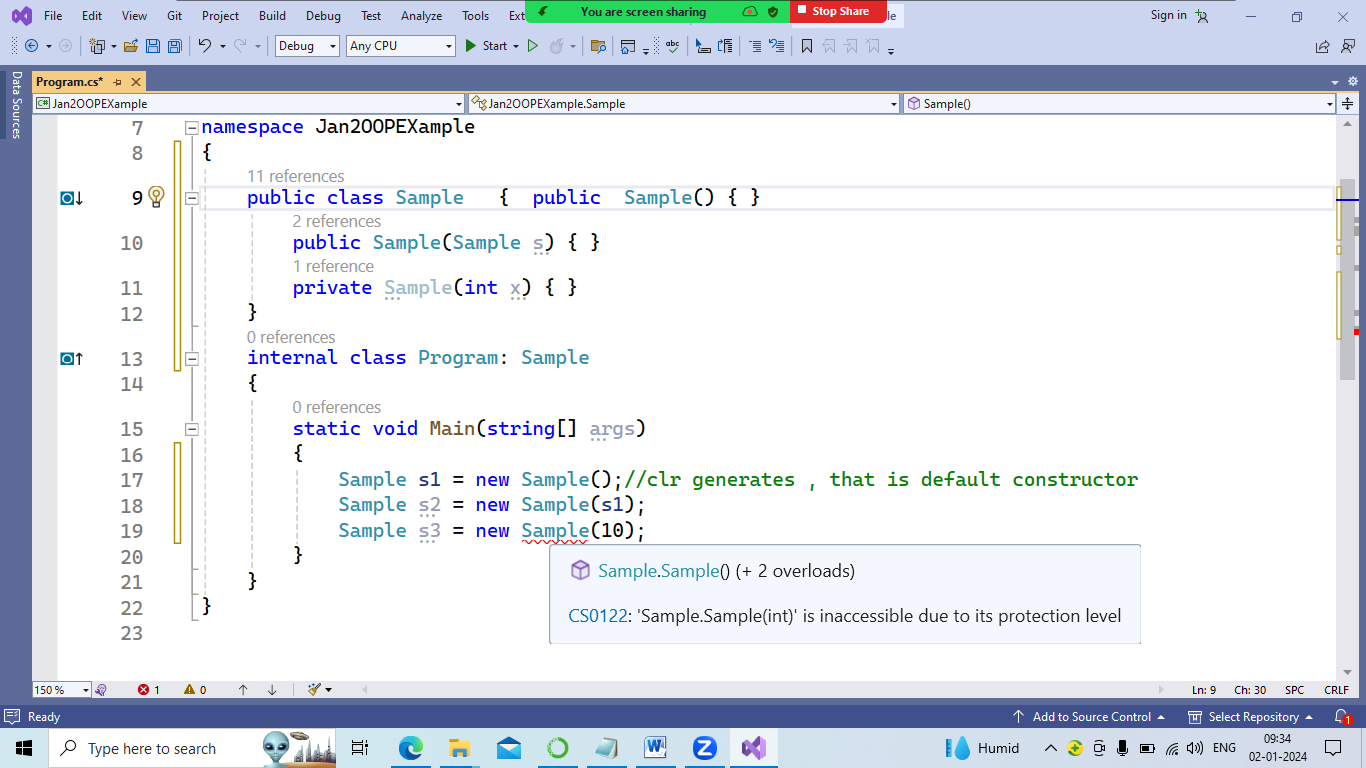
Are we allowed to define constructors in abstract class ?

If yes🡪 what type of constructors are allowed ?

this keyword usage

a)

b)



Main() of PL is predefined and fixed definition.

int Main() ,

Please check , what types of access specifers is allowed while declaring a class?

And methods also?

With property we use “value” which is builtin keyword to represent a value assigned in set block

