

CMSC 131: Chapter 17 (Supplement)

JUnit II

Creating JUnit.java File

- So far you have run JUnit tests. Let's now describe how to write them.
- Create a JUnit .java test file as follows:
 - File→New→Other→Java→JUnit→JUnit Test Case
 - If the JUnit library is not part of the project then select yes when asked: "The JUnit library 'junit.jar' is not on the build path. Do you want to add it?"
 - Pick a class name (e.g., Myclass or any name you prefer)
 - At this point you will see a class declaration that looks as follows:

```
public class Myclass extends TestCase { }
```
- The class declaration defined when we create a JUnit test file represents an inheritance example. TestCase is a class that defines methods supporting the implementation of JUnit tests, and those methods are now available in Myclass.

Writing JUnit Tests

- To create tests define public void methods that start with the word "test" (e.g., testEqualsMethod, testNumberOne, testNumberTwo, etc.). Each of these methods represents a test.
- Inside of a test method you can have typical Java code where assertions are used to specify the expected results from the test.
- The following are common assertions we use in JUnit tests:
 - assertTrue - verifies that the argument expression is true. If the argument expression is false the test fails. Otherwise execution is successful.
 - assertEquals - takes two arguments (expected value and actual value). If the values are not equal the test fails. Otherwise execution is successful.
- You can define auxiliary methods (e.g. private methods) that support the set of tests you are developing.
- You can have multiple assertions in a JUnit test. Once the first one fails the whole test is considered to have failed.
- Pick descriptive names for your test methods.

Writing JUnit Tests (Example)

```
public class AuxMath {
    public static int maximum(int x, int y) {
        if (x > y)
            return x;
        return y;
    }

    public static int minimum(int x, int y) {
        if (x < y)
            return x;
        return y;
    }
}

import junit.framework.TestCase;

public class JUnitTestExample extends TestCase {
    public void testOneMaximum() {
        int expectedResults = 20;
        assertEquals(expectedResults,
            AuxMath.maximum(10, 20));
    }

    public void testTwoMinimum() {
        int expectedResults = 5;
        assertEquals(expectedResults,
            AuxMath.minimum(30, 5));
    }
}
```