**TDD using JUnit5 and Mockito**

**1. JUnit\_Basic Testing Exercises**

**Exercise 1: Setting Up Junit  
code:**

import org.junit.Assert;

import org.junit.Test;

public class CalculatorTest {

static class Calculator {

public int add(int a, int b) {

return a + b;

}

}

@Test

public void testAdd() {

Calculator calc = new Calculator();

int result = calc.add(2, 3);

Assert.assertEquals(5, result);

}

public static void main(String[] args) {

CalculatorTest test = new CalculatorTest();

test.testAdd();

System.out.println("Test passed!");

}

}

**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>com.example</groupId>

<artifactId>junitsetup-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- JUnit 4.13.2 -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<properties>

<maven.compiler.source>24</maven.compiler.source>

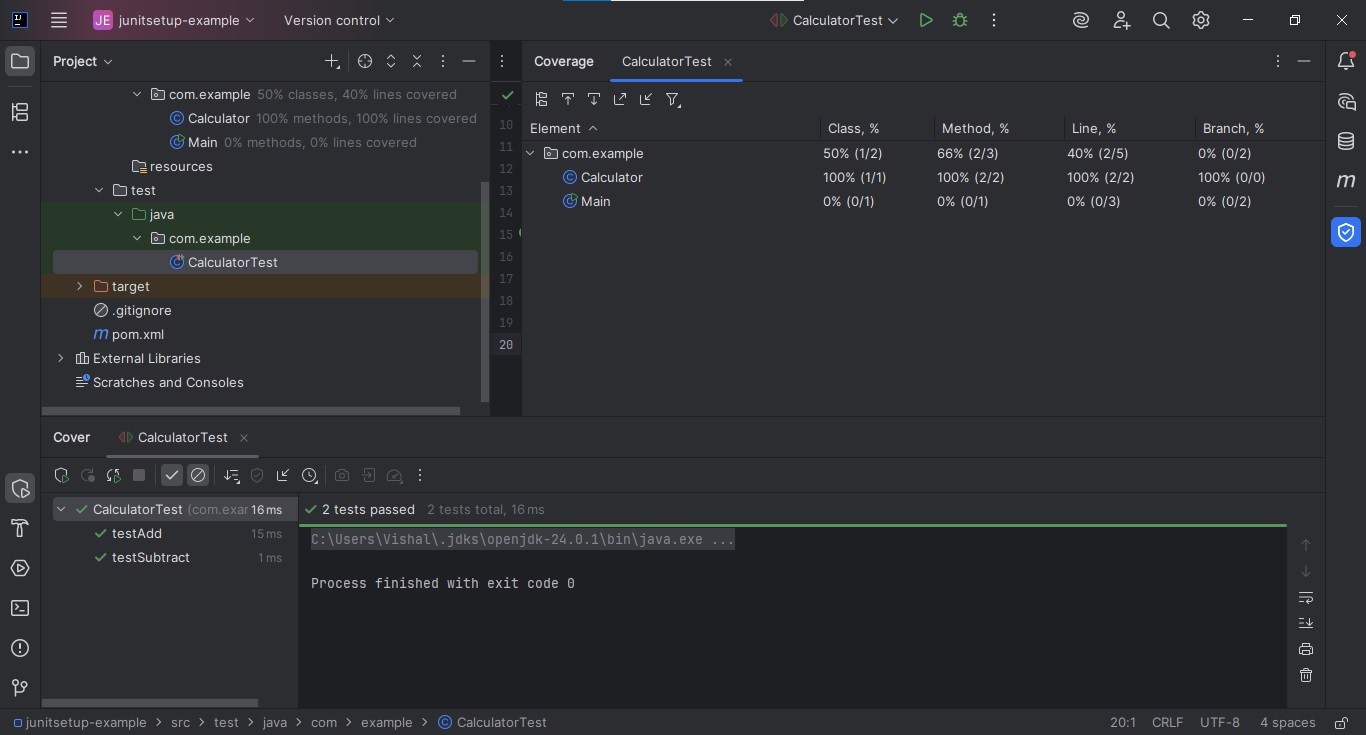
<maven.compiler.target>24</maven.compiler.target>