In case of constructors when to use

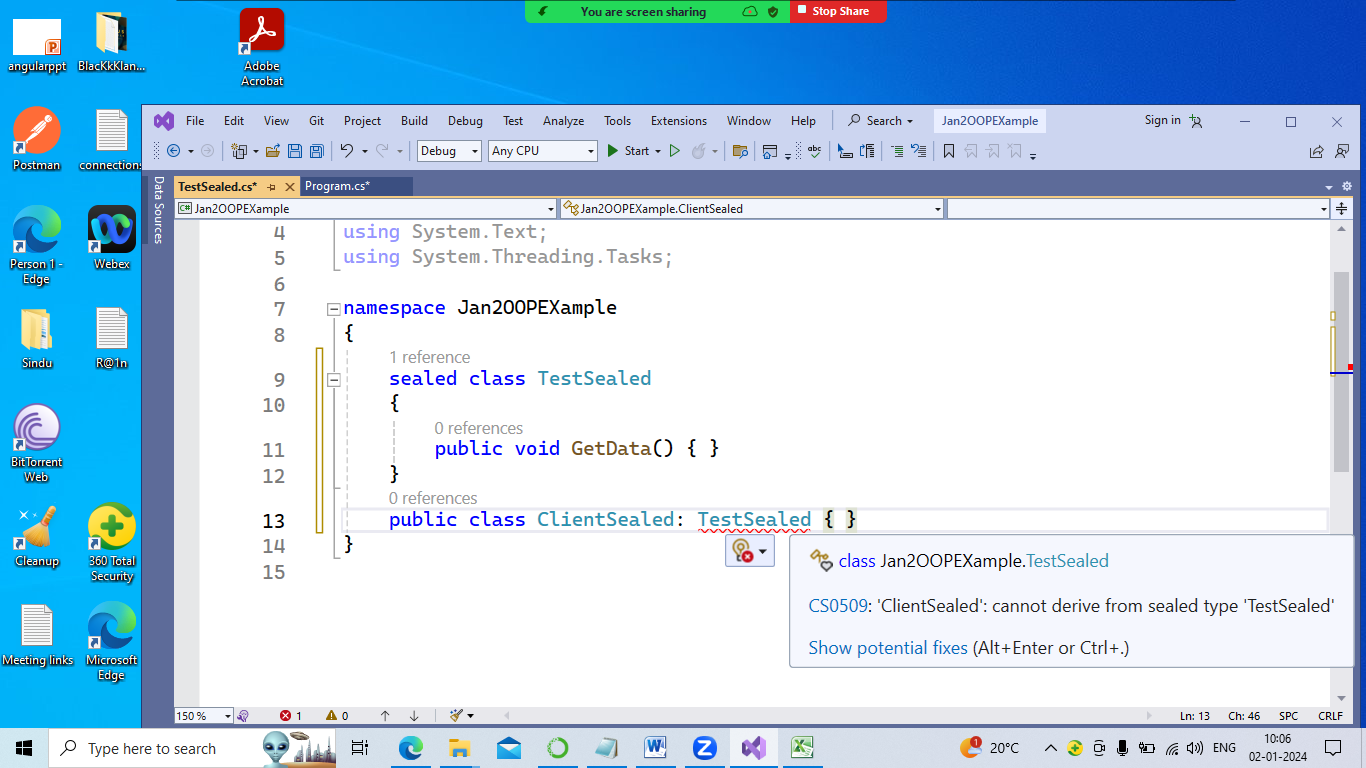
|  |  |
| --- | --- |
| this | base |
|  |  |

Sealed class: all value types are sealed classes.

Asealed class means , class can not be inherited.

It is last class in hiarch .

Keyword is “sealed” used to declare a class



Please check sealed class allows protected also for methods

Will sealed class allows constructors? Yes or no?

What types are allowed ?

=============== static ========

Static methods()

Static constructor

Also completed

|  |  |
| --- | --- |
| Static class | Private class |
| Class allows all mebers declared as static only. |  |
| Non-static members are not allowed | Members are non-static type |
| Not allowed to create object | Allowed to create object |
| What typ of constructors are allowed ? | What type of constructors are allowed ? |
|  |  |

Partial keyword is used to define a class or method.

Partial means not completed , still more definition may come

When we require define a class as “partial” ?

Multiple developers are working on same class in their systems.

So when we merge whole project , same class name is used in given code by developers?

So by declaring as partial class , it is easily merged as single unit or class

One instance is created to call methods declared in all partial classes

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Jan2OOPEXample

{

static class Product {

static void Display() { }

}

partial class Products {// dev 1 in his system

public void Display() { }

}

partial class Products { }// another dev

partial class Products { // team lead who is testing and merging

static void Main(string[] args)

{

Products ps = new Products();

ps.Display();

