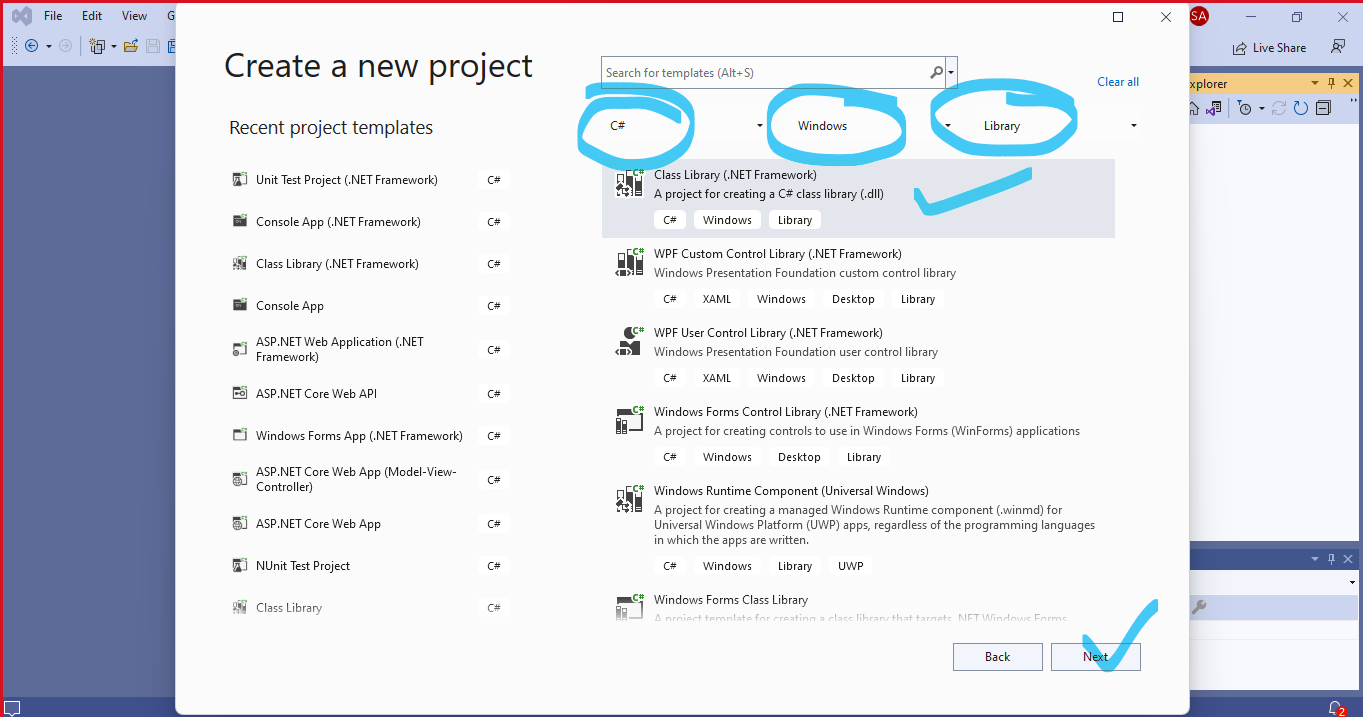
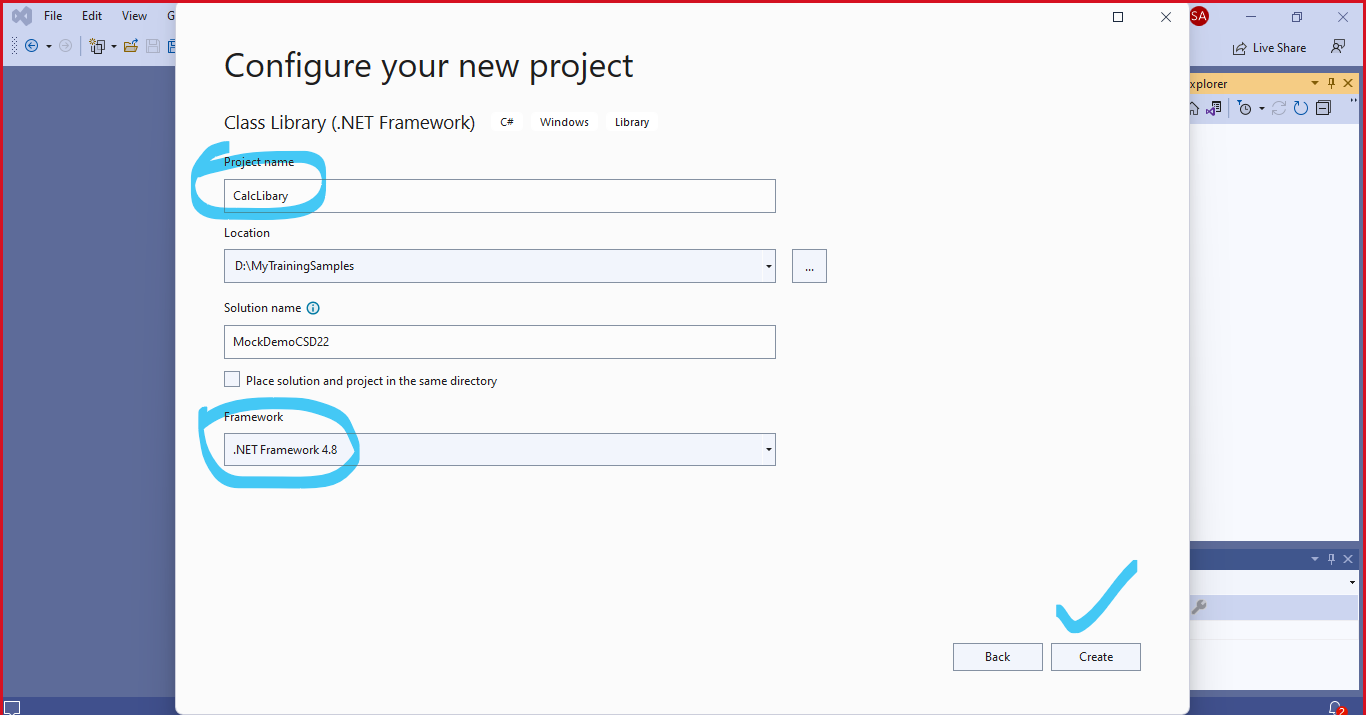
