**Week 2:**

**Exercise 1: Setting Up JUnit**

**Code:**

Pom.xml:

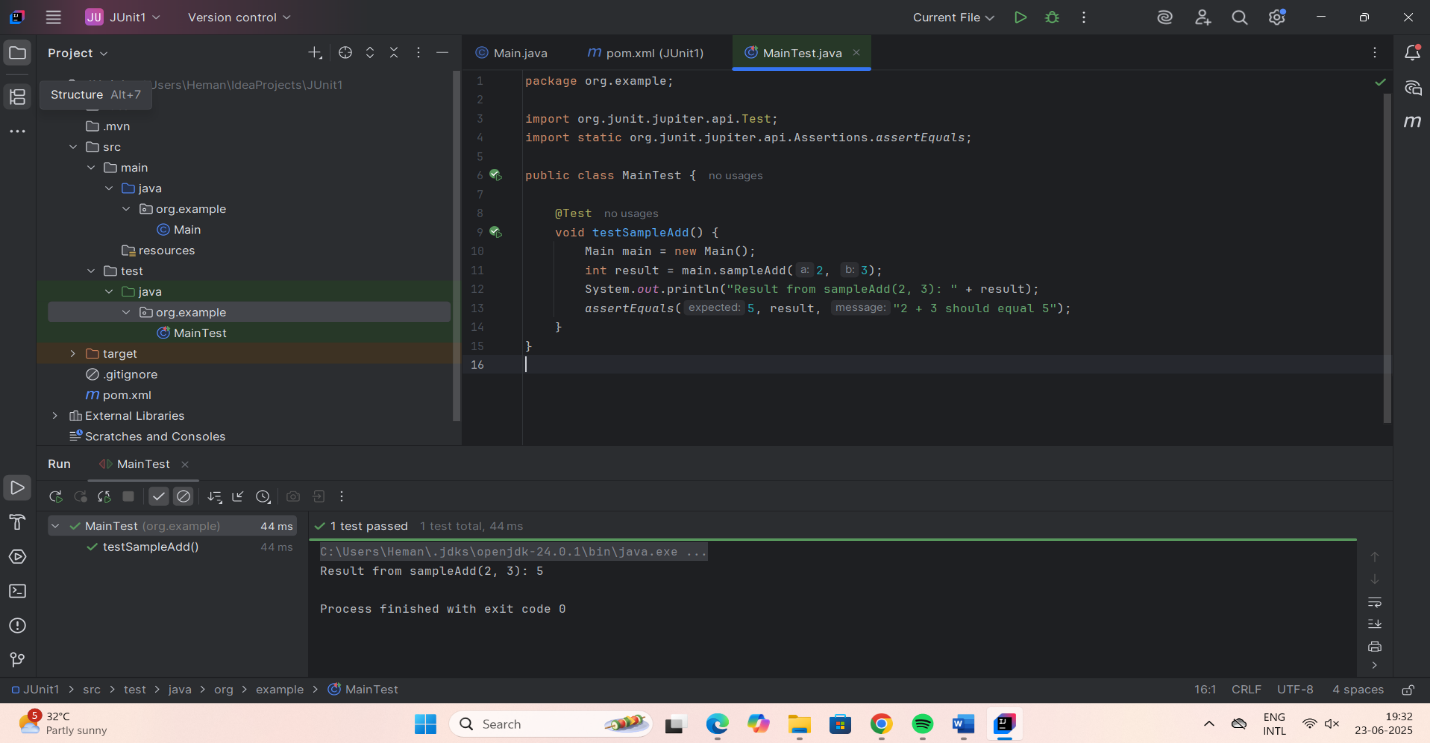
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>JUnit1</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <dependencies>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Main.java:**

package org.example;  
  
public class Main {  
 public int sampleAdd(int a, int b) {  
 return a + b;  
 }  
}

**MainTest.java:**

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
public class MainTest {  
  
 @Test  
 void testSampleAdd() {  
 Main main = new Main();  
 int result = main.sampleAdd(2, 3);  
 System.*out*.println("Result from sampleAdd(2, 3): " + result);  
 *assertEquals*(5, result, "2 + 3 should equal 5");  
 }  
}



**Exercise 3: Assertions in JUnit**

**Code:**

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>JUnitDemo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

