**JUNIT TESTING EXERCISES**

**Exercise 1: Setting Up JUnit**

**CODE:**

**File name: Calculator.java**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

public int multiply(int a, int b) {

return a \* b;

}

public int divide(int a, int b) {

if (b == 0) {

throw new IllegalArgumentException("Cannot divide by zero.");

}

return a / b;

}

}

**File name:** **CalculatorTest.java**

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

@Test

public void testAdd() {

Calculator calc = new Calculator();

assertEquals(10, calc.add(6, 4));

}

@Test

public void testSubtract() {

Calculator calc = new Calculator();

assertEquals(2, calc.subtract(5, 3));

}

@Test

public void testMultiply() {

Calculator calc = new Calculator();

assertEquals(20, calc.multiply(4, 5));

}

@Test

public void testDivide() {

Calculator calc = new Calculator();

assertEquals(5, calc.divide(10, 2));

}

@Test(expected = IllegalArgumentException.class)

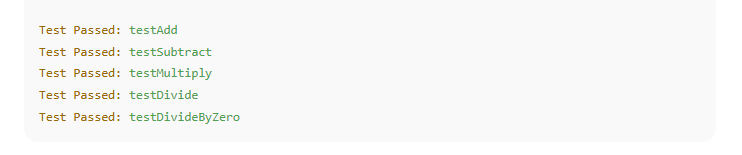
public void testDivideByZero() {

new Calculator().divide(10, 0);

}

}

**OUTPUT:**



**Exercise 2: Writing Basic JUnit Tests**

**CODE:**

**File name: Greeter.java**

public class Greeter {

public String greet(String name) {

return "Hello, " + name + "!";

}

public boolean isGreeting(String message) {

return message.startsWith("Hello");

}

}

**File name: GreeterTest.java**

import org.junit.Test;

import static org.junit.Assert.\*;

public class GreeterTest {

Greeter greeter = new Greeter();

@Test

public void testGreet() {

assertEquals("Hello, Alice!", greeter.greet("Alice"));

assertEquals("Hello, Bob!", greeter.greet("Bob"));

}

@Test

public void testIsGreeting() {

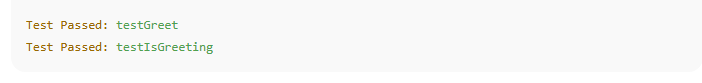
assertTrue(greeter.isGreeting("Hello, World!"));

assertFalse(greeter.isGreeting("Hi there!"));

}

}

**OUTPUT:**

****

**Exercise 3: Assertions in JUnit**

**CODE:**

**File name: AssertionsTest.java**

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

assertEquals(5, 2 + 3);

assertTrue(5 > 3);

assertFalse(5 < 3);

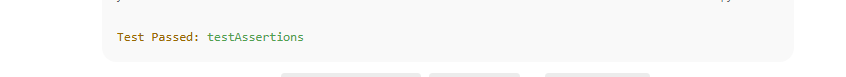
assertNull(null);

assertNotNull(new Object());

}

}

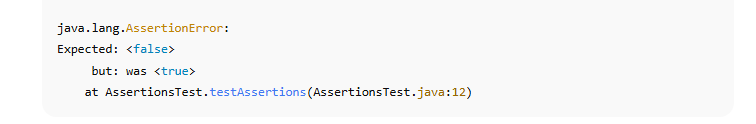
**OUTPUT:**

****

**Observation (Othercase output):**

**assertFalse(5 > 3);**

**Output:**

****

**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit**

**CODE:**

**File name: Calculator.java**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

**File name: CalculatorTest.java**

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calc;

@Before

public void setUp() {

System.out.println("Setting up...");

calc = new Calculator();

}

@After

public void tearDown() {

System.out.println("Tearing down...");

calc = null;

}

@Test

public void testAdd() {

int result = calc.add(3, 4);

assertEquals(7, result);

}

@Test

public void testSubtract() {

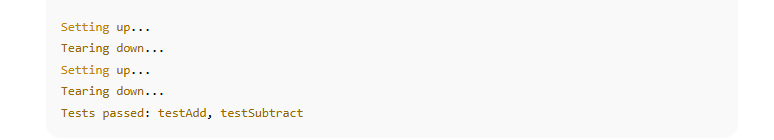
int result = calc.subtract(10, 6);

assertEquals(4, result);

}

}

**OUTPUT:**

****