**WEEK-02 HANDS ON SOLUTIONS**

**J-UNIT\_BASIC TESTING EXERCISES SOLUTIONS**

**Exercise 1: Setting Up JUnit Scenario:**

You need to set up JUnit in your Java project to start

writing unit tests.

Steps:

1. Create a new Java project in your IDE (e.g., IntelliJ IDEA, Eclipse).

2. Add JUnit dependency to your project.

If you are using Maven,

add the following to your pom.xml: junit junit 4.13.2 test

3. Create a new test class in your project.

**EXPLANATION:**

In this exercise, I set up a Maven-based Java project in Eclipse to practice writing unit tests using JUnit.

I created a new Maven project with the group ID com.testingexample and artifact ID JUnitMavenTest.

Maven automatically generated the standard project structure, including the src/main/java and src/test/java folders, along with a pom.xml file used to manage dependencies.

I then added the necessary JUnit dependency (version 4.13.2) and Hamcrest (for assertions) to the pom.xml, ensuring that my environment was equipped for unit testing.

Once the dependencies were configured, I created a simple Java class named TemperatureConverter inside the src/main/java/com/testingexample package.

This class included a method called celsiusToFahrenheit() that converts a Celsius temperature to Fahrenheit using the formula (C × 9/5) + 32.

After that, I wrote a corresponding test class named TemperatureConverterTest under src/test/java/com/testingexample.

Using the @Test annotation and assertEquals() from the JUnit library, I tested whether the method correctly converted 0°C to 32°F.

I executed the test by right-clicking on the test class and selecting

“Run As > JUnit Test.”

Eclipse successfully ran the test, and a green bar in the JUnit tab confirmed that the test passed without any errors. This output validated both the correctness of the conversion logic and the proper configuration of the JUnit testing environment within the Maven project.















