**COGNIZANT DIGI NURTURE 4.0**

**WEEK 2 MANDATORY HANDS ON BY (6421405)**

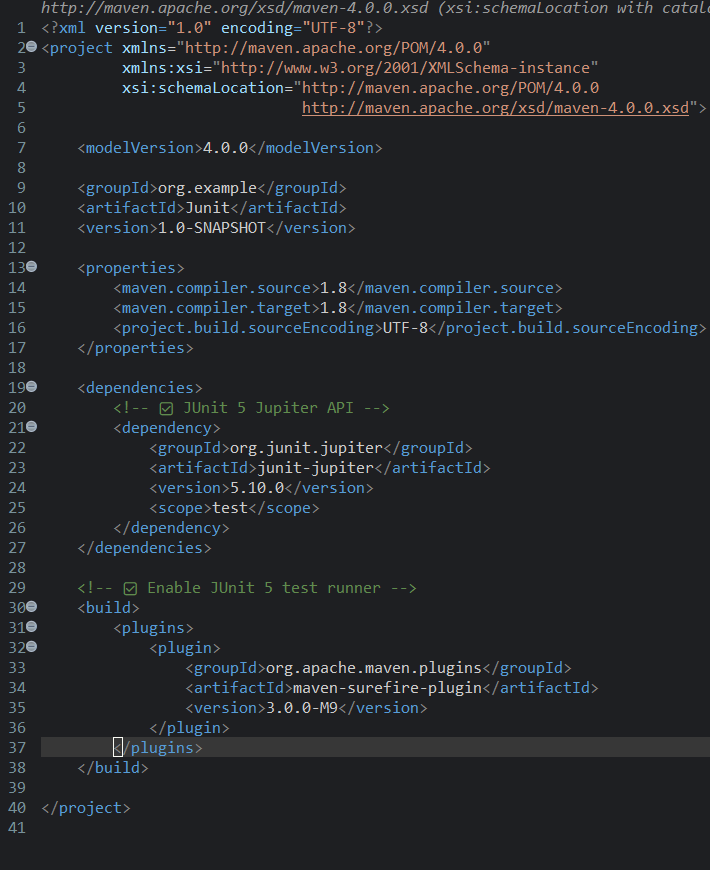
**Unit Testing:**

**Exercise 1: Setting up Junit**

**Step 1:**

Create a maven Project

**Pom.xml:**

****

**Step 2:**

In src/test/java, Create class name CalculatorTest.java

package org.example;

import org.junit.jupiter.api.Test;

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

public class CalculatorTest {

*@Test*

void testAddition() {

int result = 10 + 5;

System.***out***.println("Running testAddition...");

System.***out***.println("Result of 10 + 5 is: " + result);

*assertEquals*(15, result);

System.***out***.println("testAddition passed\n");

}

*@Test*

void testDivisionByZero() {

System.***out***.println("Running testDivisionByZero...");

*assertThrows*(ArithmeticException.class, () -> {

int x = 1 / 0;

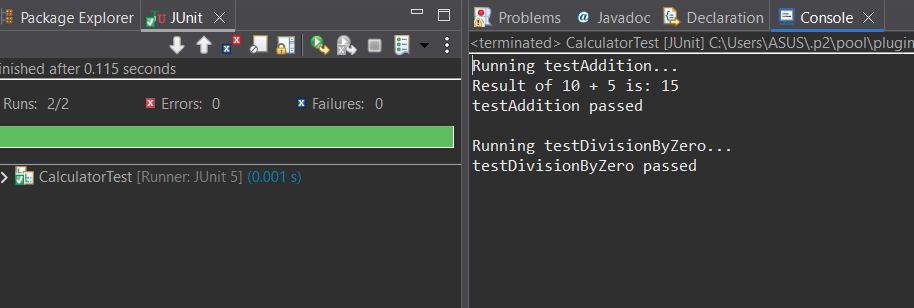
});

System.***out***.println("testDivisionByZero passed \n");

}

}

**OUTPUT**:



**Exercise 3:** Assertions in Junit

**Step 1:**

In src/test/java,Create java class named AssertionsTest.java.

**Assertiontest.java**

package org.example;

import org.junit.jupiter.api.Test;

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

public class assertionTest {

*@Test*

public void testAssertions() {

*assertEquals*(5,2+3);

System.*out*.println("checking asserts equals");

*assertTrue*(5 > 3);

System.*out*.println("checking assertTrue");

*assertFalse*(5 < 3);

System.*out*.println("Checking assertFalse");

*assertNull*(null);

System.*out*.println("Checking assertNull...");

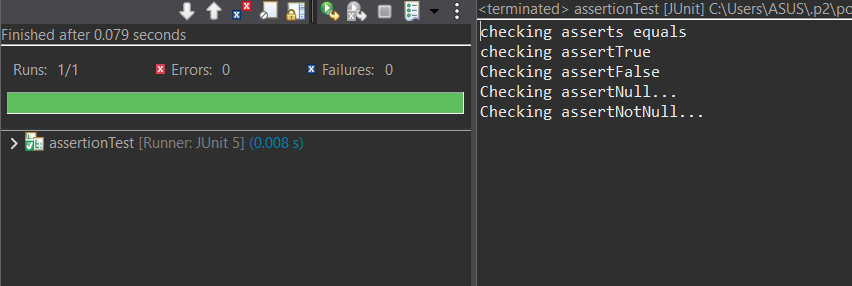
*assertNotNull*(new Object());

