Week_2_ Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit

```java
1    import org.junit.jupiter.api.*;
2    import static org.junit.jupiter.api.Assertions.*;
3
4    class Calculator {
5        int add(int a, int b) {
6            return a + b;
7        }
8
9        int subtract(int a, int b) {
10           return a - b;
11       }
12
13       int multiply(int a, int b) {
14           return a * b;
15       }
16
17       int divide(int a, int b) {
18           if (b == 0)
19               throw new IllegalArgumentException(s:"Cannot divide by zero");
20           return a / b;
21       }
22   }
23
24   public class AssertionsTest {
25
26       Calculator calc;
27
28       @BeforeEach
29       void setUp() {
30           calc = new Calculator();
31       }
32
33       @AfterEach
34       void tearDown() {
35           calc = null;
36       }
37
38       @Test
39       void additionTest() {
40           int result = calc.add(a:10, b:5);
41           assertEquals(expected:15, result);
42       }
43
44       @Test
45       void subtractionTest() {
46           int result = calc.subtract(a:10, b:5);
47           assertEquals(expected:5, result);
48       }
49
50       @Test
51       void multiplicationTest() {
52           int result = calc.multiply(a:3, b:4);
53           assertEquals(expected:12, result);
54       }
55
56       @Test
57       void divisionTest() {
58           int result = calc.divide(a:20, b:4);
59           assertEquals(expected:5, result);
60       }
61
62       @Test
63       void divideByZeroTest() {
64           assertThrows(expectedType:IllegalArgumentException.class, () -> calc.divide(a:10, b:0));
65       }
66   }
67
```