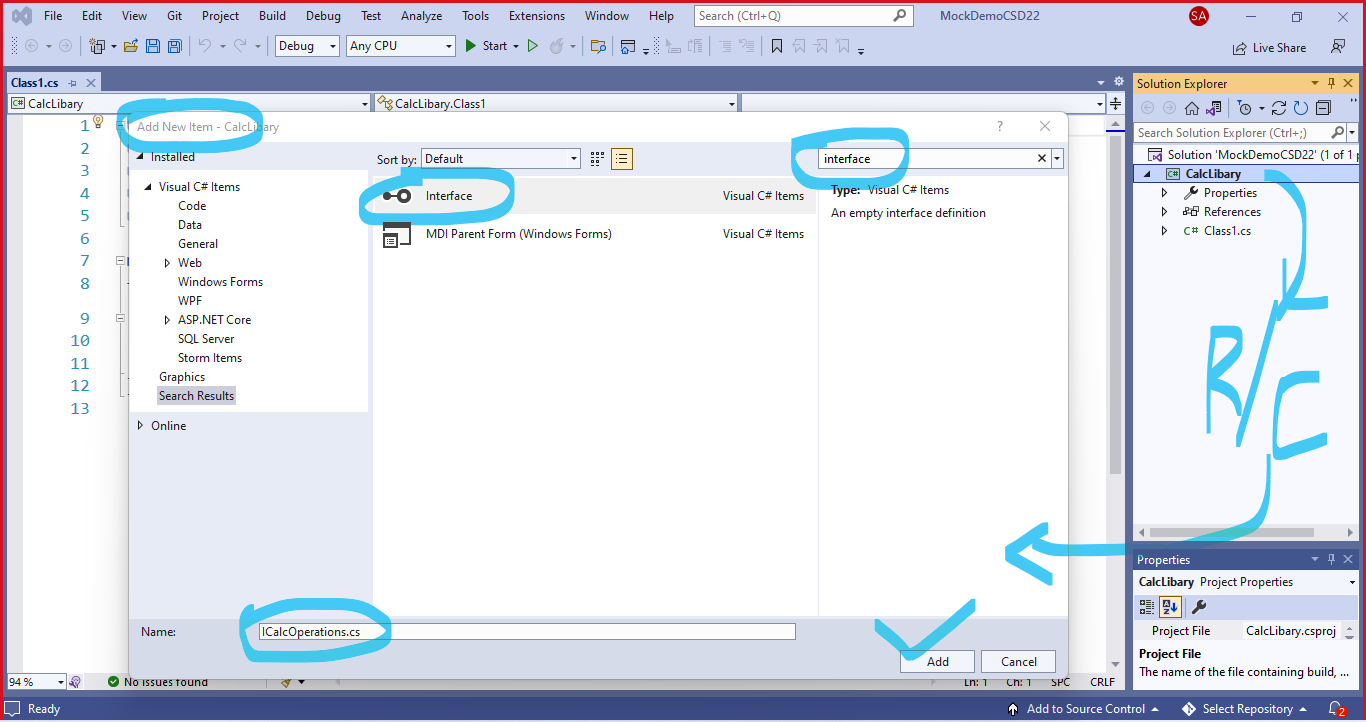
Step1 (Add a class library, Do the full implementation)



