Exercise 4: Arrange-Act-Assert (AAA) Pattern, Setup and Teardown

---

Scenario

We were told to write test using Arrange Act Assert method (AAA) and also use setup and cleanup method in JUnit using `@Before` and `@After`. This makes the tests more clean and organized.

Step 1: Write test using AAA pattern

I made a simple Calculator class which adds numbers and then used that in test class.

Code: `Calculator.java`

```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }

    public int subtract(int a, int b) {
        return a - b;
    }
```

Code: `CalculatorTest.java`

```
import org.junit.Before;
import org.junit.After;
import org.junit.Test;
import static org.junit.Assert.*;

public class CalculatorTest {
```

```
private Calculator calc;

// Setup method
@Before
public void setUp() {
    calc = new Calculator();
    System.out.println("Setup done");
}

// Teardown method
@After
public void tearDown() {
    System.out.println("Test finished");
}

@Test
public void testAddition() {
    // Arrange
    int a = 4;
    int b = 6;

    // Act
    int result = calc.add(a, b);

    // Assert
    assertEquals(10, result);
}

@Test
public void testSubtraction() {
    // Arrange
    int a = 9;
    int b = 4;

    // Act
    int result = calc.subtract(a, b);
```

```
        // Assert
        assertEquals(5, result);
    }
}
```

Output After Running `mvn test`

```
Setup done
Test finished
Setup done
Test finished

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running CalculatorTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
```

Explanation of What Happened

- `@Before` runs **before every test** → it created Calculator object
- `@After` runs **after every test** → used to clean or print message
- I wrote each test in **AAA** format:
    - **Arrange** → setup test data
    - **Act** → call the method
    - **Assert** → check if result is correct