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

</properties>

</project>

**Output:**

  
  
**Exercise 3: Assertions in Junit  
code:**

**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);

assertNull(null);

assertNotNull(new Object());

}

}

@Test

public void testArrayAssertions() {

int[] expected = {1, 2, 3};

int[] actual = {1, 2, 3};

assertArrayEquals(expected, actual);

}

**pom.xml:**

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>junitsetup-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

<version>2.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<properties>

<maven.compiler.source>24</maven.compiler.source>

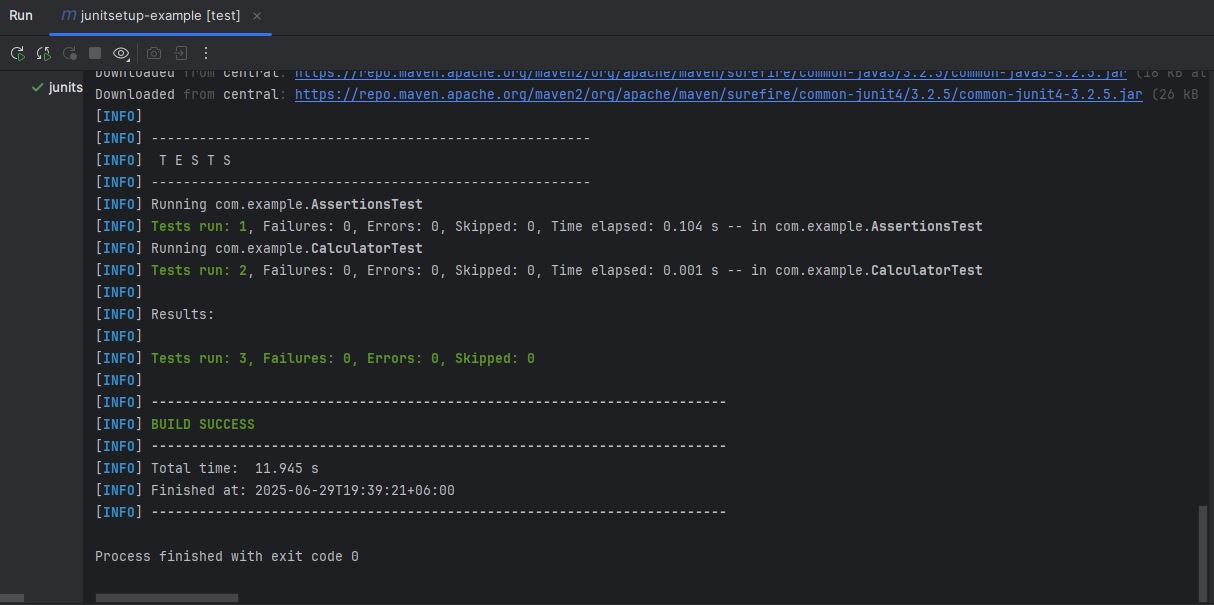
<maven.compiler.target>24</maven.compiler.target>

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

</properties>

</project>

**Output:**

****

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

package com.example;  
  
import org.junit.After;  
import org.junit.Assert;  
import org.junit.Before;  
import org.junit.Test;  
  
public class CalculatorFixtureTest {  
 private Calculator calculator;  
  
 @Before  
 public void setUp() {  
 this.calculator = new Calculator();  
 System.*out*.println("setUp: Calculator created");  
 }  
  
 @After  
 public void tearDown() {  
 this.calculator = null;  
 System.*out*.println("tearDown: Calculator cleaned up");  
 }  
  
 @Test  
 public void testAddition() {  
 int result = this.calculator.add(10, 5);  
 Assert.assertEquals(15L, (long)result);  
 }  
  