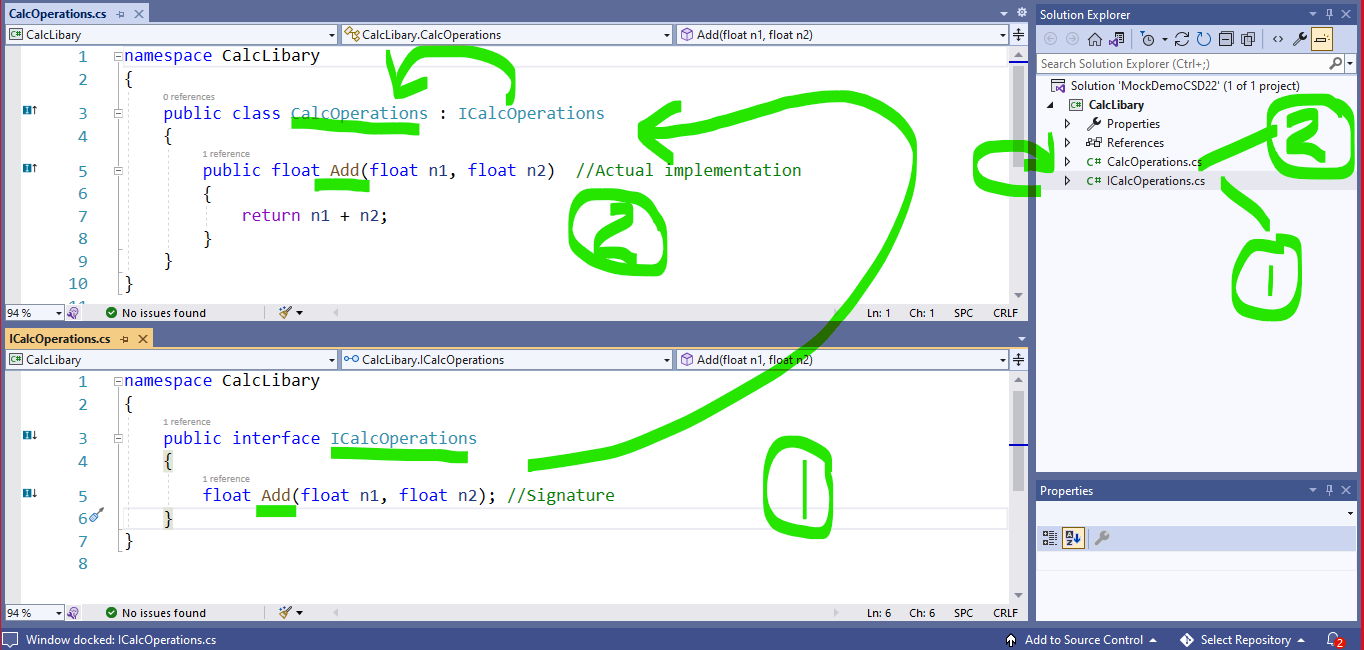
Step 2



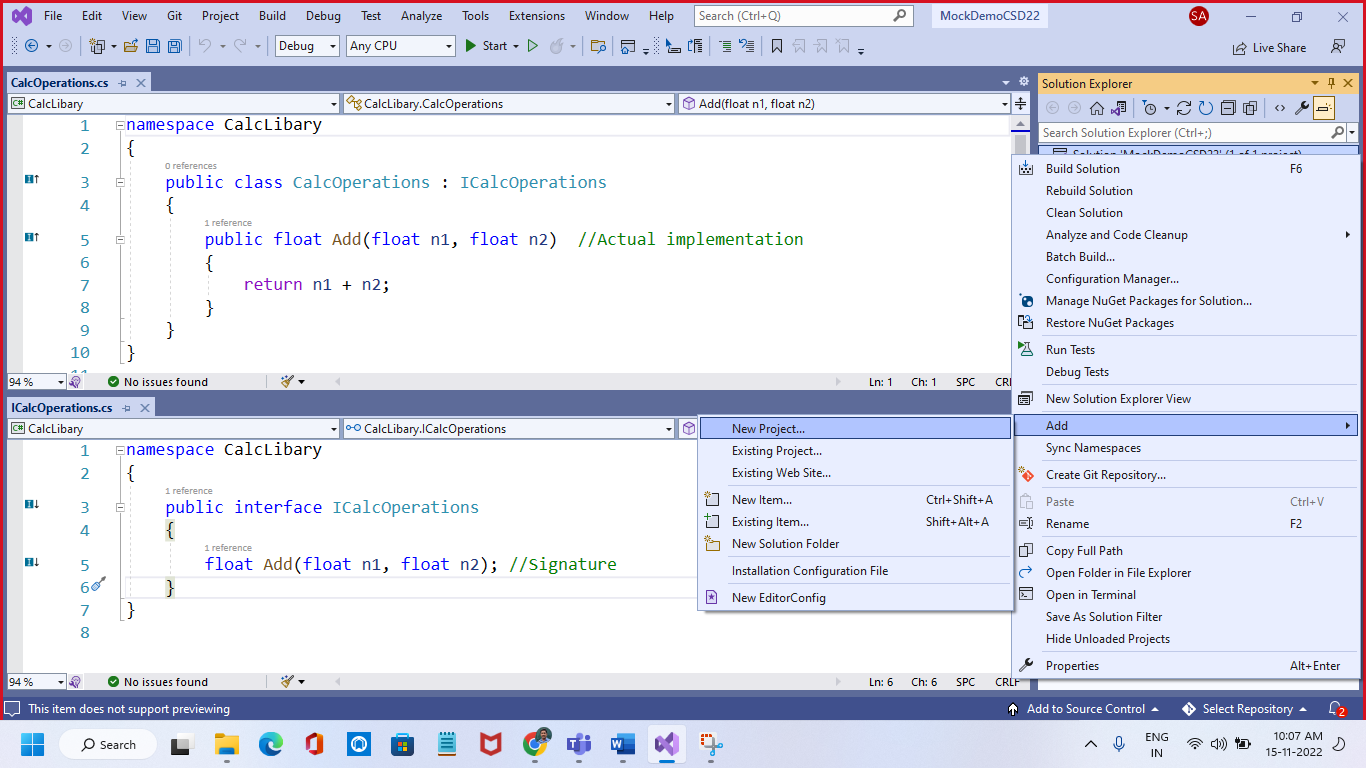
Step 3

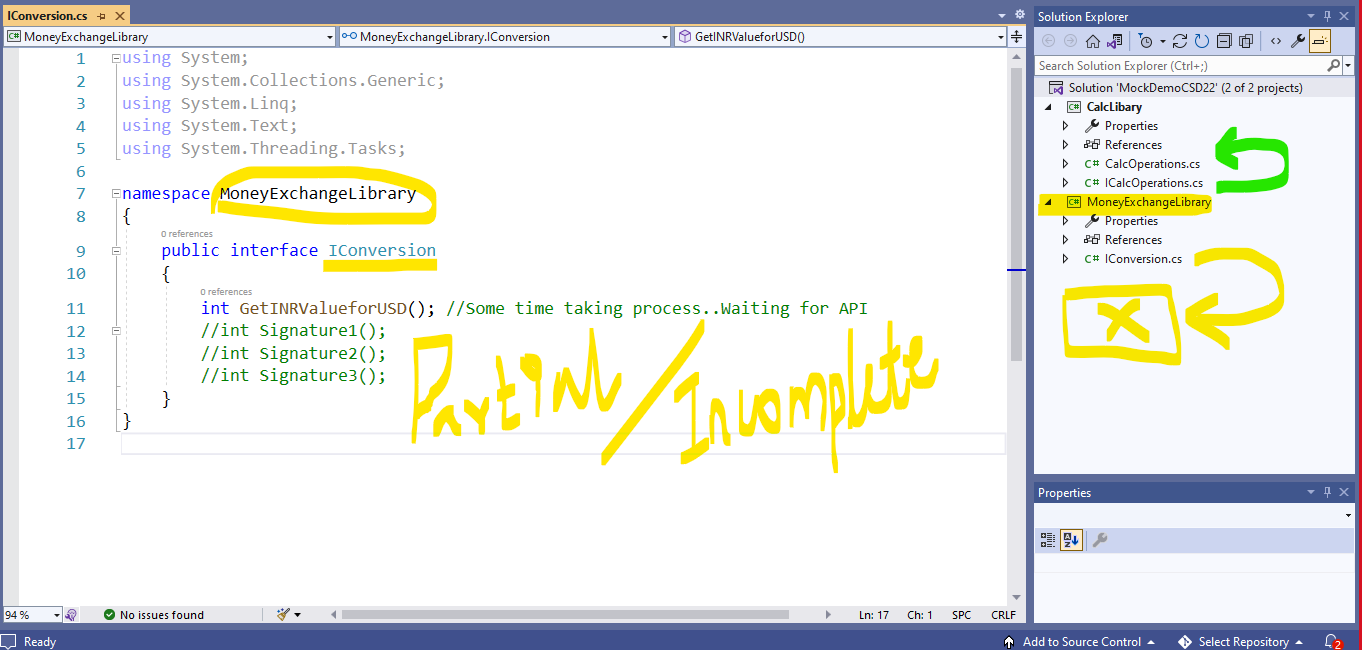


