**WEEK-2**

**JUnit Testing Exercises**

**Exercise 1: Setting Up Junit**

**Code:**

**SampleTest.java**

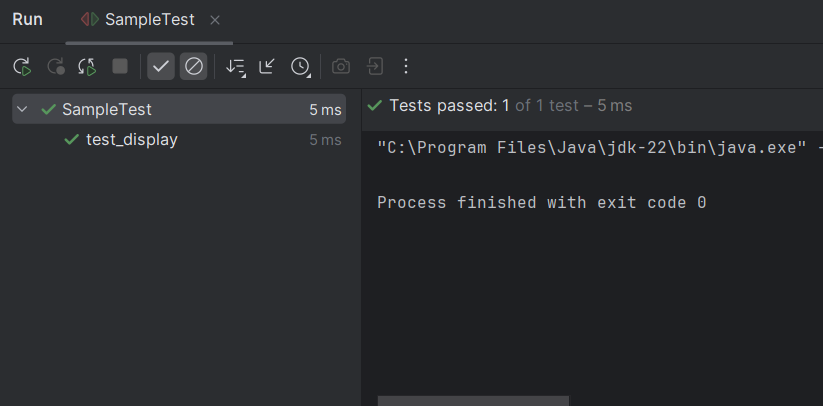
import static org.junit.Assert.\*;  
import org.junit.Test;  
  
public class SampleTest {  
  
 @Test  
 public void test\_display() {  
 String name="sanjai";  
 *assertEquals*("sanjai", name);  
 }  
}

**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>junit-setup</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <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>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Output:**

****

**Exercise 2: Writing Basic JUnit Tests**

**Code:**

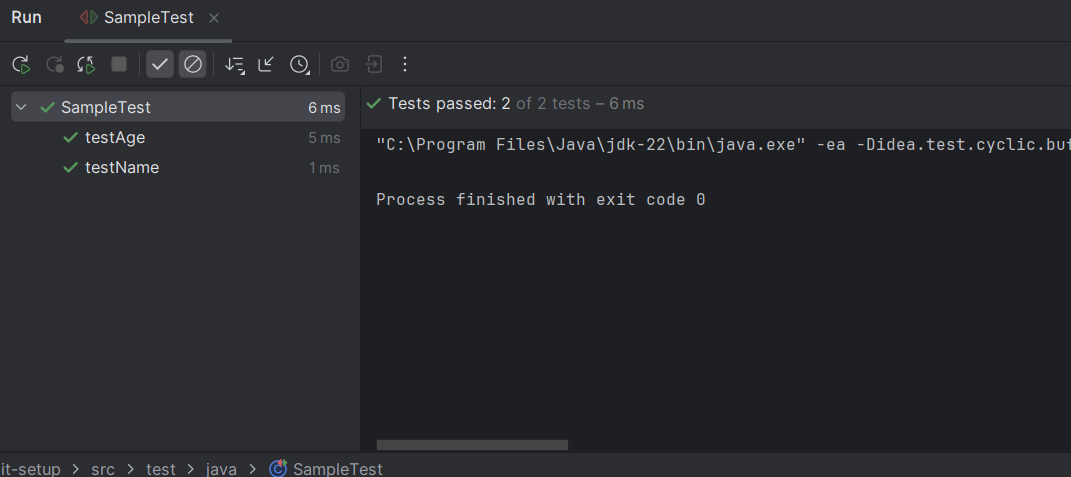
**Main.java**

package com.example;  
  
public class Main {  
 public String name(String name) {  
 return name;  
 }  
  
 public int age(int age) {  
 return age;  
 }  
}

**SampleTest.java**

import com.example.Main;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class SampleTest {  
  
 Main obj = new Main();  
  
 @Test  
 public void testName() {  
 *assertEquals*("Sanjai", obj.name("Sanjai"));  
 }  
  