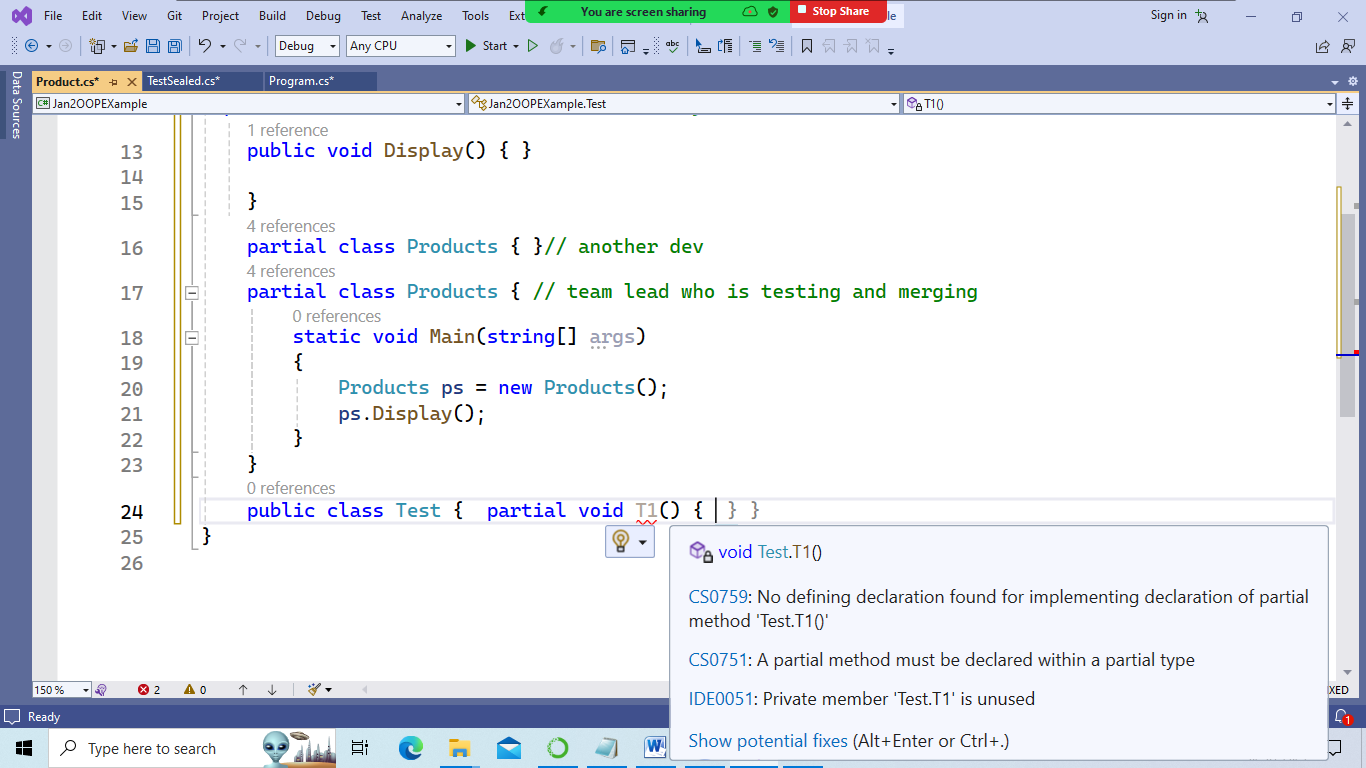
}

}

}

What are Partial Methods In C#?

Ready has the concept of partial classes in C#, butYes, we can have Partial Methods in C#. This concept also uses the Partial keyword but there are some restrictions/guidelines with the use of the Partial Methods.



a list of attributes of a partial method.

1. A partial method can only be created in partial classes or partial structs.
2. In order to create a partial method, it must be declared first(like an abstract method), with a signature only and no definition. After it is declared, its body can be defined in the same component or a different component of the partial class/struct .
3. A partial method is implicitly private. It cannot have any other access modifier.
4. A partial method can only have a void return type.

class Program

{

static void Main(string[] args)

{

ClassA cls = new ClassA();

cls.SetSalary();

}

}

public partial class ClassA

{

partial void SetData();

}

public partial class ClassA

{

public void SetSalary()

{

SetData();

// Perform other functionality here

}

partial void SetData()

{

Console.Write("This is a partial method.");

Console.ReadKey();

}

}

A partial class allows to work on same class name with all required function

1 dev 🡪partial class bank{ current acc}

2)dev 🡪public class bank{saving acc }

3) dev –public clss bank{recurring acc}

Class Bank

{

Svm()

{

Bank b = new bank();

b.

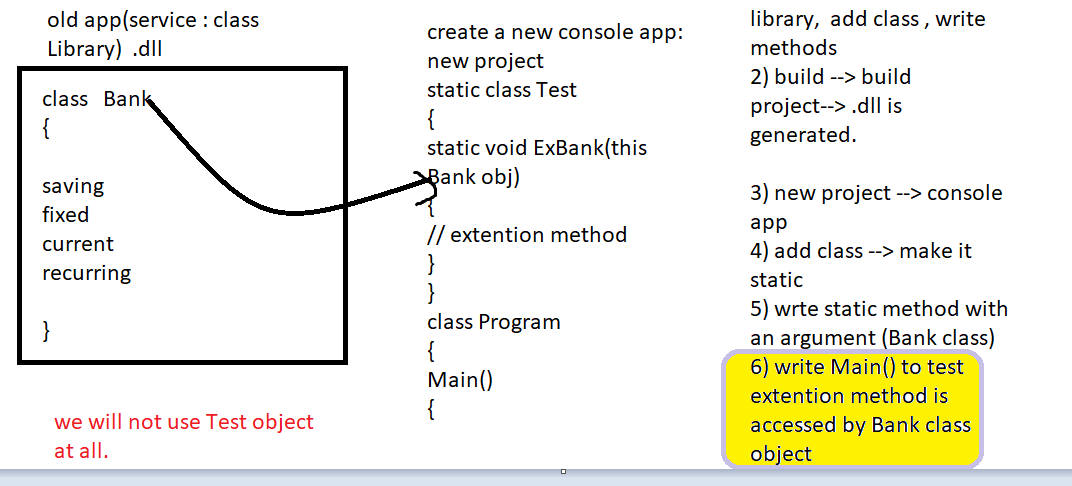
huge project divide into small modules by taking adv of partial classes.