 @Test  
 public void testSubtraction() {  
 int result = this.calculator.subtract(10, 5);  
 Assert.assertEquals(5L, (long)result);  
 }  
}

**CalculatorTest.java:**

package com.example;  
  
import org.junit.Assert;  
import org.junit.Test;  
  
public class CalculatorTest {  
 @Test  
 public void testAdd() {  
 Calculator calc = new Calculator();  
 Assert.assertEquals(5L, (long)calc.add(2, 3));  
 }  
  
 @Test  
 public void testSubtract() {  
 Calculator calc = new Calculator();  
 Assert.assertEquals(1L, (long)calc.subtract(3, 2));  
 }  
}

**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>com.example</groupId>

<artifactId>junitsetup-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

<version>2.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<properties>

<maven.compiler.source>24</maven.compiler.source>

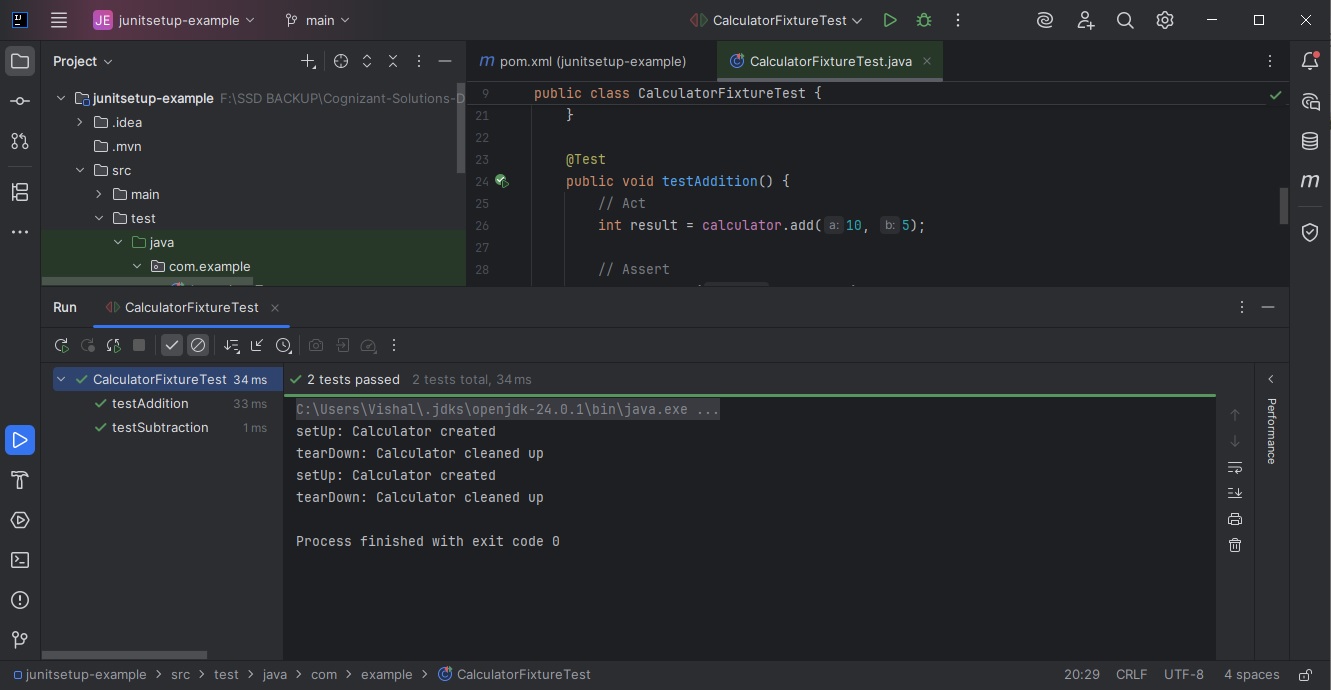
<maven.compiler.target>24</maven.compiler.target>

