**EXERCISE 1: SETTING UP JUNIT**

MAIN CLASS-setup.java

public class setup{

    public String message(){

        return "I am Dinesh";

    }

}

TEST CLASS-setuptest.java

import org.junit.\*;

public class setuptest {

    @Test

    public void testing(){

        setup s=new setup();

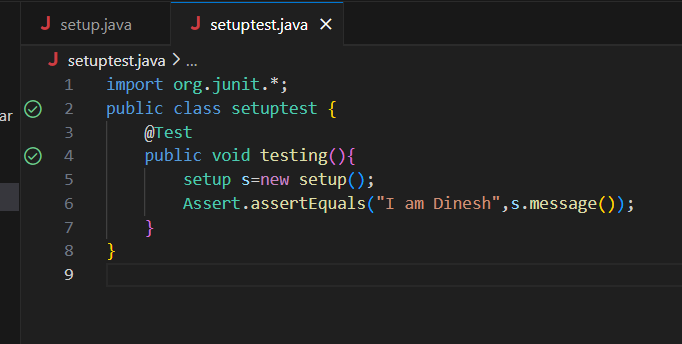
        Assert.assertEquals("I am Dinesh",s.message());

    }

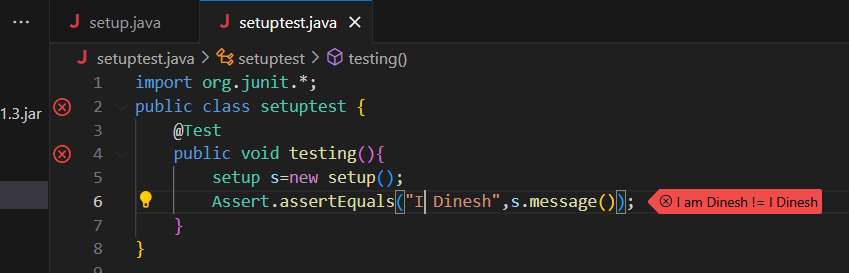
}

**EXECUTION**

SUCCESS ATTEMPT



FAILURE ATTEMPT



**EXERCISE 3: ASSERTIONS IN JUNIT**

MAIN CLASS-Assertions.java

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

public class AssertionsTest {

    @Test

    public void testEquals() {

        assertEquals(5, 2 + 3);

    }

    @Test

    public void testTrue() {

        assertTrue(5 > 3);

    }

    @Test

    public void testFalse() {

        assertFalse(5 < 3);

    }

    @Test

    public void testNull() {

        assertNull(null);

    }

    @Test

    public void testNotNull() {

        assertNotNull(new Object());

    }

    @Test

    public void testArrayEquals() {

        int[] expected = {1, 2, 3};

        int[] actual = {1, 2, 3};

        assertArrayEquals(expected, actual);

    }

    @Test

    public void testSame() {

        String str = "JUNIT";

        String ref = str;

        assertSame(str, ref);

    }

}

**EXECUTION**

SUCCESS ATTEMPT-ALL CASES PASSED



FAILURE ATTEMPT-FEW CASES FAILED



EXERCISE 4: ARRANGE-ACT-ASSERT (AAA) PATTERN, TEST FIXTURES, SETUP AND TEARDOWN METHODS IN JUNIT

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.\*;

public class aaapattern {

    private Calculator calculator;

    @BeforeEach

    public void setUp() {

        // Arrange

        calculator = new Calculator();

        System.out.println("Setup completed");

    }

    @AfterEach

    public void tearDown() {

        calculator = null;

        System.out.println("Teardown completed");

    }

    @Test

    public void testAddition() {

        // Act

        int result = calculator.add(10, 5);

        // Assert

        assertEquals(15, result);

    }

    @Test

    public void testSubtraction() {

        int result = calculator.subtract(10, 3);

        assertEquals(7, result);

    }

}

class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

    public int subtract(int a, int b) {

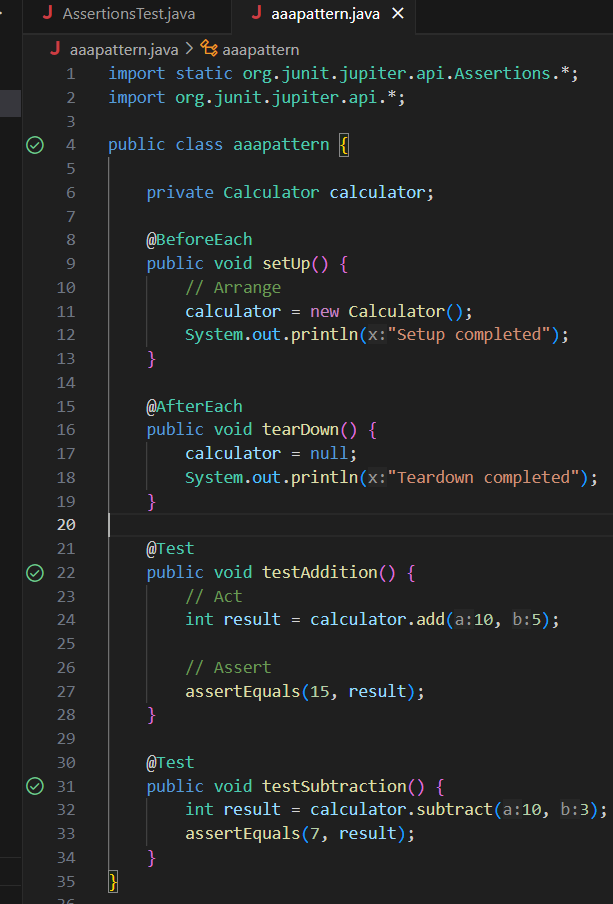
        return a - b;

    }

}

**EXECUTION**

SUCCESS ATTEMPT-ALL CASES PASSED



FAILURE ATTEMPT

