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

## 1. Objective

To organize tests using the Arrange-Act-Assert (AAA) pattern and utilize setup and teardown methods using JUnit annotations.

## 2. Problem Statement / Scenario

You need to structure your tests following the AAA pattern and prepare reusable test fixtures using @Before and @After annotations to manage setup and teardown activities.

## 3. Approach / Steps

3.1 Understand the AAA Pattern

The AAA pattern divides tests into three sections:  
- Arrange: Set up test data and environment  
- Act: Execute the code being tested  
- Assert: Verify the results

3.2 Setup and Teardown in JUnit

JUnit provides @Before (JUnit 4) or @BeforeEach (JUnit 5) for setup, and @After or @AfterEach for teardown.

## 4. Code

import org.junit.jupiter.api.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class CalculatorTest {  
  
 private Calculator calculator;  
  
 @BeforeEach  
 public void setUp() {  
 // Arrange: Initialize resources before each test  
 calculator = new Calculator();  
 }  
  
 @AfterEach  
 public void tearDown() {  
 // Cleanup resources  
 calculator = null;  
 }  
  
 @Test  
 public void testAddition() {  
 // Act  
 int result = calculator.add(2, 3);  
  
 // Assert  
 assertEquals(5, result);  
 }  
}

## 5. Output Verification

On running the tests, the output should indicate successful execution if assertions pass.

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0  
BUILD SUCCESS

## 6. Conclusion

Using the AAA pattern improves test readability, and setup/teardown methods help manage common test initialization and cleanup tasks efficiently.