Step 4



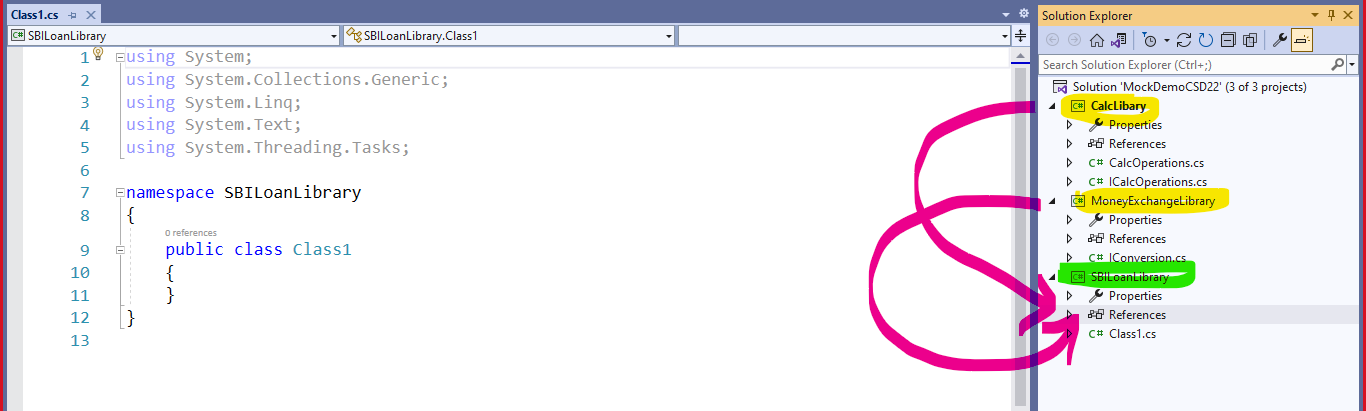
Step 5: Add another Class Library Project **(To demonstrate Partial Implementation)**

