AIM: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.

**CalcLibrary.cs:**

namespace CalcLibrary

{

    public class Calculator

    {

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

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

    }

}

**CalculatorTest.cs:**

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)

        {

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

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

        }

    }

}

**UNITTEST1.CS**

using NUnit.Framework;

using Moq;

using CustomerCommLib;

namespace CustomerCommLib.Tests

{

    public class CustomerCommTests

    {

        [Test]

        public void SendMailToCustomer\_ShouldReturnTrue\_WhenMailIsSentSuccessfully()

        {

            var mockMailSender = new Mock<IMailSender>();

            mockMailSender

                .Setup(ms => ms.SendMail(It.IsAny<string>(), It.IsAny<string>()))

                .Returns(true);

            var customerComm = new CustomerComm(mockMailSender.Object);

            var result = customerComm.SendMailToCustomer();

            Assert.IsTrue(result);

            mockMailSender.Verify(ms => ms.SendMail("cust123@abc.com", "Some Message"), Times.Once);

        }

    }

}

**OUTPUT:**

