Code For CalculatorTests.cs in CalcLibrary.cs

using NUnit.Framework;

using CalcLibrary;

using System;

namespace CalcLibrary.Tests

{

[TestFixture] // Marks class for testing

public class CalculatorTests

{

private IMathLibrary calculator;

[SetUp] // Runs before each test

public void Setup()

{

calculator = new SimpleCalculator(); // Loosely coupled - use interface

}

[TearDown] // Runs after each test

public void Teardown()

{

((SimpleCalculator)calculator).AllClear();

}

[Test] // Single test

public void Addition\_ShouldReturnCorrectResult()

{

var result = calculator.Addition(10, 5);

Assert.That(result, Is.EqualTo(15));

}

[Test] // Test with multiple inputs

[TestCase(2, 3, 5)]

[TestCase(-1, 1, 0)]

[TestCase(0, 0, 0)]

[TestCase(-5, -3, -8)]

public void Addition\_WithMultipleInputs\_ShouldReturnCorrectSum(double a, double b, double expected)

{

var result = calculator.Addition(a, b);

Assert.That(result, Is.EqualTo(expected));

}

[Test]

[TestCase(10, 5, 5)]

public void Subtraction\_ShouldReturnCorrectResult(double a, double b, double expected)

{

var result = calculator.Subtraction(a, b);

Assert.That(result, Is.EqualTo(expected));

}

[Test]

[TestCase(4, 5, 20)]

public void Multiplication\_ShouldReturnCorrectResult(double a, double b, double expected)

{

var result = calculator.Multiplication(a, b);

Assert.That(result, Is.EqualTo(expected));

}

[Test]

[TestCase(10, 2, 5)]

public void Division\_ShouldReturnCorrectResult(double a, double b, double expected)

{

var result = calculator.Division(a, b);

Assert.That(result, Is.EqualTo(expected));

}

[Test]

public void Division\_ByZero\_ShouldThrowException()

{

Assert.Throws<ArgumentException>(() => calculator.Division(10, 0));

}

[Test]

[Ignore("This test is not implemented yet")]

public void Percentage\_ShouldReturnCorrectResult()

{

// Placeholder test

}

}

}