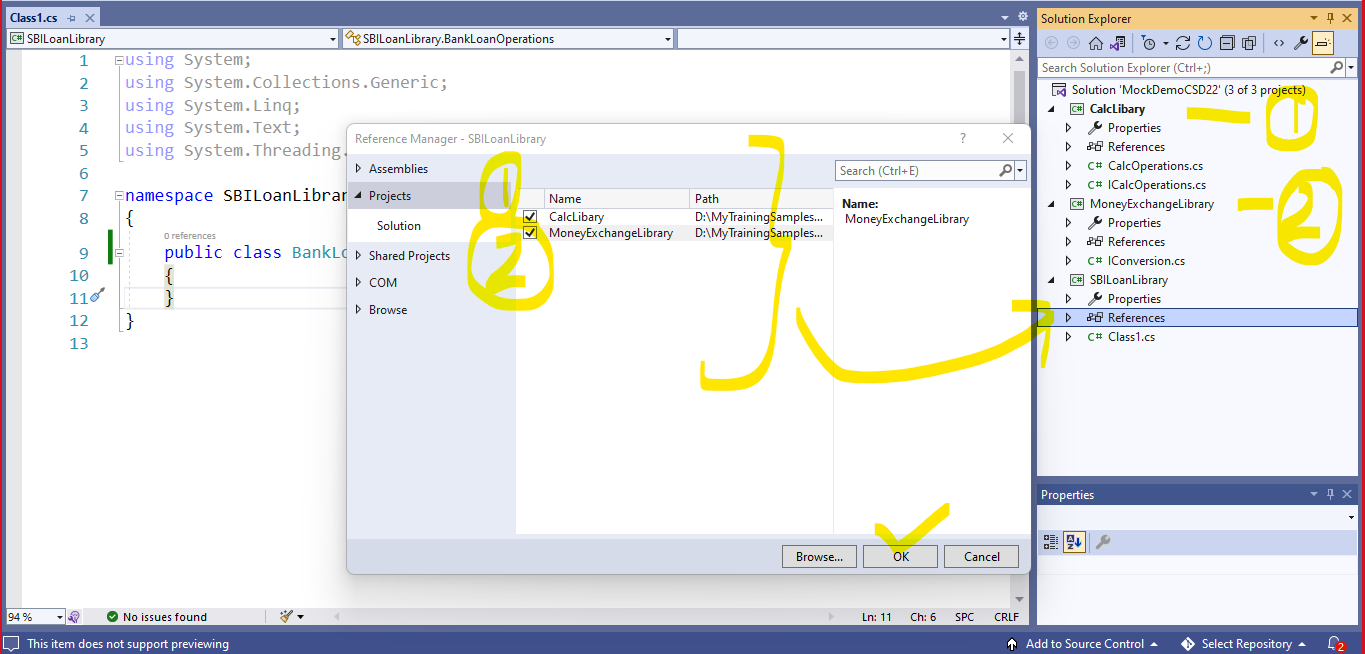




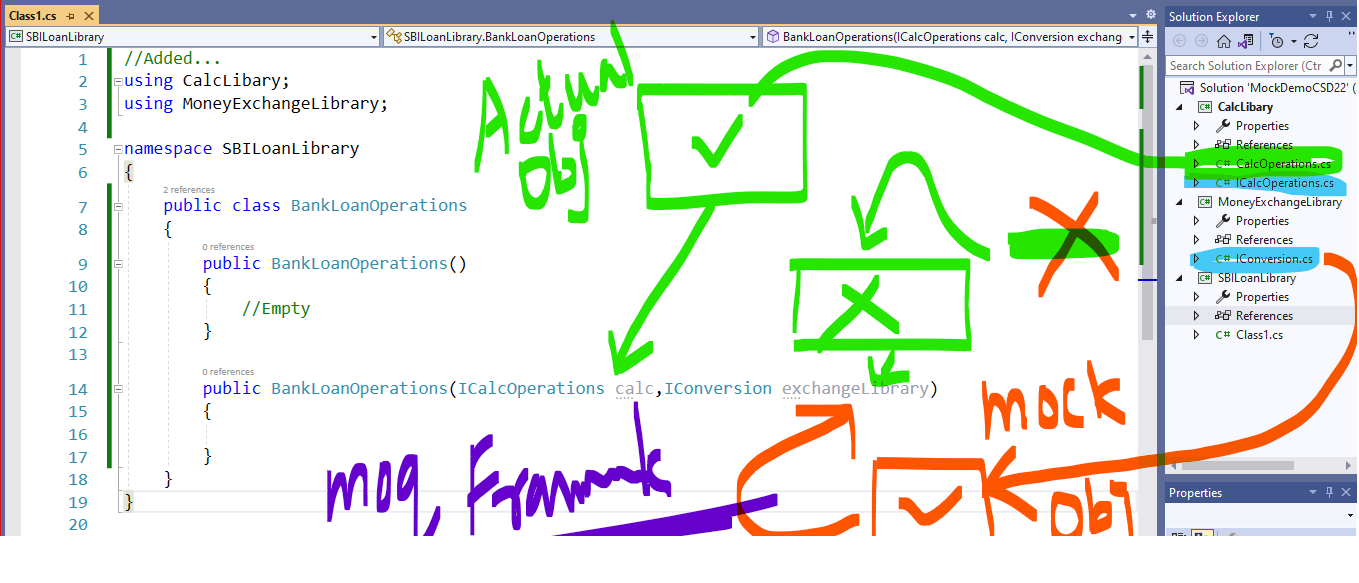
Step6: Add another Library Project (To Consume above 2 Projects – Fully and Partially implemented)

After creating project, Create References to existing projects – CalcLibary, MoneyExchangeLibrary.

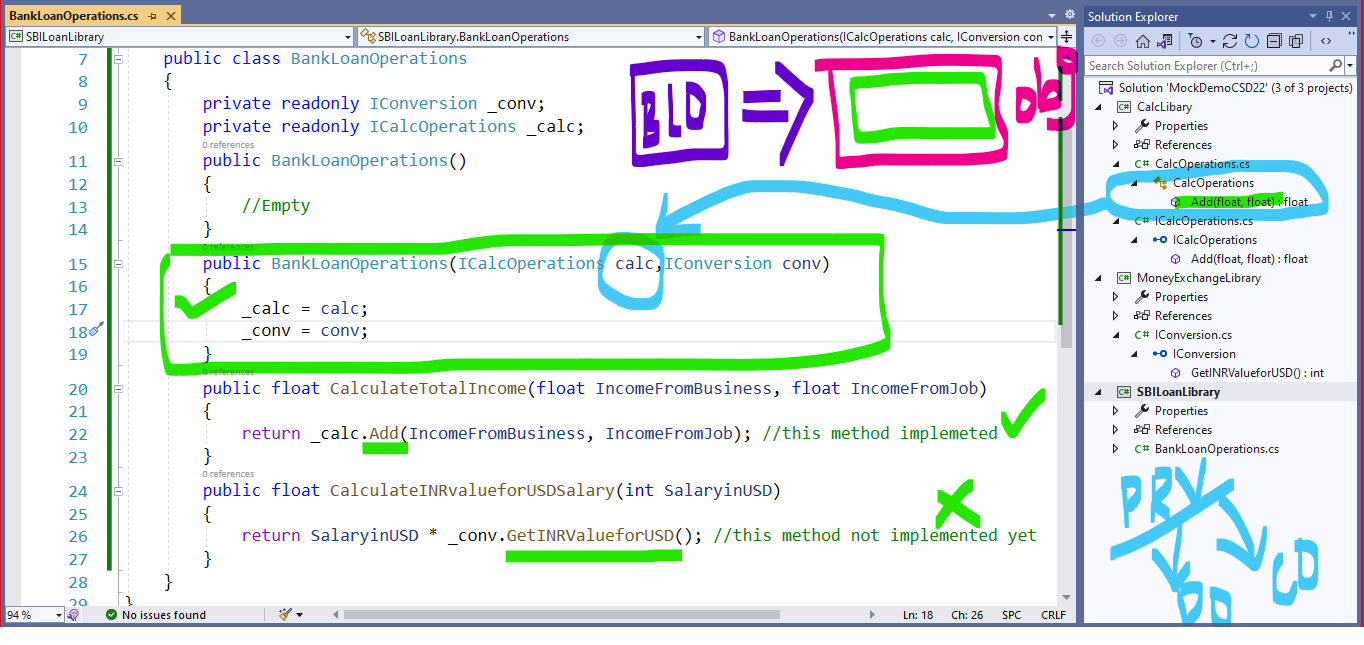




Step 7: Include added /referred projects as below and do dependency injection there as below



