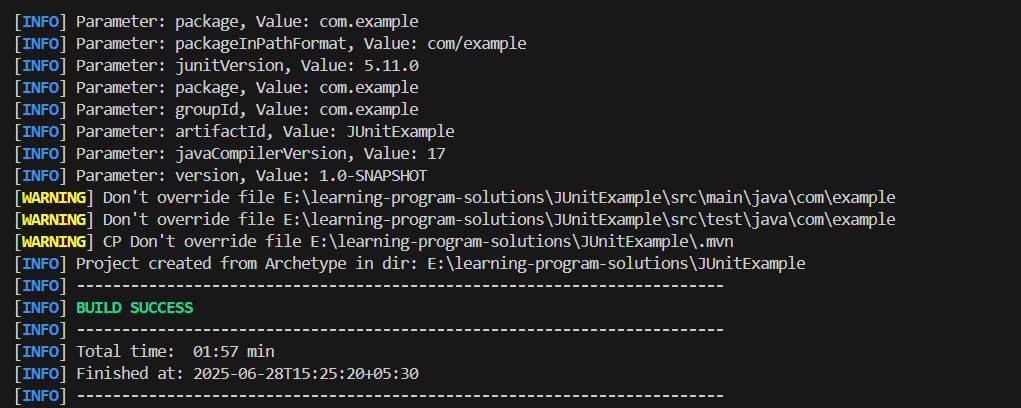
**MANDATORY\_Week2\_TDD using JUnit5 and Mockito**

**1. JUnit\_Basic Testing Exercises**

**Exercise 1: Setting Up Junit**

Output:  
  


**Exercise 2: Writing Basic JUnit Tests:  
Code:**

**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.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;

public class CalculatorTest {

    @Test

    public void testAddition() {

        Calculator calc = new Calculator();

        int result = calc.add(2, 3);

        assertEquals(5, result);

    }

}



**Exercise 3: Assertions in Junit**

**Code:**

package com.example;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

public class AssertionsTest {

    @Test

    public void testAssertions() {

        assertEquals(5, 2 + 3);

        assertTrue(5 > 3);

        assertFalse(5 < 3);

        assertNull(null);

        assertNotNull(new Object());

    }

}

**Output:**



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

**Teardown Methods in JUnit**

**CalculatorTest.java:**package com.example;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.AfterEach;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

public class CalculatorTest {

private Calculator calculator;

// Setup - runs before each test

@BeforeEach

public void setUp() {

calculator = new Calculator();

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

}

// Teardown - runs after each test

@AfterEach

public void tearDown() {

calculator = null;

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

}

@Test

public void testAddition() {

// Arrange

int a = 10;

int b = 5;

// Act

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

// Assert

assertEquals(15, result);

}

@Test

public void testAdditionWithNegative() {

// Arrange

int a = -4;

int b = 6;

// Act

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

// Assert

assertEquals(2, result);

}

}

**Calculator.Java:**

package com.example;

public class Calculator {

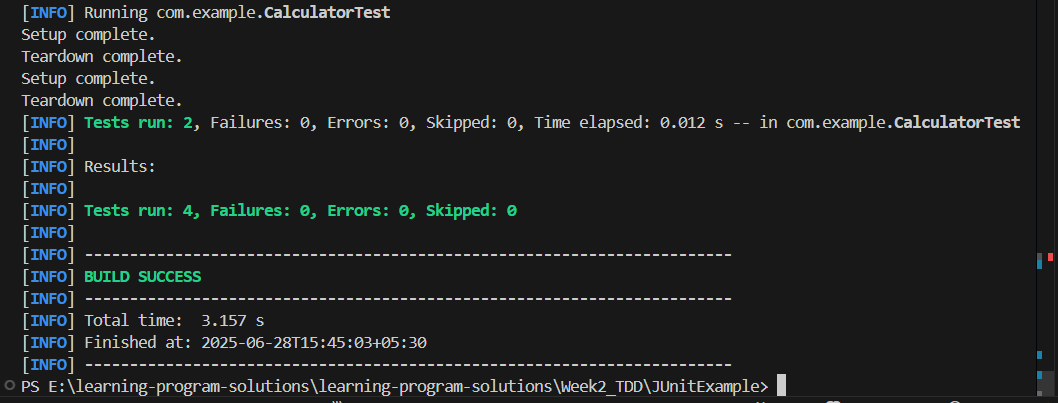
public int add(int a, int b) {

return a + b;

