Write a program to demonstrate nested and repeated tests

Pom.xml:

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>UsingJUnit</groupId>
  <artifactId>UsingJUnit</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencies>
      <dependency>
          <groupId>org.junit.jupiter</groupId>
          <artifactId>junit-jupiter-engine</artifactId>
          <version>5.4.2</version>
      </dependency>

       <dependency>
      <groupId>org.junit.platform</groupId>
      <artifactId>junit-platform-launcher</artifactId>
      <version>1.2.0</version>
    </dependency>

  </dependencies>
</project>
```

Calculator.java:

```java
package com.ecommerce.tests;

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

NestedCases.java:

```java
package com.ecommerce.tests;

import org.junit.jupiter.api.*;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;


@DisplayName("JUnit 5 Nested Example")

public class NestedCases {

    @BeforeAll
    static void beforeAll() {
```

```java
        System.out.println("Before all test methods");
    }

    @BeforeEach
    void beforeEach() {
        System.out.println("Before each test method");
    }

    @AfterEach
    void afterEach() {
        System.out.println("After each test method");
    }

    @AfterAll
    static void afterAll() {
        System.out.println("After all test methods");
    }

    @Nested
    @DisplayName("Tests for the method A")
    class A {

        @BeforeEach
        void beforeEach() {
            System.out.println("Before each test method of the A class");
        }

        @AfterEach
        void afterEach() {
            System.out.println("After each test method of the A class");
        }

        @Test
        @DisplayName("Example test for method A")
        void sampleTestForMethodA() {
            System.out.println("Example test for method A");
        }

        @Nested
        @DisplayName("When X is true")
        class WhenX {

            @BeforeEach
            void beforeEach() {
                System.out.println("Before each test method of the WhenX
class");
            }

            @AfterEach
            void afterEach() {
                System.out.println("After each test method of the WhenX
class");
            }

            @Test
            @DisplayName("Example test for method A when X is true")
            void sampleTestForMethodAWhenX() {
                System.out.println("Example test for method A when X is
true");
            }
        }
```

```
        }
}
```

RepeatedTests.java:

```java
package com.ecommerce.tests;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayName;
import org. import org.junit.jupiter.api.AfterAll;
junit.jupiter.api.RepeatedTest;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.TestInfo;

@DisplayName("JUnit 5 Repeated Tests Example")

public class RepeatedTests {
        @BeforeAll
            public static void init(){
                System.out.println("Before All init() method called");
            }

            @BeforeEach
            public void initEach(){
                System.out.println("Before Each initEach() method called");
            }

            @Test
            @DisplayName("Add operation test")
            @RepeatedTest(5)
            void addNumber(TestInfo testInfo) {
                Calculator calculator = new Calculator();
                Assertions.assertEquals(2, calculator.add(1, 1), "1 + 1
should equal 2");
                System.out.println("===addNumber testcase executed===");
            }

            @AfterEach
            public void cleanUpEach(){
                System.out.println("After Each cleanUpEach() method
called");
            }

            @AfterAll
            public static void cleanUp(){
                System.out.println("After All cleanUp() method called");
            }
}
```