JUnit Testing Exercises - Solution with Output

# Exercise 1: Setting Up JUnit

1. Create a new Java project in your IDE (e.g., IntelliJ IDEA, Eclipse).  
2. Add JUnit dependency to your project. If you are using Maven, add the following to your pom.xml:

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

3. Create a new test class in your project.

# Exercise 2: Writing Basic JUnit Tests

Java class to be tested:

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

JUnit Test class:

import static org.junit.Assert.\*;  
import org.junit.Test;  
  
public class CalculatorTest {  
  
 @Test  
 public void testAdd() {  
 Calculator calc = new Calculator();  
 assertEquals(5, calc.add(2, 3));  
 }  
  
 @Test  
 public void testSubtract() {  
 Calculator calc = new Calculator();  
 assertEquals(1, calc.subtract(3, 2));  
 }  
}

# Exercise 3: Assertions in JUnit

Solution code for different assertions:

import static org.junit.Assert.\*;  
import org.junit.Test;  
  
public class AssertionsTest {  
 @Test  
 public void testAssertions() {  
 assertEquals(5, 2 + 3);  
 assertTrue(5 > 3);  
 assertFalse(5 < 3);  
 assertNull(null);  
 assertNotNull(new Object());  
 }  
}

# Exercise 4: AAA Pattern and Setup/Teardown Methods

JUnit test using Arrange-Act-Assert pattern and setup/teardown methods:

import org.junit.Before;  
import org.junit.After;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class AaaPatternTest {  
  
 private Calculator calc;  
  
 @Before  
 public void setUp() {  
 calc = new Calculator(); // Arrange  
 }  
  
 @After  
 public void tearDown() {  
 calc = null; // Cleanup  
 }  
  
 @Test  
 public void testAddition() {  
 int result = calc.add(10, 5); // Act  
 assertEquals(15, result); // Assert  
 }  
}