 @Test  
 public void testAge() {  
 *assertEquals*(20, obj.age(20));  
 }  
}

**Output:**

****

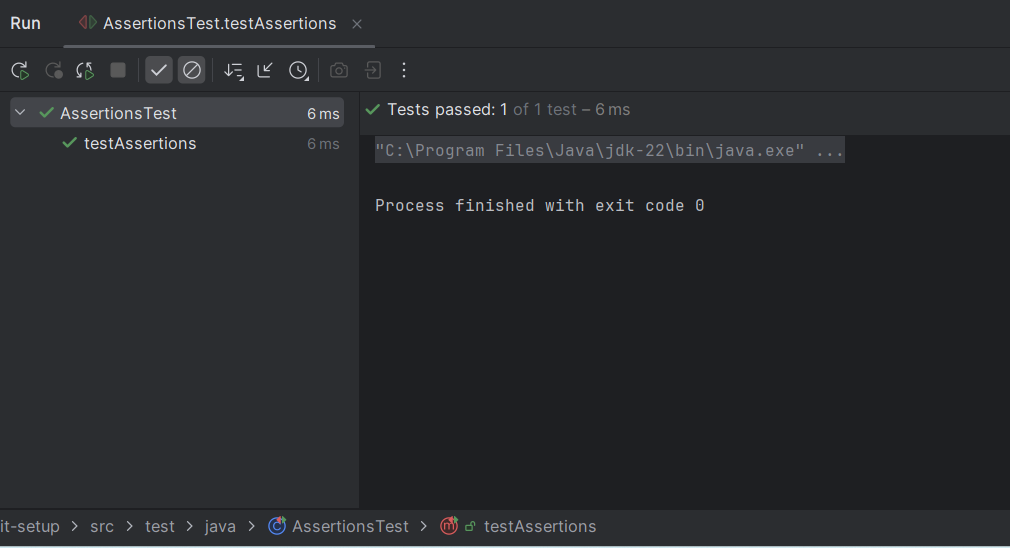
**Exercise 3: Assertions in JUnit**

**Code:**

**AssertionsTest.java**

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class AssertionsTest {  
  
 @Test  
 public void testAssertions() {  
 //Assert equals  
 *assertEquals*(5, 2 + 3);  
  
 // Assert true  
 *assertTrue*(5 > 3);  
  
 // Assert false  
 *assertFalse*(5 < 3);  
  
 // Assert null  
 *assertNull*(null);  
  
 // Assert not null  
 *assertNotNull*(new Object());  
 }  
}

**Output:**

****

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

**Code:**

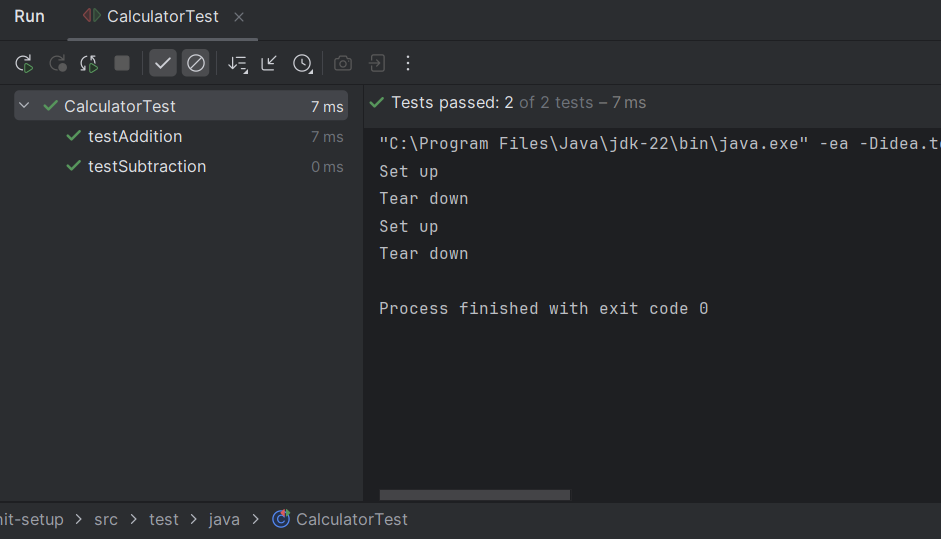
**Calculator.java**

package com.example;  
  
public class Calculator {  
 public int add(int a, int b) { return a + b; }  
 public int subtract(int a, int b) { return a - b; }  
}

**CalculatorTest.java**

import com.example.Calculator;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class CalculatorTest {  
  
 private Calculator cal;  
  
 @Before  
 public void setUp() {  
 System.*out*.println("Set up");  
 cal = new Calculator();  
 }  
  
 @After  
 public void tearDown() {  
 System.*out*.println("Tear down");  
 cal = null;  
 }  
  
 @Test  
 public void testAddition() {  
 int result = cal.add(15, 30);  
 *assertEquals*(45, result);  
 }  
  
 @Test  
 public void testSubtraction() {  
 int result = cal.subtract(30, 15);  
 *assertEquals*(15, result);  
 }  
}

**Output:**

****