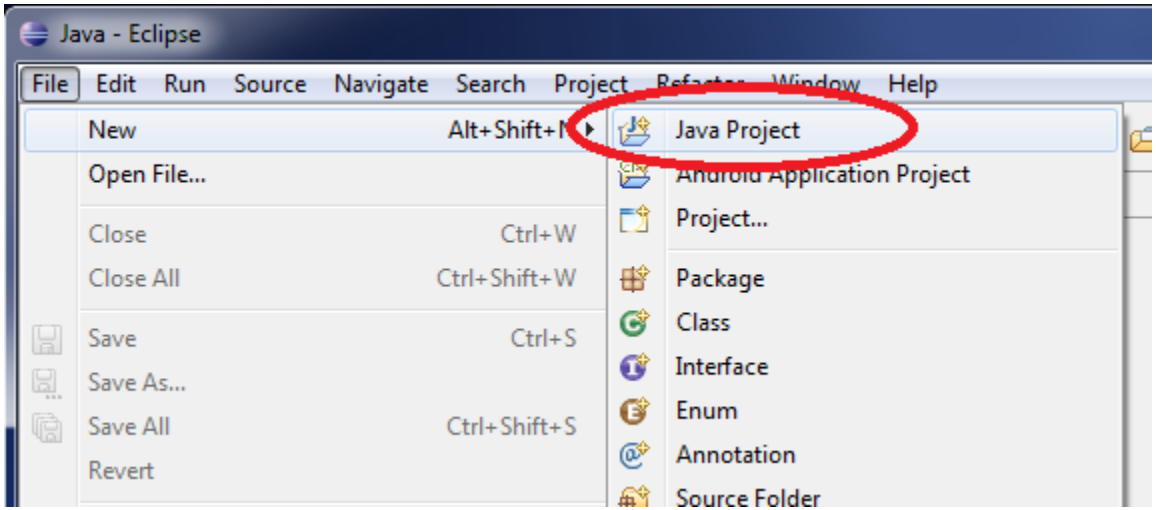
- EECS Lab:
  - Type “eclipse &” at terminal
  - Select CSE Menu > Development > Eclipse
- Personal computer:
  - Install Eclipse IDE for Java Developers
  - Double-click program icon in installation folder
- Use default “workspace” and click OK
- Close the “Welcome” tab

