Exercise 1: Setting Up Junit

<dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.13.2</version>  
 <scope>test</scope>  
 </dependency>

package org.example;  
  
import org.junit.Test;  
import static org.junit.Assert.*assertEquals*;  
  
public class CalculatorTest {  
  
 @Test  
 public void testAdd() {  
 Calculator calc = new Calculator();  
 int result = calc.add(2, 3);  
 *assertEquals*(5, result); // Passes if result is 5  
 }  
}

Exercise 3: Assertions in Junit

package org.example;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
public class AssertionTest {  
 @Test  
 public void testAssertions() {  
 //Assert equals  
 *assertEquals*(5, 2 + 3);  
 //Assert true  
 *assertTrue*(5 > 3);  
 //Assert false  
 *assertFalse*(5 < 3);  
 //Assert null  
 Object obj = null;  
 *assertNull*(obj);  
 //Assert not null  
 Object obj2 = new Object();  
 *assertNotNull*(obj2);  
 }  
}

Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and

Teardown Methods in Junit

package org.example;  
  
public class Calculator {  
 public int add(int a, int b){  
 return a + b;  
 }  
 public int subtract(int a, int b){  
 return a - b;  
 }  
}

package org.example;  
  
import org.junit.Before;  
import org.junit.After;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class CalculatorTest {  
  
 private Calculator calculator;  
  
 @Before  
 public void setUp() {  
 System.*out*.println("Setting up Calculator...");  
 calculator = new Calculator(); // Arrange  
 }  
  
 @After  
 public void tearDown() {  
 System.*out*.println("Tearing down Calculator...");  
 calculator = null;  
 }  
  
 @Test  
 public void testAddition() {  
 // Act  
 int result = calculator.add(2, 3);  
  
 // Assert  
 *assertEquals*(5, result);  
 }  
  
 @Test  
 public void testSubtraction() {  
 // Act  
 int result = calculator.subtract(5, 2);  
  
 // Assert  
 *assertEquals*(3, result);  
 }  
}

Output:

