**Exercise 1.4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit**

**CODE:**

**Calculator Class: -**

package practice.cog;  
  
public class Calculator {  
 public int add(int a,int b){  
 return a+b;  
 }  
 public int subtract(int a,int b){  
 return a-b;  
 }  
 public int multiply(int a,int b){  
 return a\*b;  
 }  
 public int divide(int a,int b){  
 if(b==0) throw new IllegalArgumentException("Cannot divide by zero");  
 return a/b;  
 }  
}

**CalculatorAAATest Class: -**

package practice.cog;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.*assertEquals*;  
  
public class CalculatorAaaTest {  
 private Calculator calculator;  
  
 @Before  
 public void setUp() {  
 calculator = new Calculator();  
 System.*out*.println("Setup complete");  
 }  
  
 @After  
 public void tearDown() {  
 System.*out*.println("Test finished");  
 }  
  
 @Test  
 public void testAdd\_AAA() {  
 // Arrange  
 int a = 4, b = 6;  
 // Act  
 int result = calculator.add(a, b);  
 // Assert  
 *assertEquals*(10, result);  
 }  
  
 @Test  
 public void testSubtract\_AAA() {  
 // Arrange  
 int a = 9, b = 4;  
 // Act  
 int result = calculator.subtract(a, b);  
 // Assert  
 *assertEquals*(5, result);  
 }  
}

**OUTPUT:**

A screenshot of a computer program

AI-generated content may be incorrect.