========

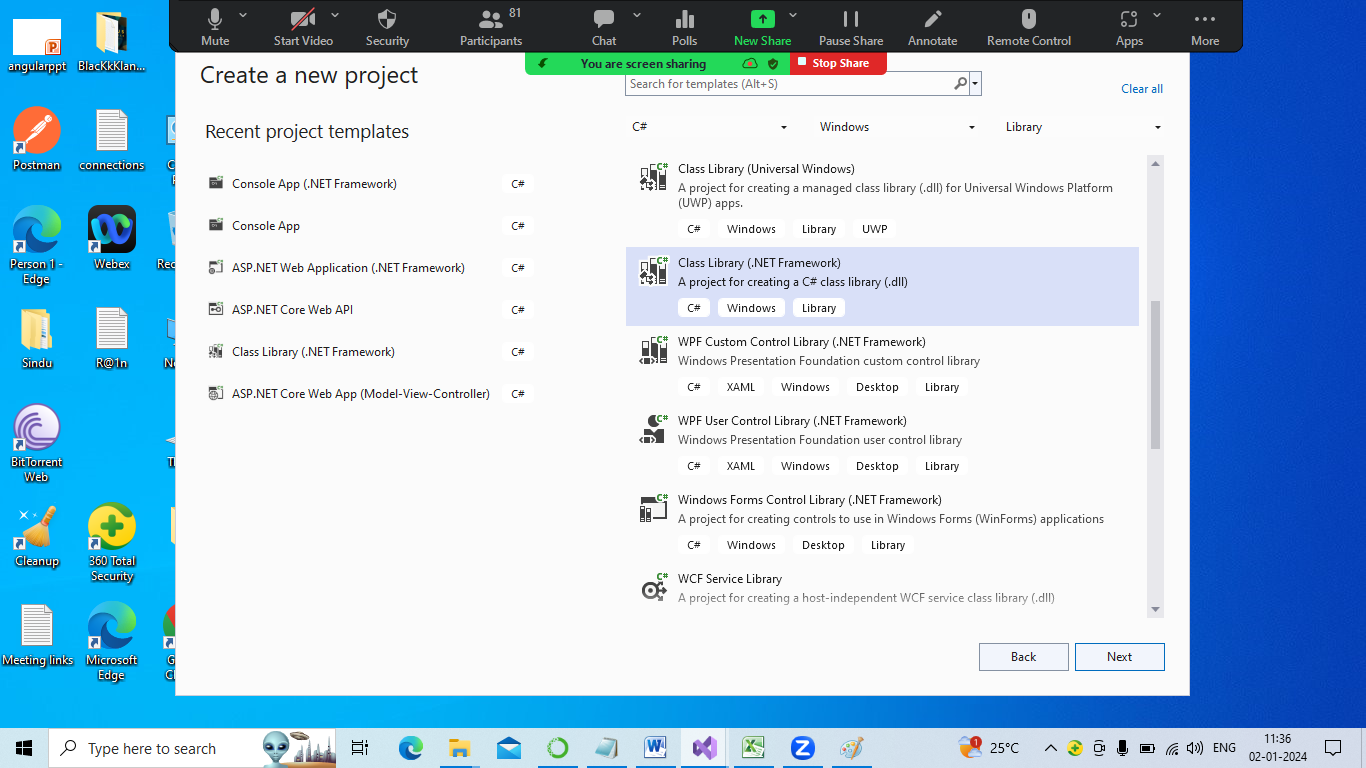
A C# extension method allows developers to extend the functionality of an existing type without creating a new derived type, recompiling, or otherwise modifying the original type. C# extension method is a special kind of static method that is called as if it was an instance method on the extended type. we will create a class library and extend its functionality from the caller code by implementing extension methods in C#.

