**JUnit Testing Exercises Answers**

**Exercise 1: Setting Up JUnit**

**Steps to Set Up JUnit:**

1. **Create a new Java project** in your IDE (IntelliJ IDEA, Eclipse, etc.)
   * In IntelliJ: File → New → Project → Java
2. **Add JUnit dependency** (Maven example - corrected version):

xml

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

1. **Create a test class**:
   * Right-click on src/test/java
   * New → Java Class
   * Name it (e.g., MyFirstTest.java)

**Exercise 2: Writing Basic JUnit Tests**

**Example Solution:**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

*// Test class*

import org.junit.Test;

import static org.junit.Assert.\*;

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(4, 3));

}

}

**Exercise 3: Assertions in JUnit**

**Corrected Solution (fixing the original error):**

java

public class AssertionsTest {

@Test

public void testAssertions() {

*// Assert equals*

assertEquals(5, 2 + 3);

*// Assert true*

assertTrue(5 > 3);

assertFalse(5 < 3);

assertNull(null);

assertNotNull(new Object());

}

}

**Exercise 4: AAA Pattern and Test Fixtures**

**Example Implementation:**

java

import org.junit.\*;

import static org.junit.Assert.\*;

public class BankAccountTest {

private BankAccount account;

@Before

public void setUp() {

*// Arrange - runs before each test*

account = new BankAccount(1000);

}

@After

public void tearDown() {

*// Teardown - runs after each test*

account = null;

}

@Test

public void testWithdraw() {

*// Act*

account.withdraw(500);

*// Assert*

assertEquals(500, account.getBalance());

}

@Test

public void testDeposit() {

*// Act*

account.deposit(200);

*// Assert*

assertEquals(1200, account.getBalance());

}

}

**Key Notes:**

* Follows **Arrange-Act-Assert (AAA)** pattern
* Uses @Before for setup and @After for cleanup
* Each test is independent