JUnit Setup and Testing Exercises

(Week 2)

# Exercise 1: Setting Up JUnit Without Maven

This section guides you through setting up JUnit in Visual Studio Code without Maven.

## 1. Folder Structure

JUnitExample/  
├── lib/ ← JUnit and Hamcrest JARs  
│ ├── junit-4.13.2.jar  
│ └── hamcrest-core-1.3.jar  
└── src/  
 ├── Calculator.java  
 └── CalculatorTest.java

## 2. Java Class: Calculator.java

public class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
}

## 3. Test Class: CalculatorTest.java

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class CalculatorTest {  
 @Test  
 public void testAdd() {  
 Calculator calc = new Calculator();  
 assertEquals(15, calc.add(10, 5));  
 }  
}

## 4. Compile and Run Commands

cd src  
javac -cp "../lib/junit-4.13.2.jar;../lib/hamcrest-core-1.3.jar" \*.java  
java -cp ".;../lib/junit-4.13.2.jar;../lib/hamcrest-core-1.3.jar" org.junit.runner.JUnitCore CalculatorTest

## Output

JUnit version 4.13.2  
.  
Time: 0.002  
  
OK (1 test)

# Exercise 2: Assertions in JUnit

Demonstrating the use of various assertions available in JUnit.

## AssertionsTest.java

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

    @Test

    public void testAssertions() {

        // Assert equals

        assertEquals("Sum should be 5", 5, 2 + 3);

        // Assert true

        assertTrue("5 is greater than 3", 5 > 3);

        // Assert false

        assertFalse("5 is not less than 3", 5 < 3);

        // Assert null

        Object obj1 = null;

        assertNull("Object should be null", obj1);

        // Assert not null

        Object obj2 = new Object();

        assertNotNull("Object should not be null", obj2);

    }

}

## Compile and Run Commands

javac -cp "../lib/junit-4.13.2.jar;../lib/hamcrest-core-1.3.jar" AssertionsTest.java  
java -cp ".;../lib/junit-4.13.2.jar;../lib/hamcrest-core-1.3.jar" org.junit.runner.JUnitCore AssertionsTest

## Output

JUnit version 4.13.2  
.  
Time: 0.001  
  
OK (1 test)

# Exercise 3: Arrange-Act-Assert with Setup and Teardown

Tests organized using the AAA pattern with @Before and @After annotations.

## CalcAAA.java

public class CalcAAA {  
 public int add(int a, int b) {  
 return a + b;  
 }  
 public int divide(int a, int b) {  
 return a / b;  
 }  
}

## CalcAAATest.java

import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class CalcAAATest {  
 private CalcAAA calculator;  
  
 @Before  
 public void setUp() {  
 calculator = new CalcAAA();  
 System.out.println("🔧 Setup: New CalcAAA instance created");  
 }  
  
 @After  
 public void tearDown() {  
 calculator = null;  
 System.out.println("🧹 Teardown: CalcAAA instance cleared");  
 }  
  
 @Test  
 public void testAddition() {  
 int result = calculator.add(10, 5);  
 assertEquals("Addition result should be 15", 15, result);  
 }  
  
 @Test  
 public void testDivision() {  
 int result = calculator.divide(20, 4);  
 assertEquals("Division result should be 5", 5, result);  
 }  
  
 @Test(expected = ArithmeticException.class)  
 public void testDivisionByZero() {  
 calculator.divide(10, 0);  
 }  
}

## Compile and Run Commands

javac -cp "../lib/junit-4.13.2.jar;../lib/hamcrest-core-1.3.jar" \*.java  
java -cp ".;../lib/junit-4.13.2.jar;../lib/hamcrest-core-1.3.jar" org.junit.runner.JUnitCore CalcAAATest

## Output

🔧 Setup: New CalcAAA instance created  
🧹 Teardown: CalcAAA instance cleared  
🔧 Setup: New CalcAAA instance created  
🧹 Teardown: CalcAAA instance cleared  
🔧 Setup: New CalcAAA instance created  
🧹 Teardown: CalcAAA instance cleared  
  
OK (3 tests)