**Hands-On: Stage 3 – .NET Advanced Features** ï **NUnit Day 1 - Session 1**

Follow the steps listed below to write the NUnit test cases for the application.

· Create a Unit test project(.NET Framework) in the solution provided.

· Add the CalcLibrary project as reference

· Create a class “CalculatorTests” to write all the test cases for the methods in the solution

· Use the ‘TestFixture’, ‘SetUp’ and ‘TearDown’ attributes, to declare, initialize and cleanup activities respectively

· Create a Test method to check the addition functionality

· Use the ‘TestCase’ attribute to send the inputs and the expected result

· Use Assert.That to check the actual and expected result match

PROJECT CODE:

MathLibrary.cs

using System;

namespace CalcLibrary

{

public class SimpleCalculator : IMathLibrary

{

double result = 0;

public double Addition(double a, double b)

{

result = a + b;

return result;

}

public double Subtraction(double a, double b)

{

result = a - b;

return result;

}

public double Multiplication(double a, double b)

{

result = a \* b;

return result;

}

public double Division(double a, double b)

{

if (b == 0)

throw new ArgumentException("Second Parameter Can't be Zero");

result = a / b;

return result;

}

public void AllClear()

{

result = 0;

}

public double GetResult

{

get { return result; }

}

}

}

IMathLibrary.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace CalcLibrary

{

interface IMathLibrary

{

double Addition(double a, double b);

double Subtraction(double a, double b);

double Multiplication(double a, double b);

double Division(double a, double b);

}

}

UnitTest1.cs

using NUnit.Framework;

using CalcLibrary;

namespace CalcLibrary

{

[TestFixture]

public class CalculatorTests

{

[Test]

public void Addition\_InputTwoDoubleValues\_ReturnsSum()

{

SimpleCalculator mathLibrary = new SimpleCalculator();

//Arrange

double expectedResult = 15;

//Act

double Result = mathLibrary.Addition(5, 10);

//Assert

Assert.AreEqual(expectedResult, Result);

}

[Test]

public void Subtraction\_InputTwoDoubleValues\_ReturnsDifference()

{

SimpleCalculator mathLibrary = new SimpleCalculator();

double expectedResult = 5;

double Result = mathLibrary.Subtraction(10, 5);

Assert.AreEqual(expectedResult, Result);

}

[Test]

public void Multiplication\_InputTwoDoubleValues\_ReturnsProduct()

{

SimpleCalculator mathLibrary = new SimpleCalculator();

double expectedResult = 50;

double Result = mathLibrary.Multiplication(10, 5);

Assert.AreEqual(expectedResult, Result);

}

[Test]

public void Division\_InputTwoDoubleValues\_ReturnsQuotient()

{

SimpleCalculator mathLibrary = new SimpleCalculator();

double expectedResult = 2;

double Result = mathLibrary.Division(10, 5);

Assert.AreEqual(expectedResult, Result);

}

}

}

OUTPUT :