# **Exercise 1: Setting Up JUnit**

## **Scenario**

**You need to set up JUnit in your Java project to start writing unit tests.**

**Exercise 2: Writing Basic JUnit Tests**

**Scenario:**

**You need to write basic JUnit tests for a simple Java class.**

**Exercise 3: Assertions in JUnit**

**Scenario:**

**You need to use different assertions in JUnit to validate your test results.**

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

**Teardown Methods in JUnit**

**Scenario:**

**You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup**

**and teardown methods.**

## **Source Code**

## **MathOps.java**

package org.demo;

public class MathOps {

public int sum(int a, int b) {

return a + b;

}

public int diff(int a, int b) {

return a - b;

}

public int prod(int a, int b) {

return a \* b;

}

public int quot(int a, int b) {

if (b == 0) {

throw new ArithmeticException("Cannot divide by zero");

}

return a / b;

}

public int rem(int a, int b) {

if (b == 0) {

throw new ArithmeticException("Cannot mod by zero");

}

return a % b;

}

public boolean isNumberEven(int value) {

return value % 2 == 0;

}

public boolean isNumberPositive(int value) {

return value > 0;

}

}

## **Test Code**

## **MathOpsTest.java**

package org.demo;

import org.junit.Test;

public class MathOpsTest {

private final MathOps ops = new MathOps();

@Test

public void testSum() {

int result = ops.sum(4, 3);

System.out.println("sum(4, 3) = " + result);

}

@Test

public void testDiff() {

int result = ops.diff(10, 6);

System.out.println("diff(10, 6) = " + result);

}

@Test

public void testProd() {

int result = ops.prod(3, 5);

System.out.println("prod(3, 5) = " + result);

}

@Test

public void testQuot() {

int result = ops.quot(20, 5);

System.out.println("quot(20, 5) = " + result);

}

@Test

public void testRem() {

int result = ops.rem(10, 3);

System.out.println("rem(10, 3) = " + result);

}

@Test

public void testIsNumberEven() {

boolean res = ops.isNumberEven(8);

System.out.println("isNumberEven(8) = " + res);

}

@Test

public void testIsNumberPositive() {

boolean res = ops.isNumberPositive(-5);

System.out.println("isNumberPositive(-5) = " + res);

}

}

## **pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>org.demo</groupId>

<artifactId>mathopsapp</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.2</version>

<configuration>

<redirectTestOutputToFile>false</redirectTestOutputToFile>

</configuration>

</plugin>

</plugins>

</build>

</project>

OUTPUT :