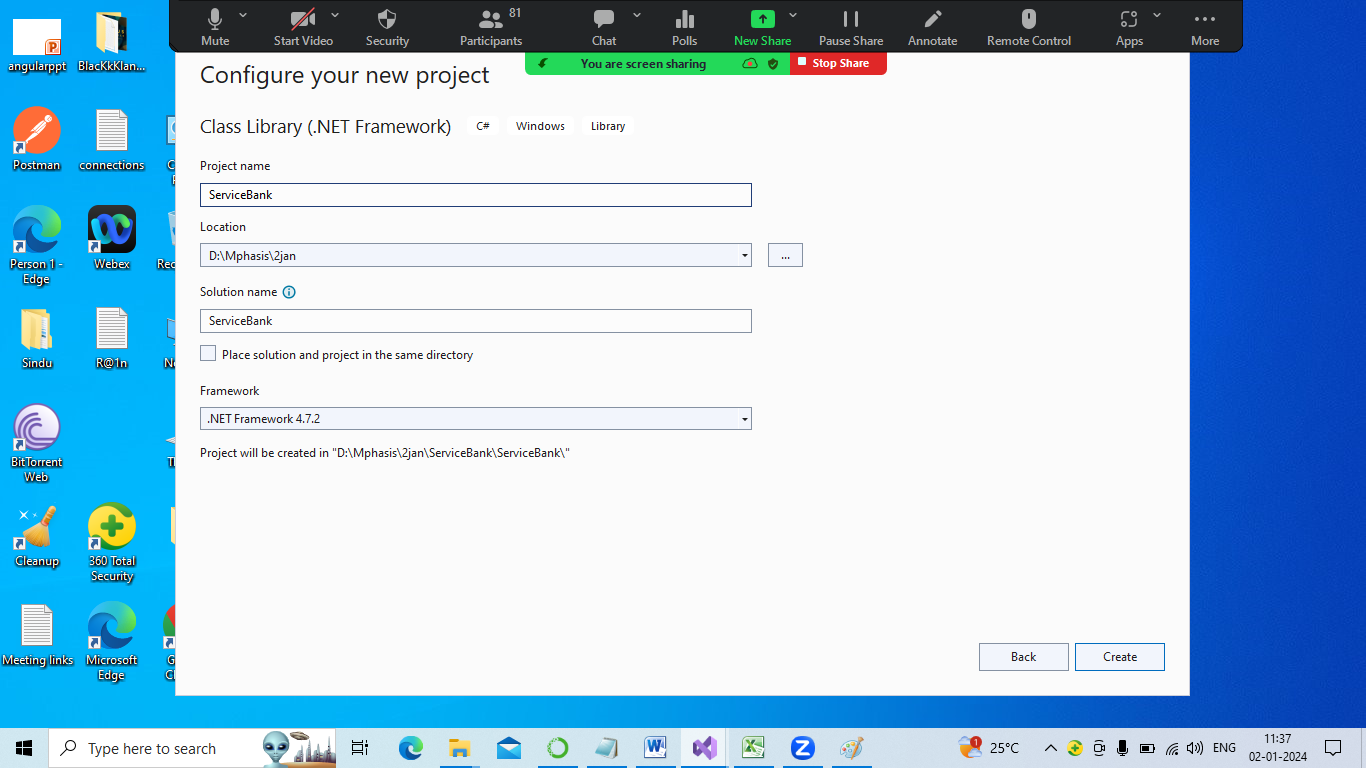
## C# Extension Method

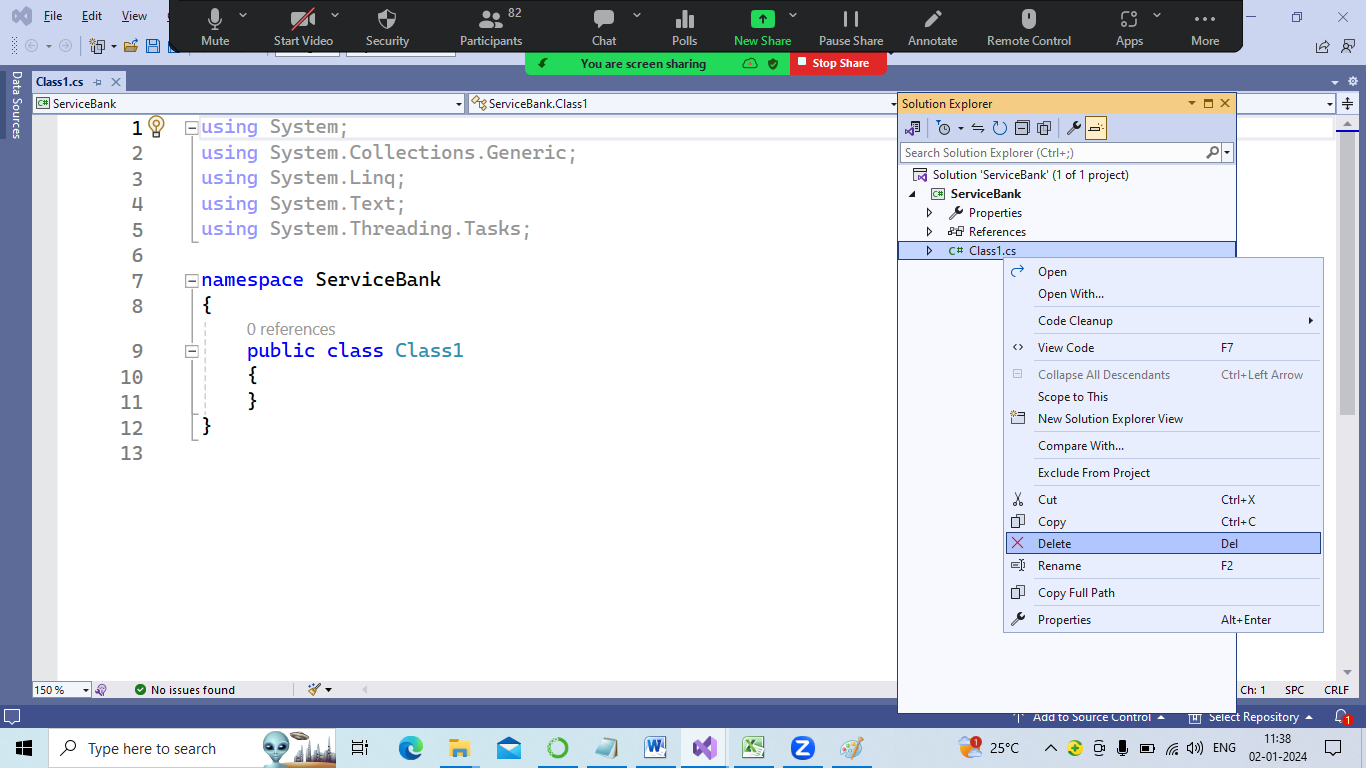
C# extension method is a static method of a static class, where the "this" modifier is applied to the first parameter. The type of the first parameter will be the type that is extended.