}

}

**Output**

****

**3. Mockito exercises**

**Exercise 1: Mocking and Stubbing Scenario:**

**ExternalApi.java**

package com.example;

public interface ExternalApi {

String getData();

}

**MyService.java**

package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData(); // Could be transformed/processed here

}

}

**MyServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

public class MyServiceTest {

    @Test

    public void testExternalApi() {

        ExternalApi mockApi = mock(ExternalApi.class);

        when(mockApi.getData()).thenReturn("Mock Data");

        MyService service = new MyService(mockApi);

        String result = service.fetchData();

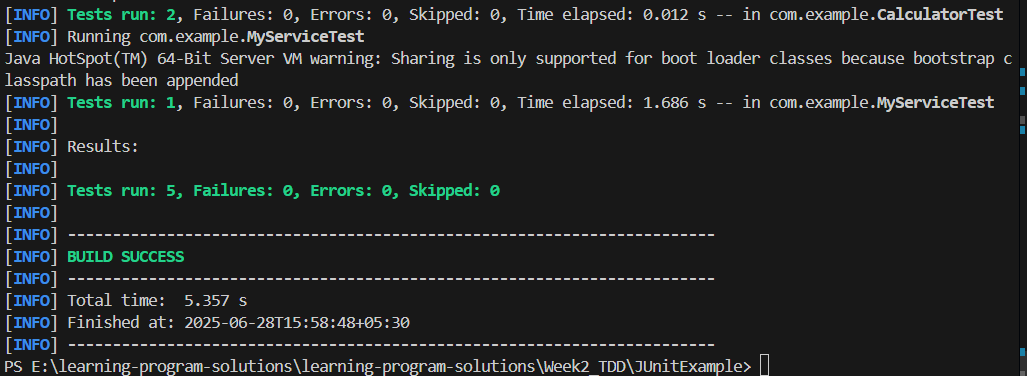
        assertEquals("Mock Data", result);

        verify(mockApi).getData();

    }

}

**Ouput**

****

**Exercise 2: Verifying Interactions**

**ExternalApi.java**

package com.example;

public interface ExternalApi {

String getData();

}

**MyService.java**

package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**MyServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

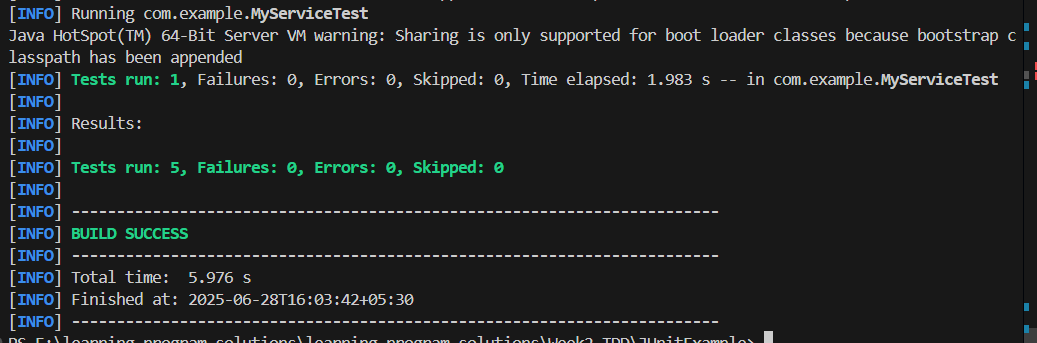
service.fetchData();

verify(mockApi).getData();

}

}

**Output**



**6. SL4J Logging exercises**

**LoggingExample.java**

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingExample {

    private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class);

    public static void main(String[] args) {

        logger.error("This is an error message");

        logger.warn("This is a warning message");

    }

}

**Output:**