Start Eclipse



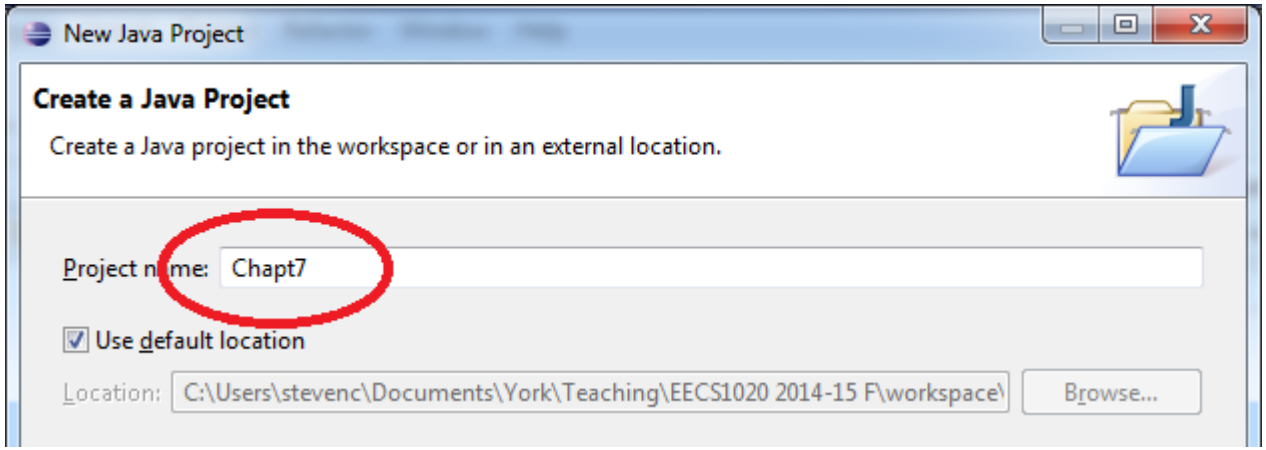
Create a Project

- Must be done before creating a class
- Select File > New > Java Project



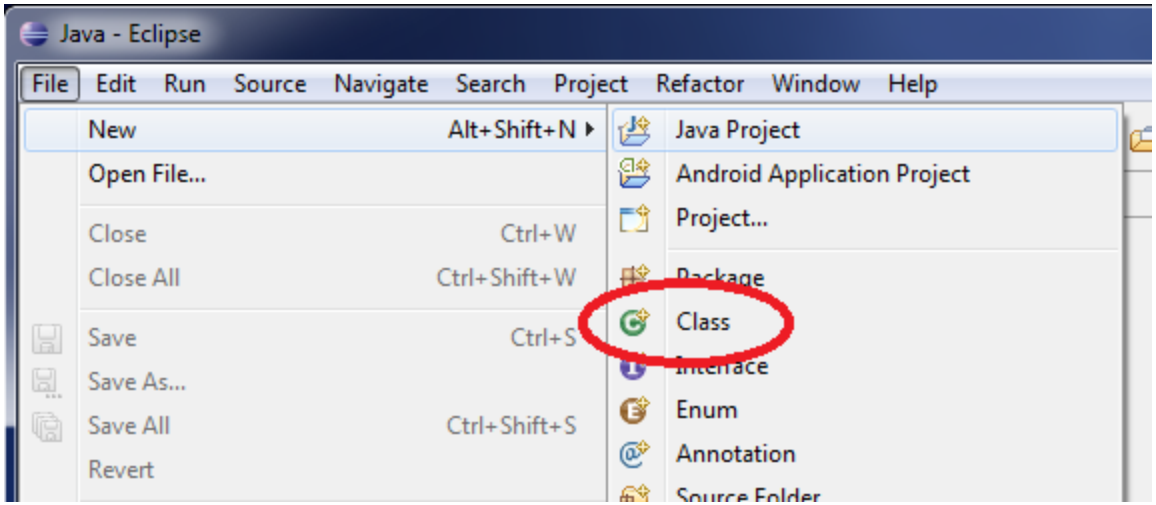
Create a Project

- Type “Chap7” as the project name, then click Finish (ignore all other options)



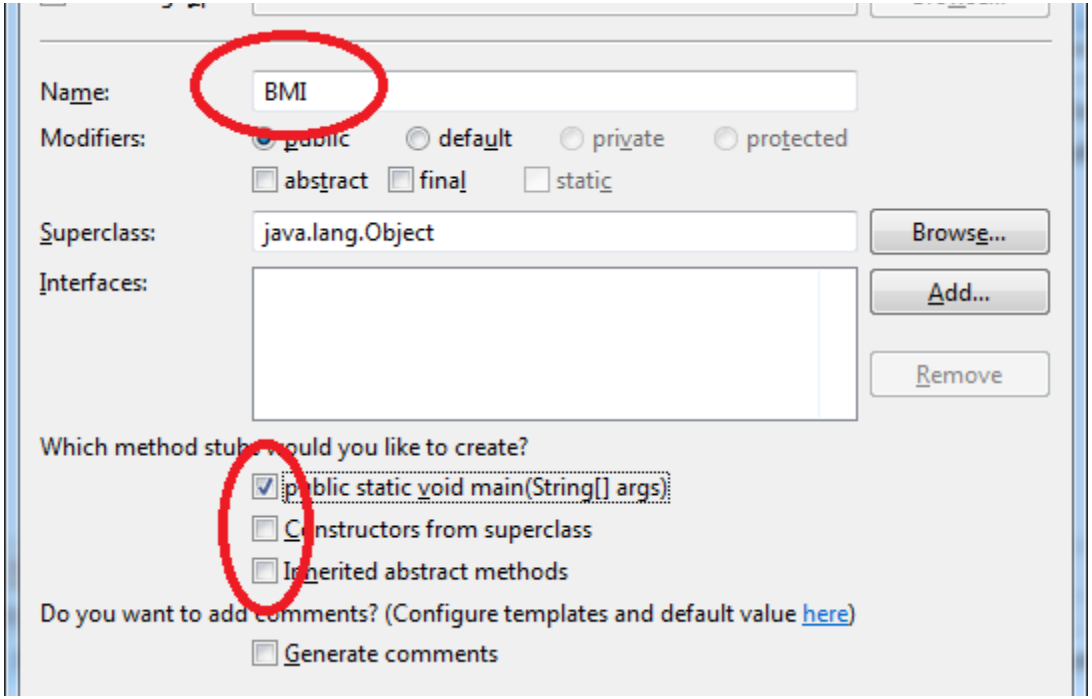
Create a Class

- Select File > New > Class



Create a Class

- Type “BMI” as the name and enable the main method stub



Calculate BMI

- Prompt for:
  - Weight in lbs
  - Height in feet'inches (convert to inches)