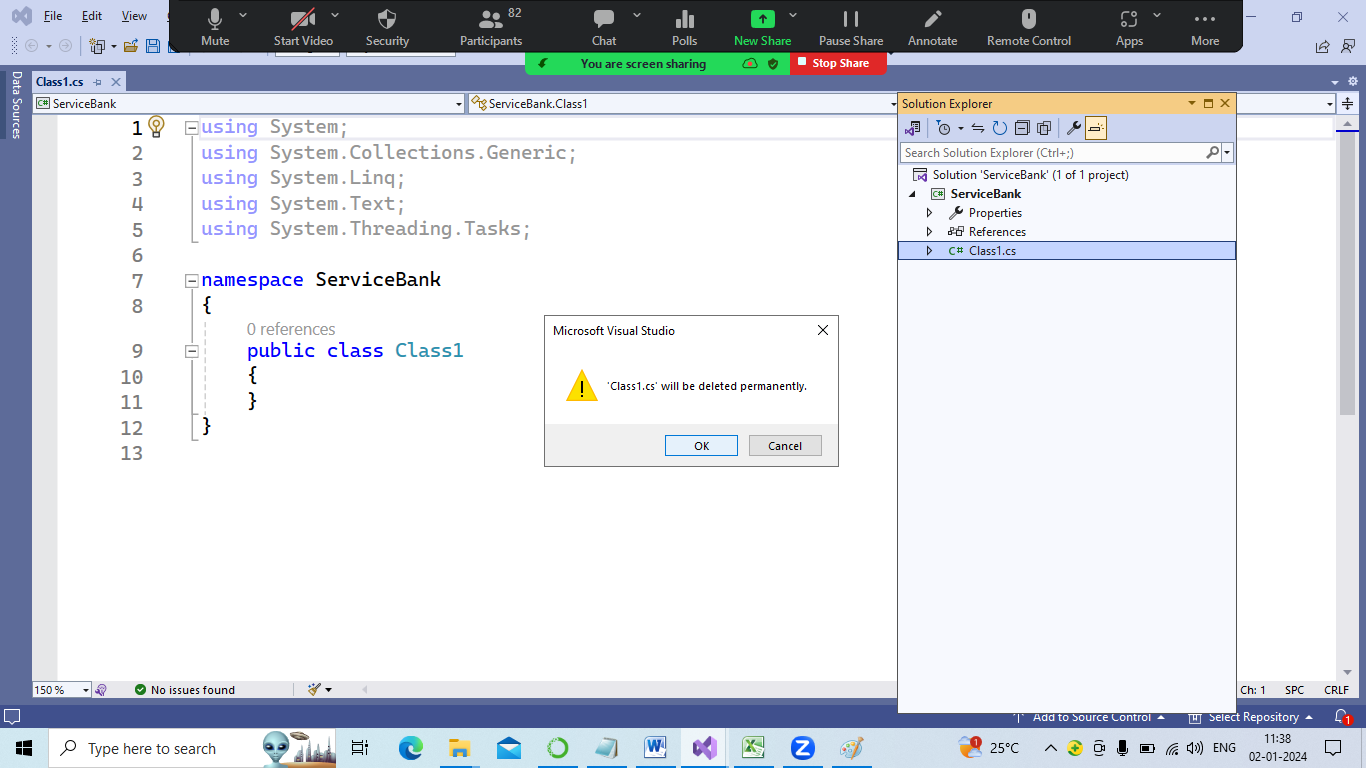


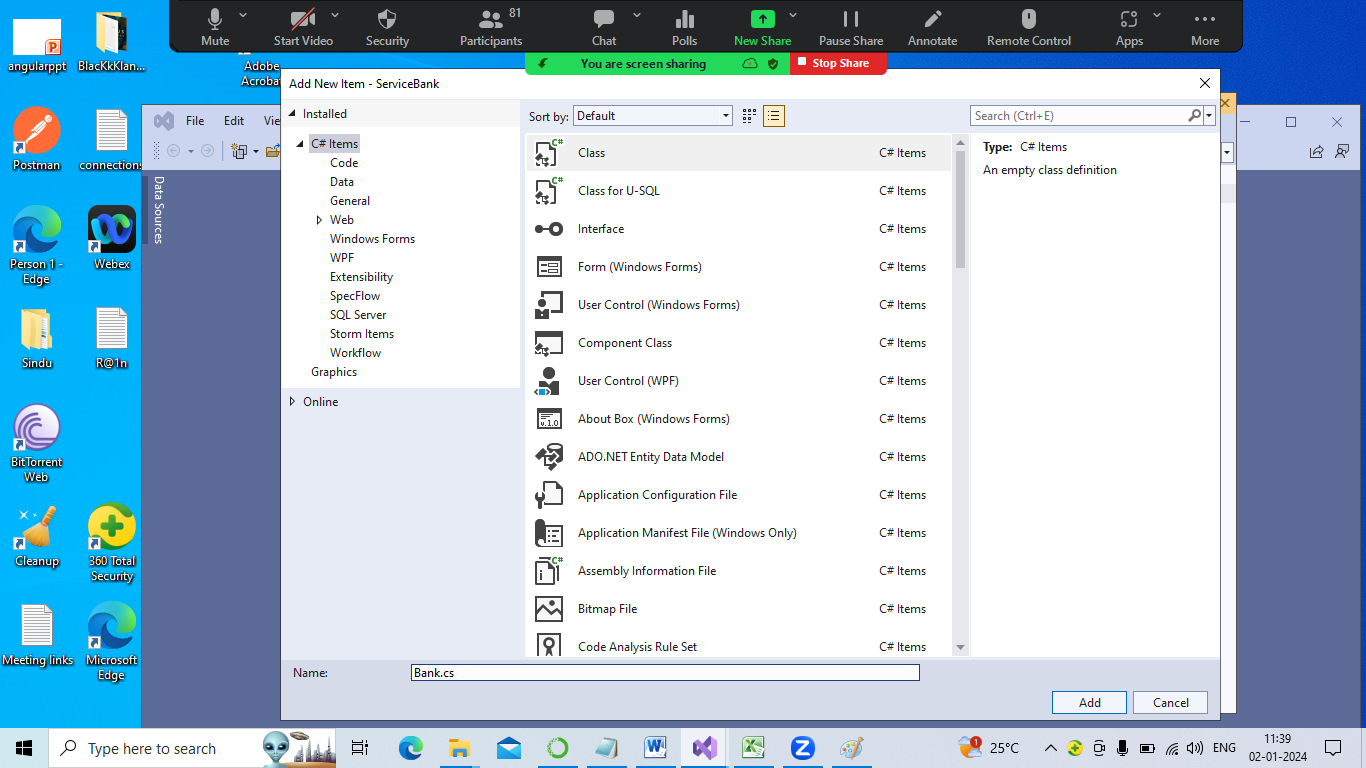
Extension methods are only in scope when you explicitly import the namespace into your source code with a using directive.