Step8: Do the following implementation

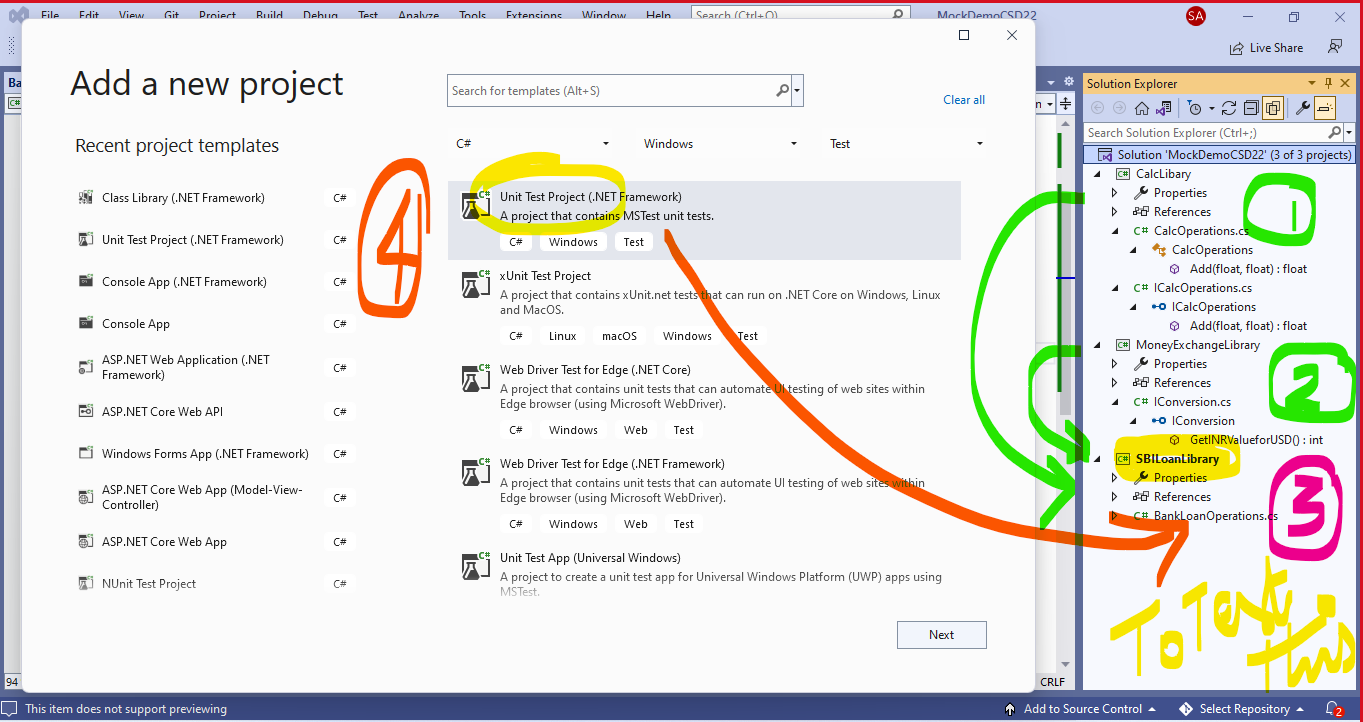


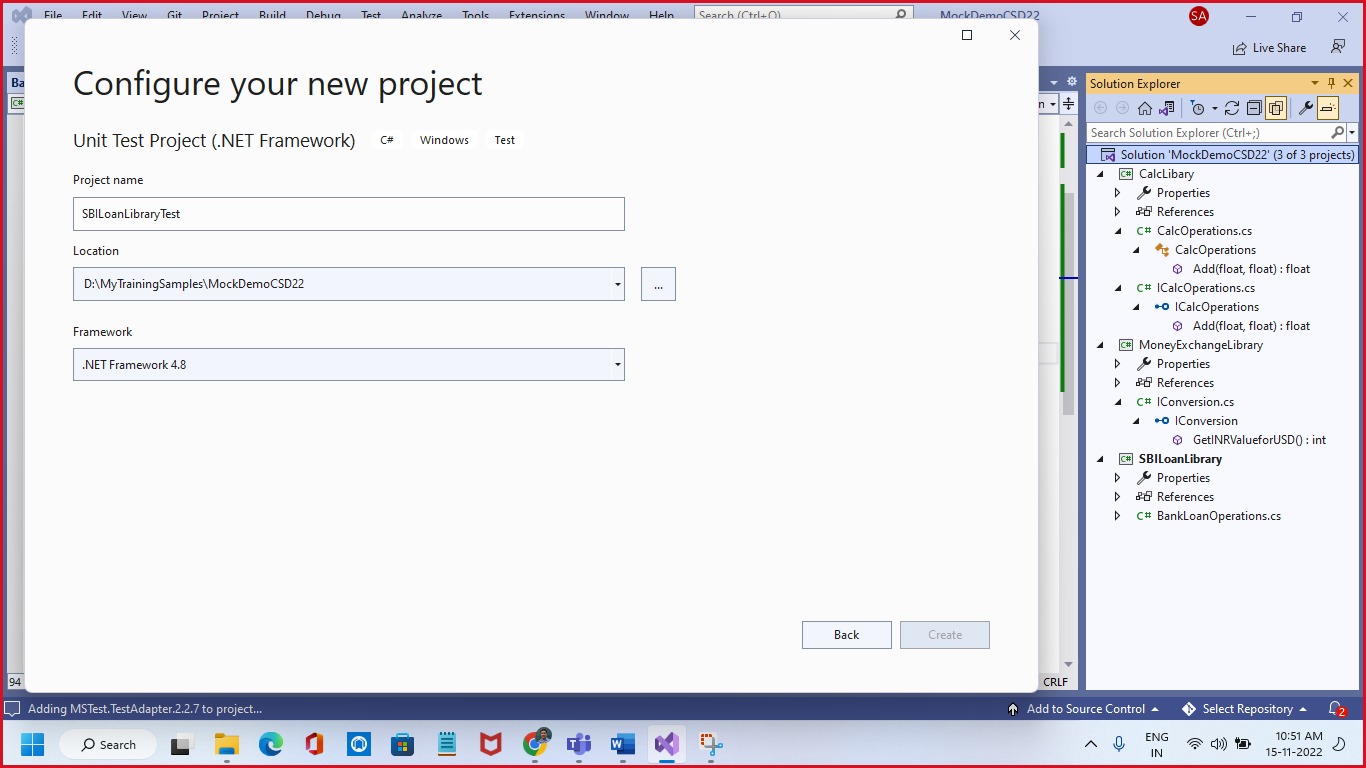
Note: A parent reference variable can point derived type object. Here calc, conv both are parent type (Interface Type) can refer object of derived classes

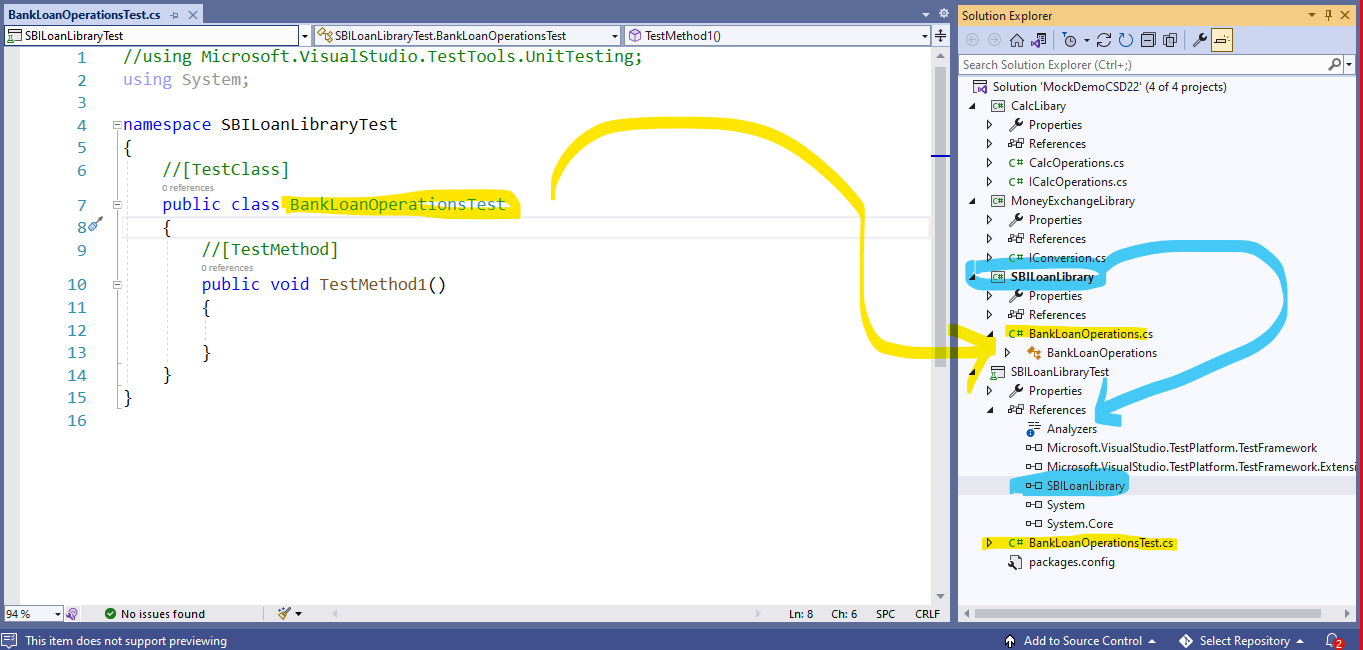
Recommended Reading

<https://syed-nawas-codes.blogspot.com/p/runtime-polymorphism-by-virtual-methods.html>

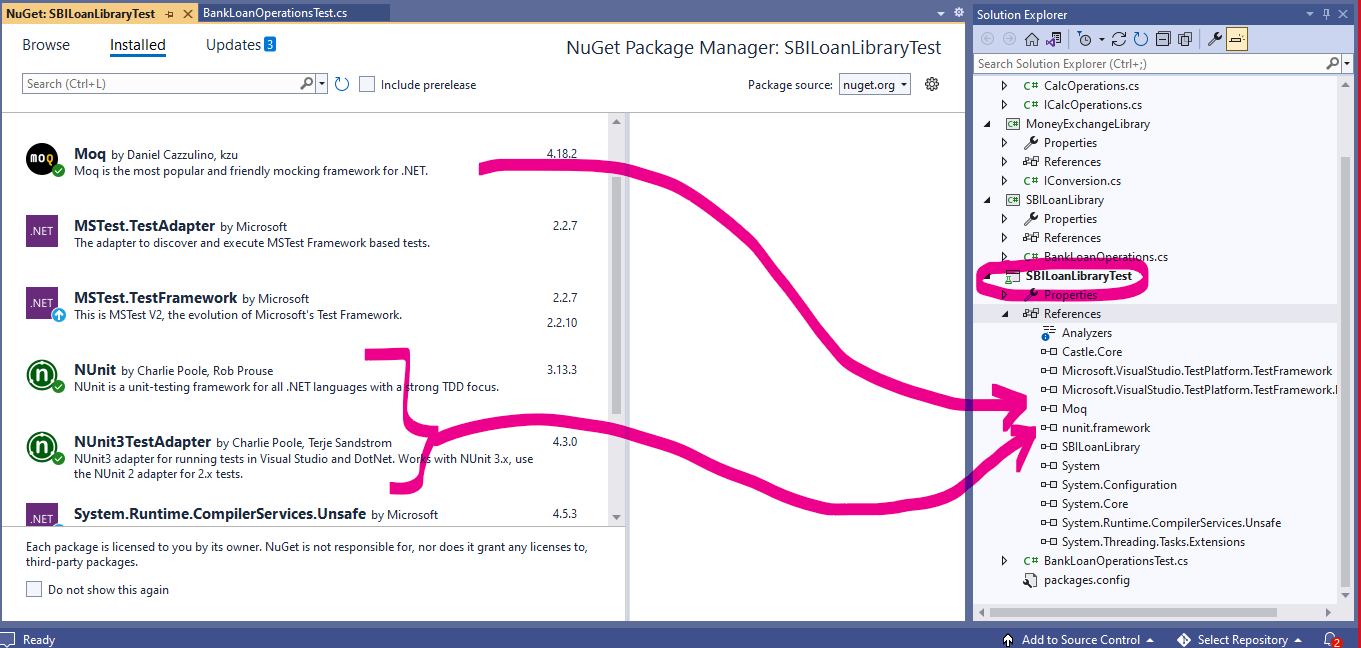
Step 9: Add a testing project to test above BankLibrary Project

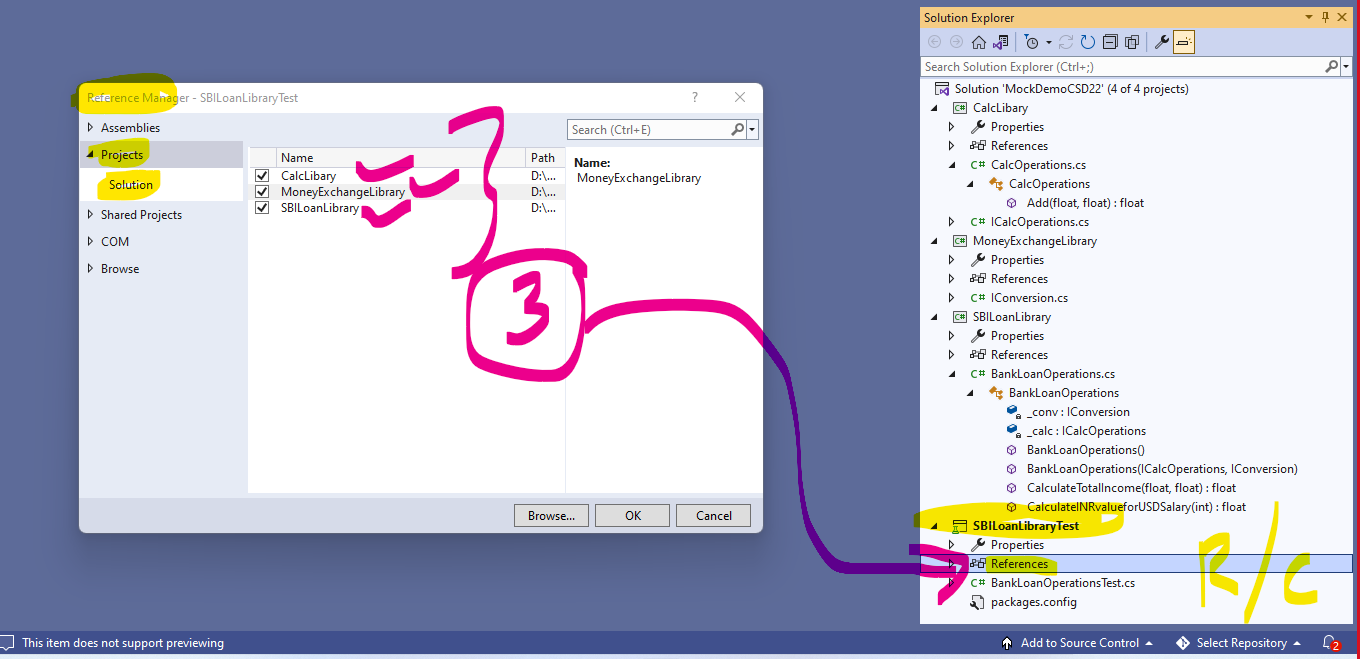






Step 10: Add NuGet Packages for NUnit, Nunit3TestAdaptor, and Moq Framework





Step 11: add the following code in BankLoanOperationsTest.cs

//using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

//Added...

using NUnit.Framework;

using Moq;

//Addd...

using CalcLibary;

using MoneyExchangeLibrary;

using SBILoanLibrary;

namespace SBILoanLibraryTest

{

//[TestClass]

[TestFixture]

public class BankLoanOperationsTest

{

//[TestMethod]

[Test]

public void CalculateTotalIncome\_Accepts2Incomes\_ReturnsTotalIncome()

{

//Arrange

CalcOperations calc = new CalcOperations();

int ExpectedResult = 20000;

//Act

BankLoanOperations ops1 = new BankLoanOperations(calc, null);

float ActualResult = ops1.CalculateTotalIncome(10000, 10000);

//Assert

Assert.AreEqual(ExpectedResult, ActualResult);

}

[Test]

public void CalculateINRvalueforUSDSalary\_AcceptsSalaryinUSD\_ReturnINRValue()

{

//Arrage

Mock<IConversion> mockObject = new Mock<IConversion>();

mockObject.Setup(meth => meth.GetINRValueforUSD()).Returns(80); //for Time being It will return 80

//mockObject.Setup(meth => meth.Signature1()).Returns(1);

//mockObject.Setup(meth => meth.Signature2()).Returns(2);

//mockObject.Setup(meth => meth.Signature3()).Returns(3);

float expectedResult = 80000.0f;

//Act

BankLoanOperations ops = new BankLoanOperations(null, mockObject.Object);

float actualResult = ops.CalculateINRvalueforUSDSalary(1000);

//Assert

Assert.AreEqual(expectedResult, actualResult);

}

}

}

Note If needed downgrade the Packages suitable for .net framework 4.8

