WEEK 2

Mandatory Hands- On

**TEST DRIVEN DEVELOPMENT**

**EX:1** **JUNIT TESTING EXERCISES**

**STEP 1: CREATING A JAVA PROJECT**

I created a new **Maven-based Java project** using **IntelliJ IDEA**. The project was named JUnitDemo 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 ( Calculator.java)**

package com.example;

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 ArithmeticException("Division by zero");

return a / b;

}

}

#### **JUNIT TEST CLASS (Calculatortest.Java**)

package com.example;

import org.junit.Test;

import static org.junit.Assert;

public class CalculatorTest {

Calculator calc = new Calculator();

@Test

public void testAddition() {

assertEquals(10, calc.add(7, 3));

}

@Test

public void testSubtraction() {

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

}

@Test

public void testMultiplication() {

assertEquals(15, calc.multiply(3, 5));

}

@Test

public void testDivision() {

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

}

@Test(expected = ArithmeticException.class)

public void testDivisionByZero() {

calc.divide(10, 0);

}

}

### ****Sample Output****

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

BUILD SUCCESSFUL