- Calculation:

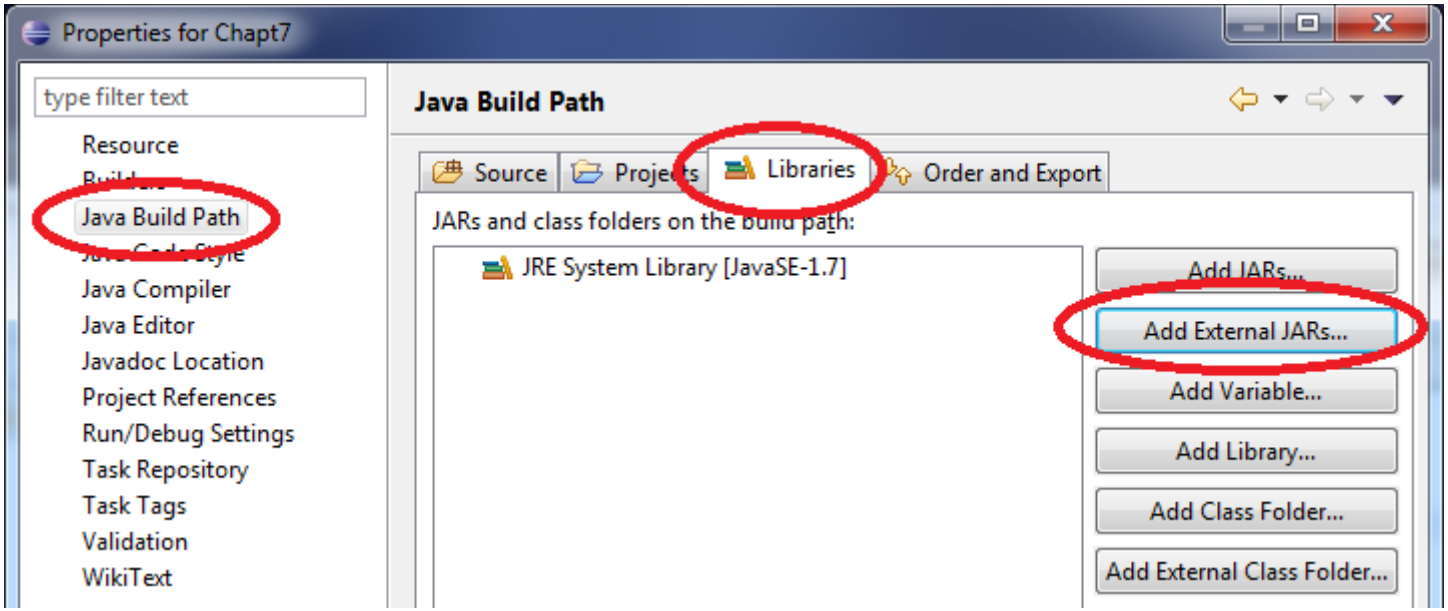
$$\frac{weight}{inches^2} \times 703$$

Calculate BMI

```
final int INCHES_PER_FOOT = 12;
final int BMI_IMPERIAL_CONV = 703;
// Get input
output.print("Enter weight in lbs. ");
double weight = input.nextDouble();
output.print("Enter height in ft'in. ");
String height = input.next();
// Calculate BMI
String h1 = height.substring(0, 1); // "feet" part of height
String h2 = height.substring(2); // "inches" part of height
int inches = Integer.parseInt(h1) * INCHES_PER_FOOT +
        Integer.parseInt(h2);
double bmi = weight / (inches * inches) * BMI_IMPERIAL_CONV;
            // Output
            output.printf("BMI: %.2f%n", bmi);
```