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

</properties>

</project>

**Output:**

****

**3. Mockito exercises  
  
Exercise 1: Mocking and Stubbing  
code:**

package com.example;  
  
import org.junit.Assert;  
import org.junit.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {  
 @Test  
 public void testExternalApi() {  
 ExternalApi mockApi = (ExternalApi)Mockito.mock(ExternalApi.class);  
 Mockito.when(mockApi.getData()).thenReturn("Mock Data");  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
 Assert.assertEquals("Mock Data", result);  
 }  
}

**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>com.example</groupId>

<artifactId>junitsetup-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

<version>2.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<properties>

<maven.compiler.source>24</maven.compiler.source>

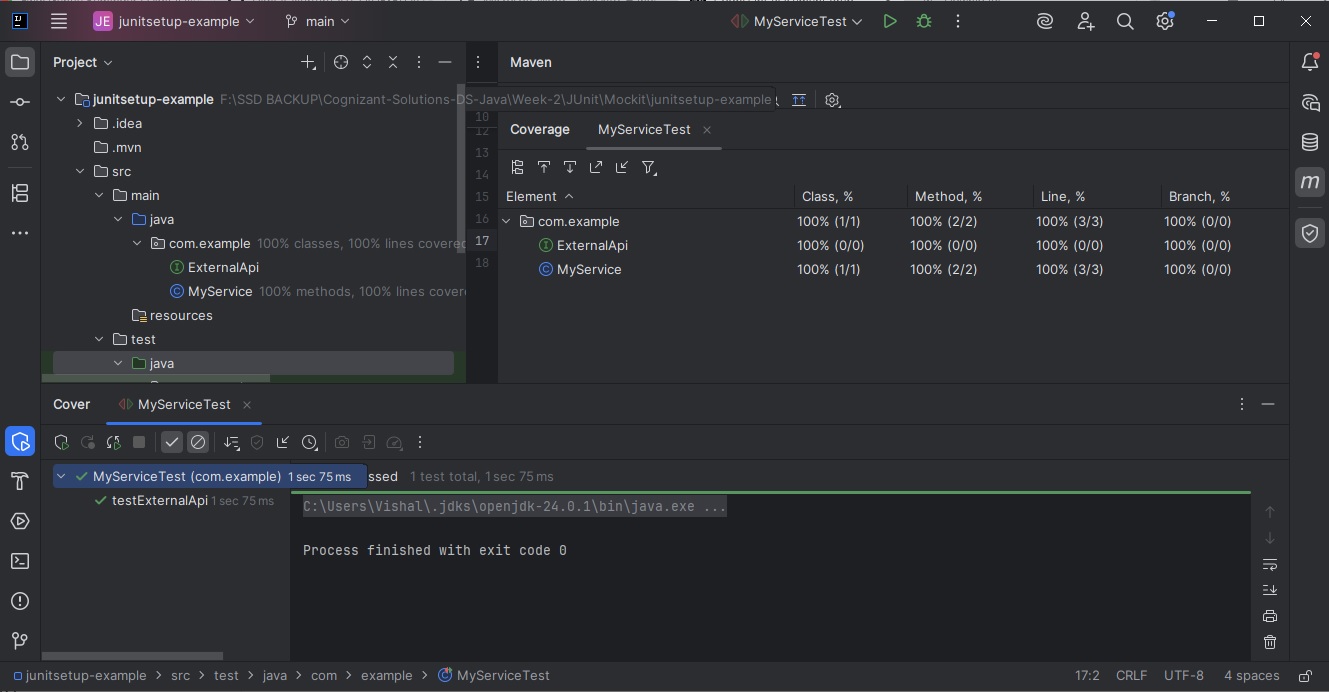
<maven.compiler.target>24</maven.compiler.target>

