**WEEK-02 HANDS ON SOLUTIONS**

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

**Exercise 3: Assertions in JUnit**

Scenario: You need to use different assertions in JUnit to validate your test results.

1. Write tests using various JUnit assertions.

Solution

Code:

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());

}

}

**EXPLANATION:**

In this exercise, the focus was on utilizing various JUnit assertion methods to validate test conditions in a Java application.

A test class named AssertionsTest was created in the src/test/java/com.testingexample package.

Inside this class, a method testAssertions() was defined using the @Test annotation.

This method included several types of assertions such as assertEquals(), assertTrue(), assertFalse(), assertNull(), and assertNotNull() to demonstrate how JUnit can verify logical conditions and object references during testing.

Each assertion served a specific purpose—for example, assertEquals(5, 2 + 3) checked whether the sum of two numbers matched the expected value, while assertNull(null) ensured that a value was indeed null.

The test class was executed using Eclipse’s JUnit runner, and all assertions passed successfully, as indicated by the green bar in the JUnit panel.

This exercise helped reinforce how different assertions can be used effectively in unit testing to detect errors early and ensure code reliability.