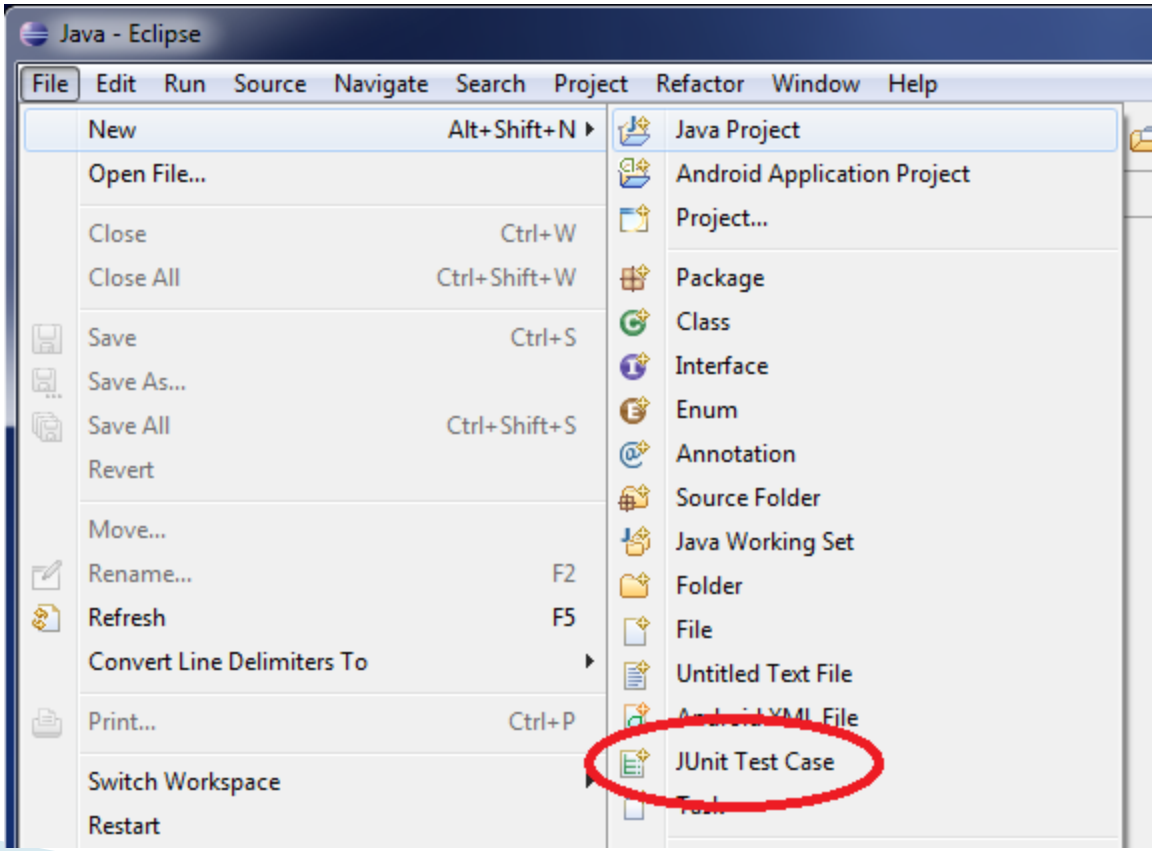
Import Type

- Download type.jar to your directory
- Add it to the project to help with testing
  - Project > Properties > Java Build Path > Libraries



