***WEEK 2***

***1. JUnit Testing Exercises***

**Exercise 1: Setting Up Junit**

**Calculator.java**

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**CalculatorTest.java**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Test;

public class CalculatorTest {

*@Test*

public void testAdd() {

Calculator calc = new Calculator();

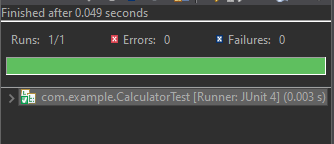
int result = calc.add(5, 3);

*assertEquals*(8, result);

}

}

**OUTPUT**

****

**Exercise 2: Writing Basic JUnit Tests**

**MathUtils.java**

package com.example;

public class MathUtils {

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;

}

}

**MathUtilsTest.java**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Test;

public class MathUtilsTest {

MathUtils mathUtils = new MathUtils();

*@Test*

public void testAdd() {

*assertEquals*(10, mathUtils.add(6, 4));

}

*@Test*

public void testSubtract() {

*assertEquals*(2, mathUtils.subtract(5, 3));

}

*@Test*

public void testMultiply() {

*assertEquals*(20, mathUtils.multiply(4, 5));

}

*@Test*

public void testDivide() {

*assertEquals*(3, mathUtils.divide(9, 3));

}

*@Test*(expected = IllegalArgumentException.class)

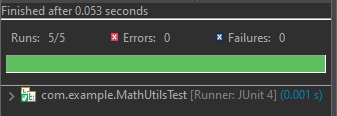
public void testDivideByZero() {

mathUtils.divide(10, 0);

}

}

**OUTPUT**

****

**Exercise 3: Assertions in Junit**

**AssertionsTest.java**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Test;

public class AssertionsTest {

*@Test*

public void testAssertions() {

// Assert equals

*assertEquals*(5, 2 + 3);

// Assert true

*assertTrue*(5 > 3);

// Assert false

*assertFalse*(5 < 3);

// Assert null

Object obj1 = null;

*assertNull*(obj1);

// Assert not null

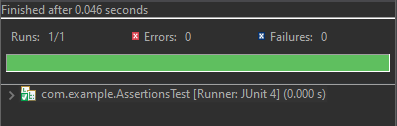
Object obj2 = new Object();

*assertNotNull*(obj2);

}

}

**OUTPUT**

****

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

**BankAccount.java**

package com.example;

public class BankAccount {

private String owner;

private double balance;

public BankAccount(String owner, double balance) {

this.owner = owner;

this.balance = balance;

}

public void deposit(double amount) {

if (amount <= 0) throw new IllegalArgumentException("Deposit must be positive");

balance += amount;

}

public void withdraw(double amount) {

if (amount > balance) throw new IllegalArgumentException("Insufficient balance");

balance -= amount;

}

public double getBalance() {

return balance;

}

public String getOwner() {

return owner;

}

}

**BankAccountTest.java**

package com.example;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class BankAccountTest {

private BankAccount account;

// Setup method - runs before each test

*@Before*

public void setUp() {

account = new BankAccount("Indhumathi", 1000.0);

System.***out***.println("Setup complete");

}

// Teardown method - runs after each test

*@After*

public void tearDown() {

account = null;

System.***out***.println("Teardown complete");

}

*@Test*

public void testDeposit() {

// Arrange - Already done in setUp()

// Act

account.deposit(500.0);

// Assert

*assertEquals*(1500.0, account.getBalance(), 0.001);

}

*@Test*

public void testWithdraw() {

// Arrange

// Act

account.withdraw(400.0);

// Assert

*assertEquals*(600.0, account.getBalance(), 0.001);

}

*@Test*(expected = IllegalArgumentException.class)

public void testWithdrawInsufficientFunds() {

// Act

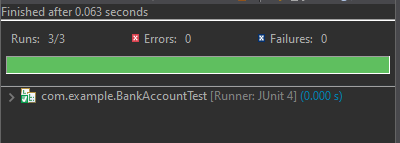
account.withdraw(1500.0); // Should throw exception

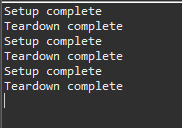
// Assert is handled by expected exception

}

}

**OUTPUT**

****

****