**WEEK – 2**

**JUnit Basic Testing Exercises**

**Exercise 1: Setting Up Junit**

**CODE:**

**MainTest.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(5, calc.add(2, 3));

}

@Test

public void testSub() {

Calculator calc = new Calculator();

assertEquals(1, calc.sub(4, 3));

}

}

**Main.java:**

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

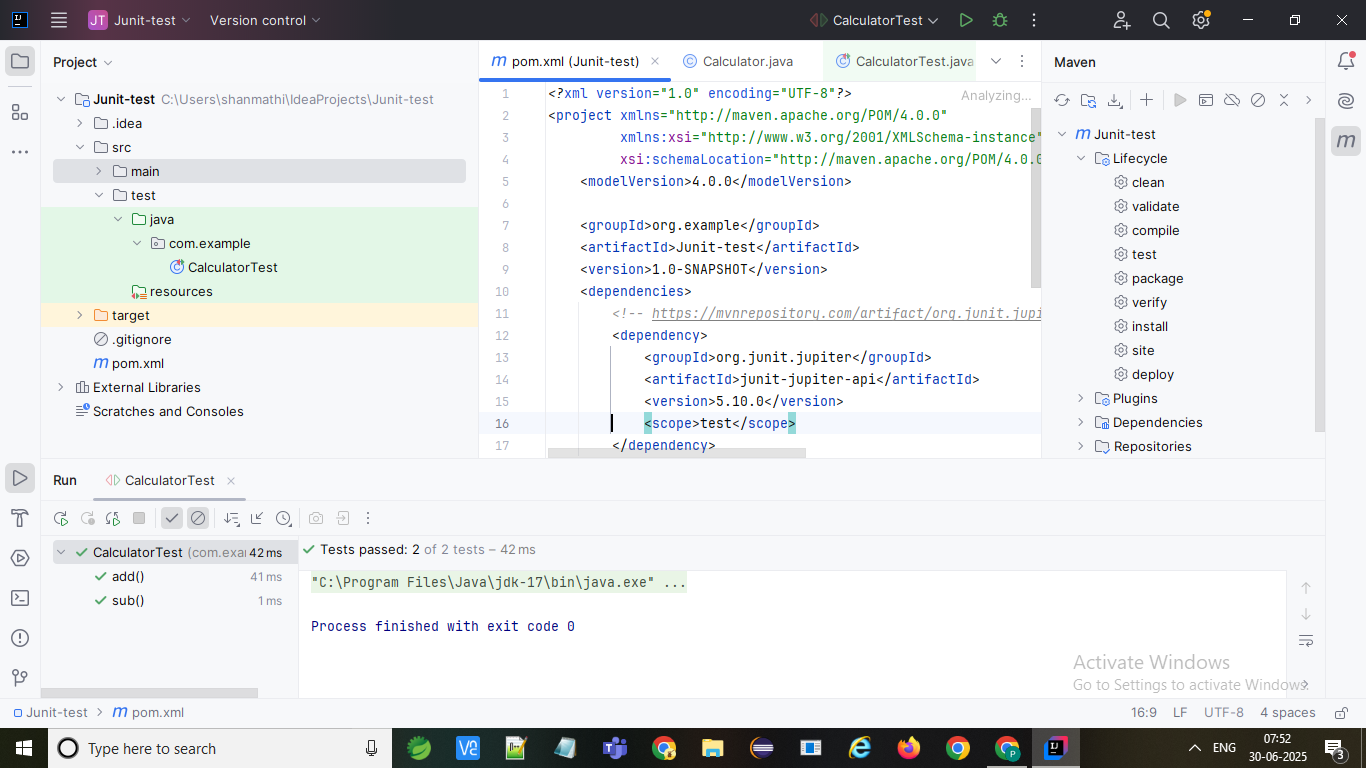
public int sub(int a, int b) {

return a - b;

}

}

**Output:**



**Exercise 3: Assertions in Junit**

**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);

String str = null;

assertNull(str);

Object obj = new Object();

assertNotNull(obj);

}

}

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

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

**CODE:**

**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;

}

}

**CalculatorFixtureTest.java:**import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorFixtureTest {

private Calculator calculator;

@Before

public void setUp() {

calculator = new Calculator();

System.out.println("Setup: New Calculator created");

}

@After

public void tearDown() {

calculator = null;

System.out.println("Teardown: Calculator cleared");

}

@Test

public void testAddition() {

int a = 10;

int b = 5;

int result = calculator.add(a, b);

assertEquals(15, result);

}

@Test

public void testSubtraction() {

int a = 10;

int b = 3;

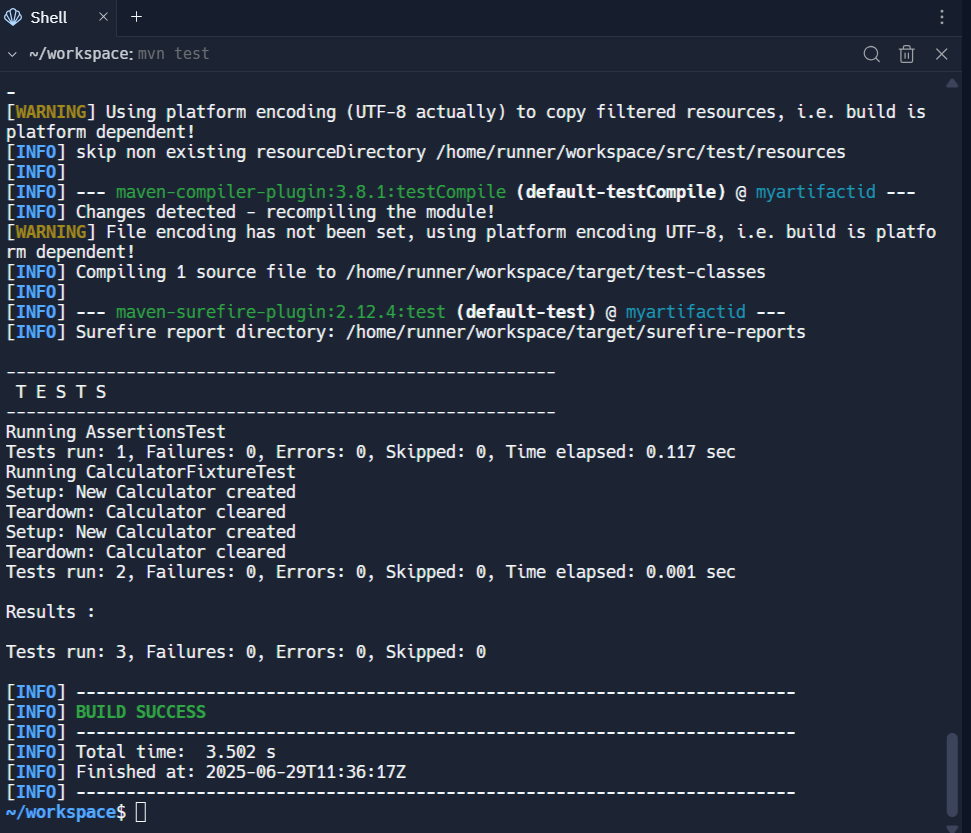
int result = calculator.subtract(a, b);

assertEquals(7, result);

}

}

**OUTPUT:**

****