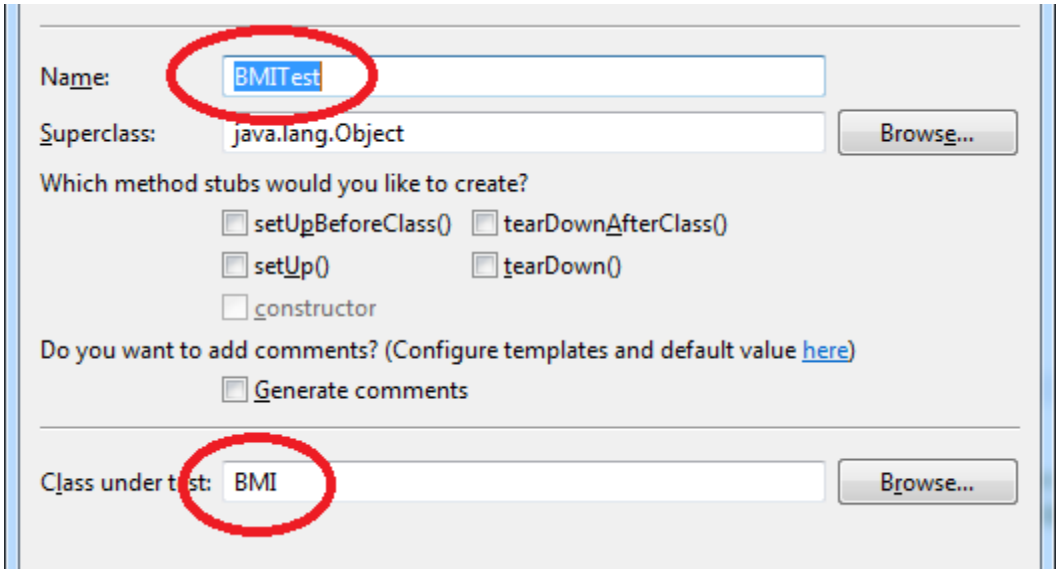
Create a JUnit Test

- Select File > New > JUnit Test Case



Create a JUnit Test

- Enter the name “BMITest” and ensure that class under test is “BMI”
- Click “OK” to add JUnit to the build path

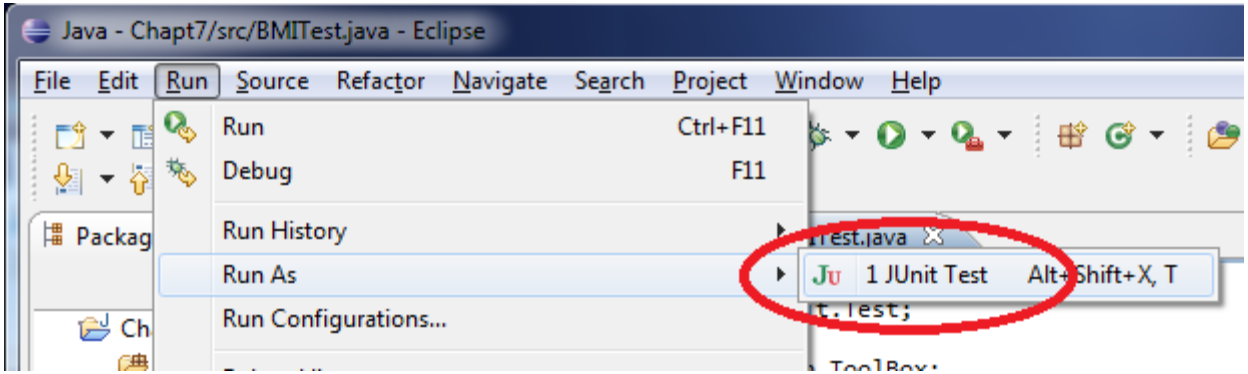


Create a JUnit Test

```
@Test
public void test()
{
    final double WEIGHT = 160;
    final String HEIGHT = "5'9";
    double oracle = ToolBox.getBMI(WEIGHT, HEIGHT);
    String expected = "Enter weight in lbs. " +
            "Enter height in ft'in. " +
    String.format("BMI: %.2f%n", oracle);
    String actual = ToolBox.launch("BMI",
            "" + WEIGHT + "\n" + HEIGHT + "\n");
    assertEquals(expected, actual);
}
```

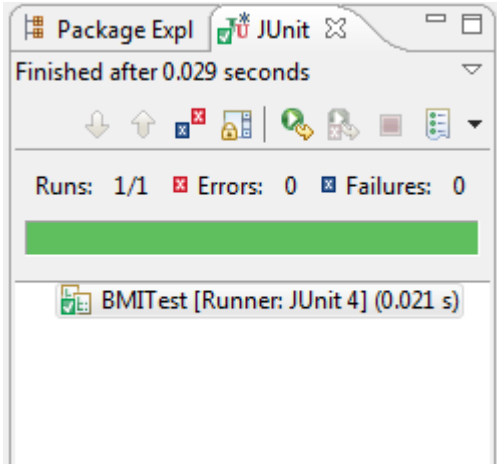
Run a JUnit Test

- Ensure the BMITest.java tab is selected and select Run > Run As > JUnit Test



Pass a JUnit Test

- If the test passes, then you should see a green bar in the JUnit pane



Fail a JUnit Test

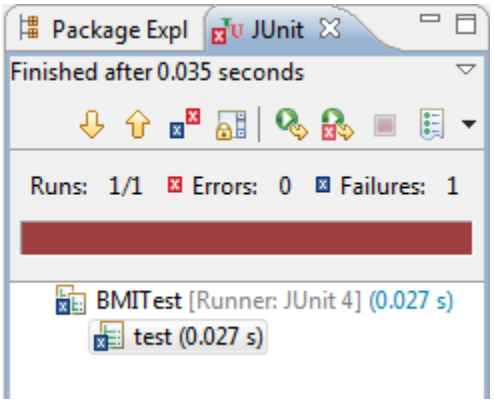
- Change the following lines in BMI.java:

```
output.print("Enter weight in lbs. ");
double weight = input.nextDouble();
output.print("Enter hieght in ft'in.");
```