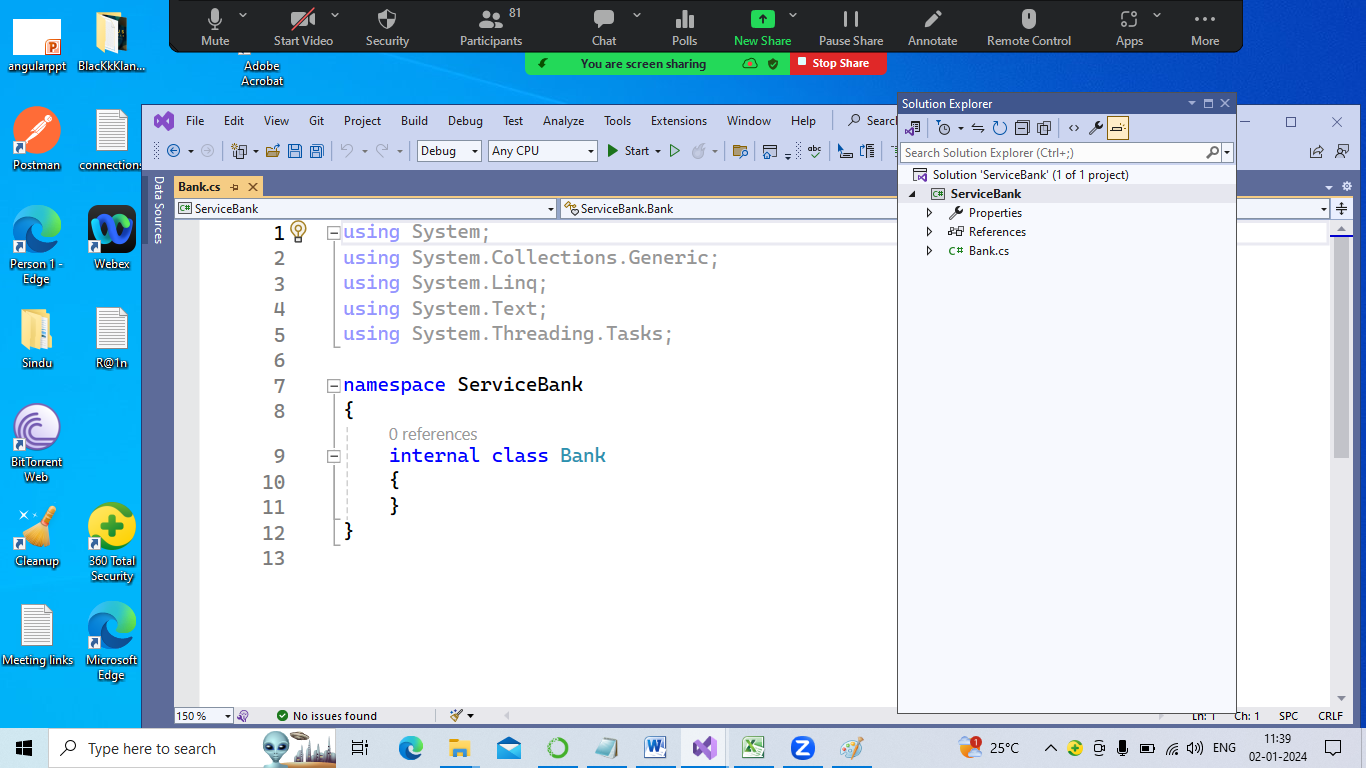


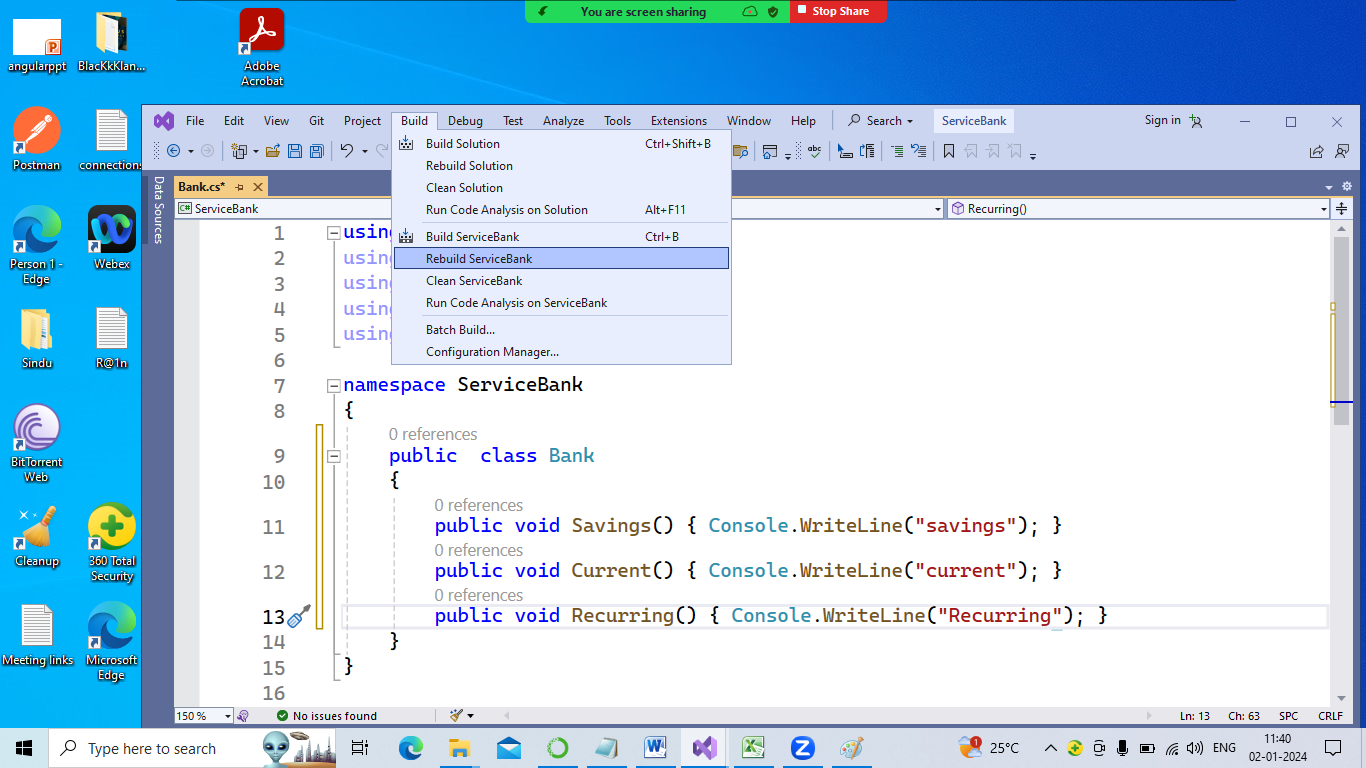