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

</properties>

</project>

**Output:**

****

**Exercise 2: Verifying Interactions  
code:**

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;   
public class MyService {  
 private ExternalApi api;  
  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
 public String fetchData() {  
 return this.api.getData();  
 }  
}

public interface ExternalApi {  
 String getData();  
}

public class MyServiceInteractionTest {  
 @Test  
 public void testVerifyInteraction() {  
 ExternalApi mockApi = (ExternalApi)Mockito.mock(ExternalApi.class);  
 MyService service = new MyService(mockApi);  
 service.fetchData();  
 ((ExternalApi)Mockito.verify(mockApi)).getData();  
 System.out.println("Mock Data Verified");  
 }  
}

public class MyServiceTest {  
 @Test  
 public void testExternalApi() {  
 ExternalApi mockApi = (ExternalApi)Mockito.mock(ExternalApi.class);  
 Mockito.when(mockApi.getData()).thenReturn("Mock Data");  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
 Assert.assertEquals("Mock Data", result);  
 }  
}

**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>com.example</groupId>

<artifactId>junitsetup-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

<version>2.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<properties>

<maven.compiler.source>24</maven.compiler.source>

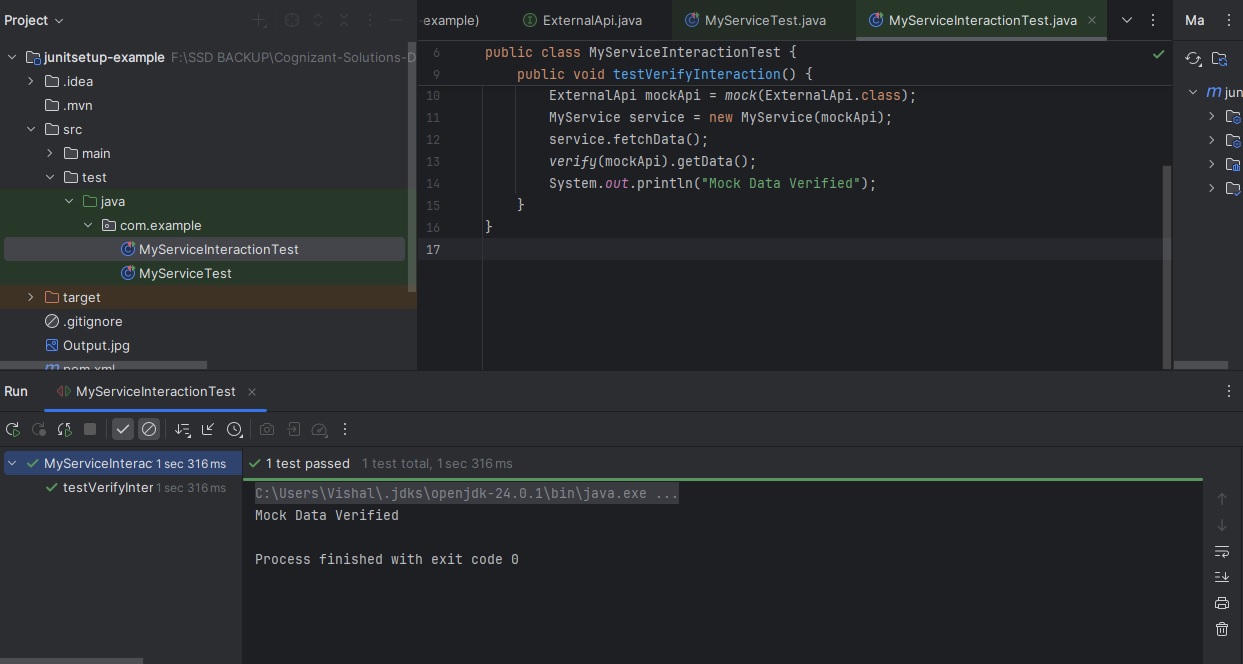
<maven.compiler.target>24</maven.compiler.target>

