**WEEK 2 - JUNIT TESTING EXERCISES**

**EXERCISE 1:**

**Setting Up JUnit Java project to start writing unit tests**

**STEP 1: CREATING A JAVA PROJECT**

I created a new **Maven-based Java project** using **IntelliJ IDEA**. The project was named MathAppTest and configured with **Java 1.8 JDK**. The standard folder structure was generated with src/main/java for source files and src/test/java for test files.

**STEP 2: ADDING JUNIT DEPENDENCY**

For this exercise, we needed the **JUnit 4.13.2** library to write test cases.

xml

CopyEdit

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

After saving the file, IntelliJ automatically downloaded the required JARs and linked the JUnit library to the project. This allowed me to use JUnit annotations like @Test and methods like assertEquals() in the test classes.

**STEP 3: JAVA CLASS AND JUNIT TEST CLASS**

**MAIN JAVA CLASS (MathUtility.java)**

package com.assignment;

public class MathUtility {

public int sum(int a, int b) {

return a + b;

}

public int diff(int a, int b) {

return a - b;

}

public int product(int a, int b) {

return a \* b;

}

public int divide(int a, int b) {

if (b == 0) throw new ArithmeticException("Cannot divide by zero");

return a / b;

}

}

**JUNIT TEST CLASS (MathUtilityTest.java)**

package com.assignment;

import org.junit.Test;

import static org.junit.Assert.\*;

public class MathUtilityTest {

MathUtility math = new MathUtility();

@Test

public void testSum() {

assertEquals(14, math.sum(9, 5));

}

@Test

public void testDiff() {

assertEquals(2, math.diff(7, 5));

}

@Test

public void testProduct() {

assertEquals(21, math.product(7, 3));

}

@Test

public void testDivide() {

assertEquals(4, math.divide(20, 5));

}

@Test(expected = ArithmeticException.class)

public void testDivideByZero() {

math.divide(8, 0);

}

}

**Sample Output**

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0

BUILD SUCCESSFUL