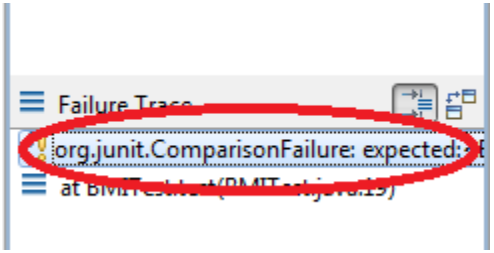
- Notice the missing space and spelling mistake

Fail a JUnit Test

- Run the test again and a red bar will appear

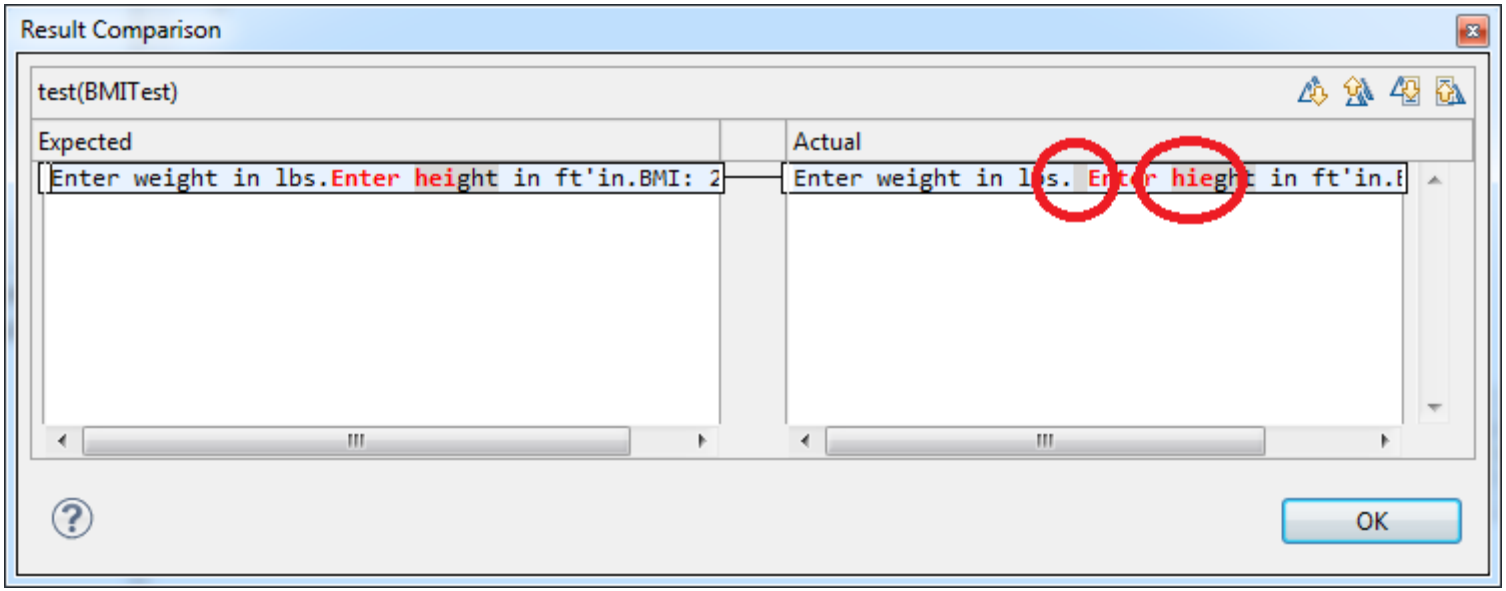


- Double click on the test in the Failure Trace



Fail a JUnit Test

- The expected and actual output are shown, with differences highlighted in red



In Practice

- Multiple tests would be created, each testing a different aspect of the program
- Discrepancies would be fixed and all the tests would be re-run
- Continue until all tests pass