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

</properties>

</project>

**Output:**

****

**6. SL4J Logging exercises**

**Exercise 1: Logging Error Messages and Warning Levels   
  
code:**

import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Logging {  
 private static final Logger logger = LoggerFactory.getLogger(Logging.class);  
  
 public static void main(String[] args) {  
 logger.error("This is an error message!!!!!!");  
 logger.warn("This is a warning message----");  
 logger.info("️This is an info message.............");  
 }  
} **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>com.example</groupId>

<artifactId>junitsetup-example</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<!-- JUnit 4 for testing -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<!-- Hamcrest (used by JUnit assertions) -->

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

<version>2.2</version>

<scope>test</scope>

</dependency>

<!-- Mockito for mocking -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

<!-- SLF4J logging API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback (SLF4J implementation) -->

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

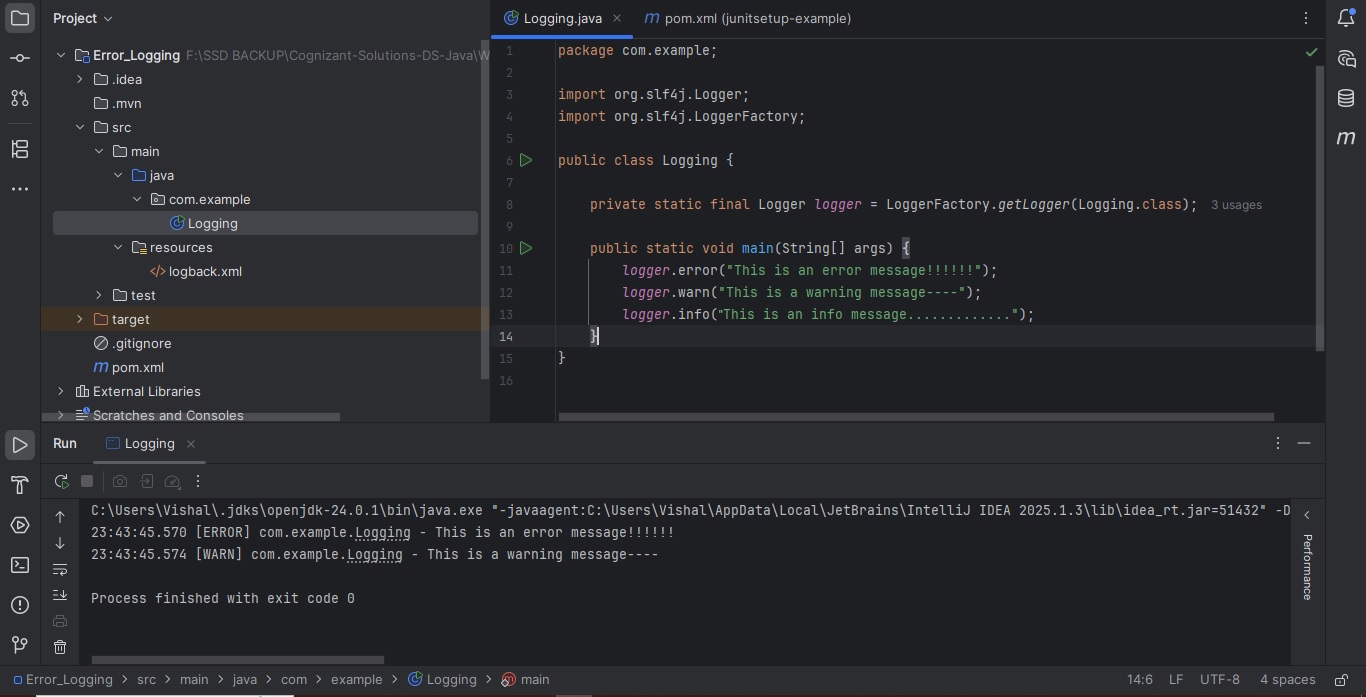
<version>1.2.3</version>

</dependency>

</dependencies>

</project>

**Output:**

****