System.*out*.println("Checking assertNotNull...");

}

}

**OUTPUT:**

****

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

**Step1 :**

**In src/test/java/org.example,create a class named Calculator.java.**

package org.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**Step 2:**

In src/test/java/org.example,create a class named CalculatorTest.java.

package org.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 calc;

*@BeforeEach*

public void setUp() {

System.*out*.println("Setup: Initializing Calculator");

calc = new Calculator();

*@AfterEach*

public void tearDown() {

System.*out*.println("Teardown: Nullifying Calculator");

calc = null;

}

*@Test*

public void testAddition() {

int a = 15;

int b = 20;

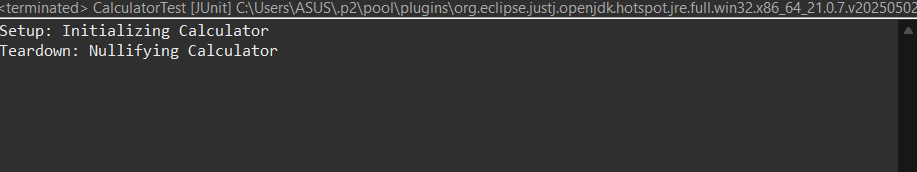
int result = calc.add(a, b);

*assertEquals*(35, result, "Addition should return");

}

}

**OUTPUT**

****

**3.Mockito excersice:**

**Step 1:**

Setup 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.CTS</groupId>

<artifactId>MockitoTestProject</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-api</artifactId>

<version>5.7.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-engine</artifactId>

<version>5.7.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.2.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.0.0-M8</version>

</plugin>

</plugins>

</build>

</project>

**Step 2:**

In src/test/java/org.example create an Interface named ExternalAPI.java

package org.example;

public interface ExternaAPI {

String getData();

}

**Step 3:**

In src/test/java/org.example create a Class Myservice.java

package org.example;

public class MyService {

private ExternalAPI api;

public MyService(ExternalAPI api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**Step 4:**

In **src/test/java/org.example,**create a class named **MyServiceTest.**

package org.example;

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

*@Test*

public void testExternalApi() {

ExternalAPI mockApi = Mockito.*mock*(ExternalAPI.class);

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

MyService service = new MyService(mockApi);

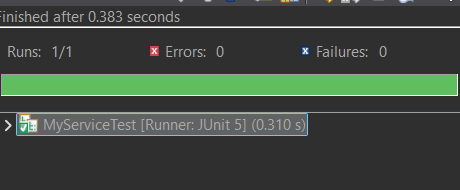
String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

}

**OUTPUT:**

****

**Exersice 2: Verifying Interaction**

In **src/test/java/org.example,**create or alter a class named **MyServiceTest.**

package org.example;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

*@Test*

public void testVerifyInteraction() {

ExternalAPI mockApi = Mockito.*mock*(ExternalAPI.class);

MyService service = new MyService(mockApi);

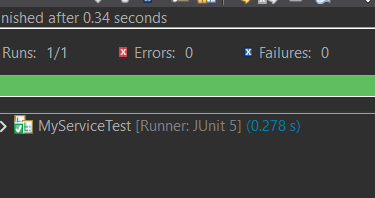
service.fetchData();

*verify*(mockApi).getData();

}

}

**OUTPUT:**

****

**6. SL4J Logging exercises**

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

**Step 1:**

Add the following dependencies on pom.xml.

<dependencies>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<dependency>

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

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

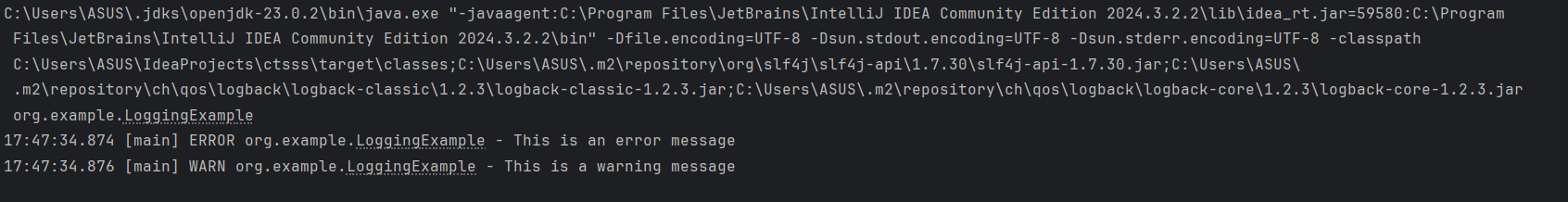
</dependencies>

**Step2:**

In **src/main/java/org.example** create an Interface named **LoggingExample.java.**

package org.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:**

****