N-unit Testing:

Calculator Logic:

namespace CalcLibrary

{

public class Calculator

{

public int Add(int a, int b) => a + b;

public int Subtract(int a, int b) => a - b;

}

}

Nunit Tests:

using NUnit.Framework;

using CalcLibrary;

namespace CalcLibrary.Tests

{

[TestFixture]

public class CalculatorTests

{

private Calculator \_calculator;

[SetUp]

public void Setup()

{

\_calculator = new Calculator();

}

[TearDown]

public void Teardown()

{

\_calculator = null;

}

[Test]

[TestCase(2, 3, 5)]

[TestCase(-1, -2, -3)]

[TestCase(0, 0, 0)]

public void Add\_WhenCalled\_ReturnsExpectedSum(int a, int b, int expected)

{

var result = \_calculator.Add(a, b);

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

}

[Test]

[TestCase(5, 3, 2)]

[TestCase(0, 0, 0)]

[TestCase(-5, -5, 0)]

public void Subtract\_WhenCalled\_ReturnsExpectedDifference(int a, int b, int expected)

{

var result = \_calculator.Subtract(a, b);

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

}

}

}

Output:

