Exercise 3: Assertions in JUnit

Scenario

We had to write junit test cases and use diffirent assertions to check if the code is working as expected. Assertions help us to compare actual result with what we expect.

Step 1: Write test using diffrent JUnit assertions

I created a java class called `AssertionsTest` where I wrote tests for simple conditions using multiple types of assertions like `assertTrue`, `assertFalse`, `assertEquals` etc.

This is my code:

Code: `AssertionsTest.java`

```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 After Running `mvn test` in Terminal

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running AssertionsTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
```

Explanation of Each Assertion

- `assertEquals(5, 2 + 3);` → checks if sum is 5, which it is

- `assertTrue(5 > 3);` → checks if 5 is greater than 3, which is true

- `assertFalse(5 < 3);` → checks if 5 is not less than 3, so true

- `assertNull(null);` → checks if value is null, which it is

- `assertNotNull(new Object());` → checks that an object is not null