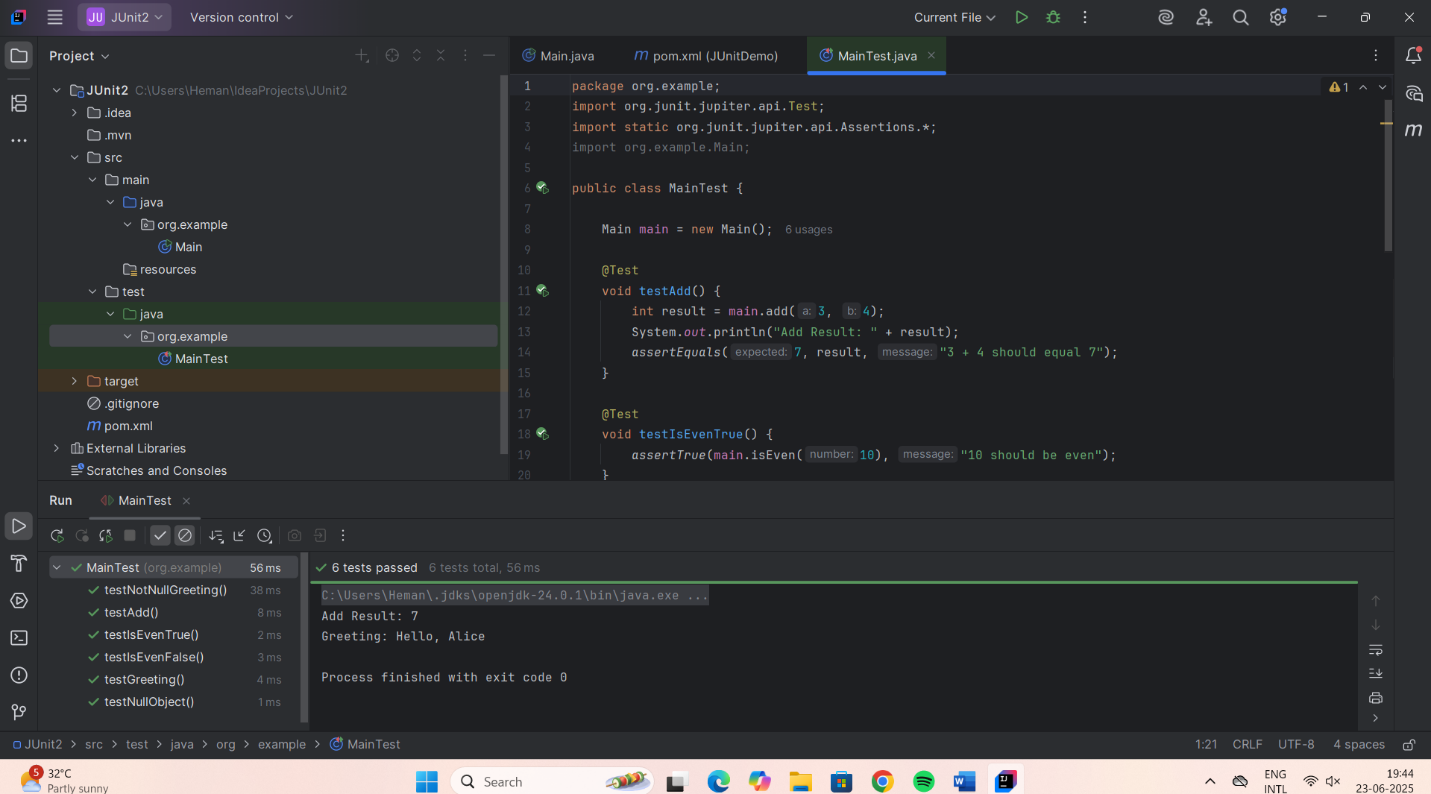
**Main.java:**

package org.example;  
public class Main {  
 // Adds two integers  
 public int add(int a, int b) {  
 return a + b;  
 }  
 // Checks if a number is even  
 public boolean isEven(int number) {  
 return number % 2 == 0;  
 }  
 // Returns a greeting message  
 public String greet(String name) {  
 return "Hello, " + name;  
 }  
 // Returns null (to test assertNull)  
 public Object getNull() {  
 return null;  
 }  
 public static void main(String[] args) {  
 System.*out*.println("Main class running.");  
 }  
}

**MainTest.java:**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
import org.example.Main;  
  
public class MainTest {  
  
 Main main = new Main();  
  
 @Test  
 void testAdd() {  
 int result = main.add(3, 4);  
 System.*out*.println("Add Result: " + result);  
 *assertEquals*(7, result, "3 + 4 should equal 7");  
 }  
  
 @Test  
 void testIsEvenTrue() {  
 *assertTrue*(main.isEven(10), "10 should be even");  
 }  
  
 @Test  
 void testIsEvenFalse() {  
 *assertFalse*(main.isEven(7), "7 should not be even");  
 }  
  
 @Test  
 void testGreeting() {  
 String message = main.greet("Alice");  
 System.*out*.println("Greeting: " + message);  
 *assertEquals*("Hello, Alice", message);  
 }  
  
 @Test  
 void testNotNullGreeting() {  
 *assertNotNull*(main.greet("Bob"), "Greeting should not be null");  
 }  
  
 @Test  
 void testNullObject() {  
 *assertNull*(main.getNull(), "Expected null from getNull()");  
 }  
}

**output:**



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

**Code:**

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>JUnitDemo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.9.3</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>  
  
  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-deploy-plugin</artifactId>  
 <version>2.8.2</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Main.java:**

package org.example;  
  
public class Main {  
 public int sampleAdd(int a, int b) {  
 return a + b;  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("Hello from Main!");  
 }  
 public int add(int a, int b) {  
 return a + b;  
 }  
  
 public boolean isEven(int number) {  
 return number % 2 == 0;  
 }  
  
 public String greet(String name) {  
 return "Hello, " + name;  
 }  
  
 public Object getNull() {  
 return null;  
 }  
  
}

**MainTest.java:**

package org.example;  
  
import org.junit.jupiter.api.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class MainTest {  
  
 private Main main;  
  
 @BeforeEach  
 void setUp() {  
 main = new Main();  
 }  
  
 @Test  
 void testAdd() {  
 *assertEquals*(8, main.add(5, 3));  
 }  
  
 @Test  
 void testIsEven() {  
 *assertTrue*(main.isEven(6));  
 }  
  
 @Test  
 void testGreet() {  
 *assertEquals*("Hello, Vedak", main.greet("Vedak"));  
 }  
  
 @Test  
 void testGetNull() {  
 *assertNull*(main.getNull());  
 }  
}

**Output:**