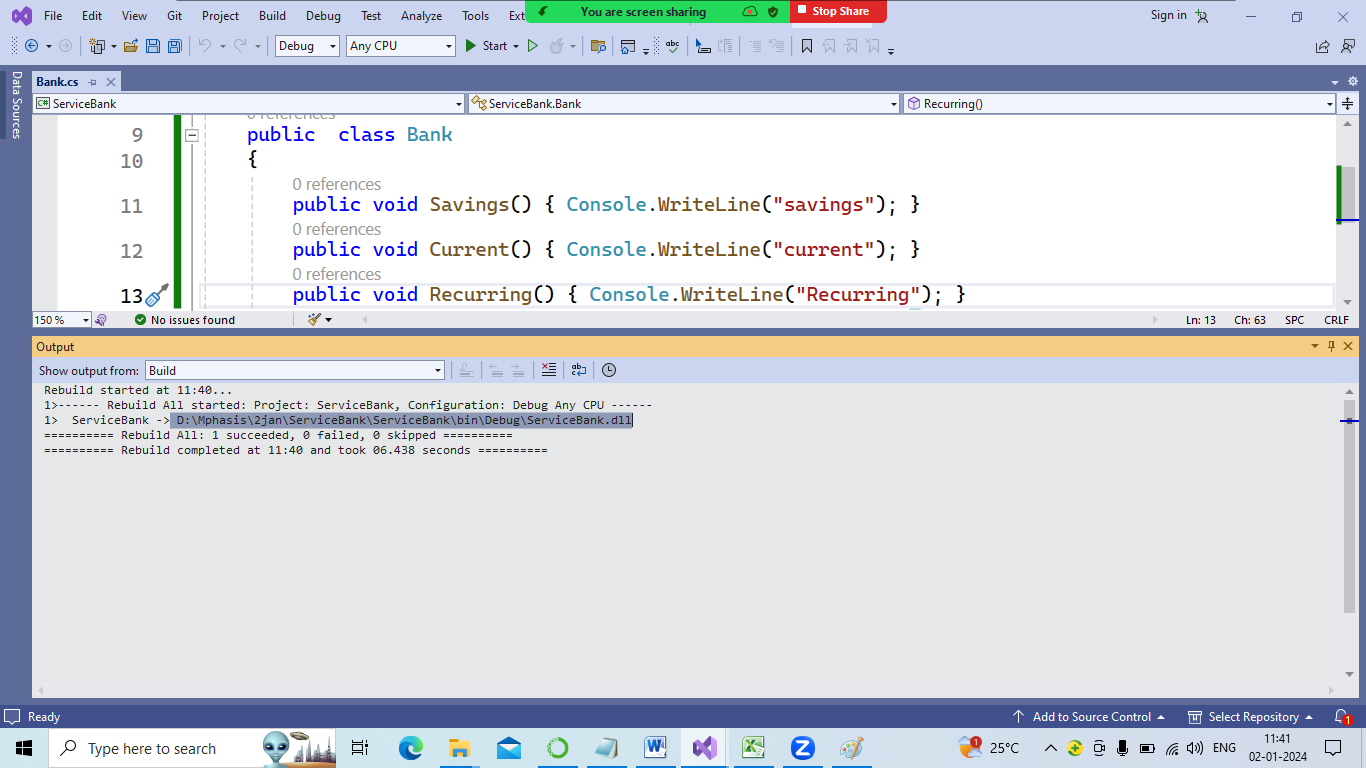










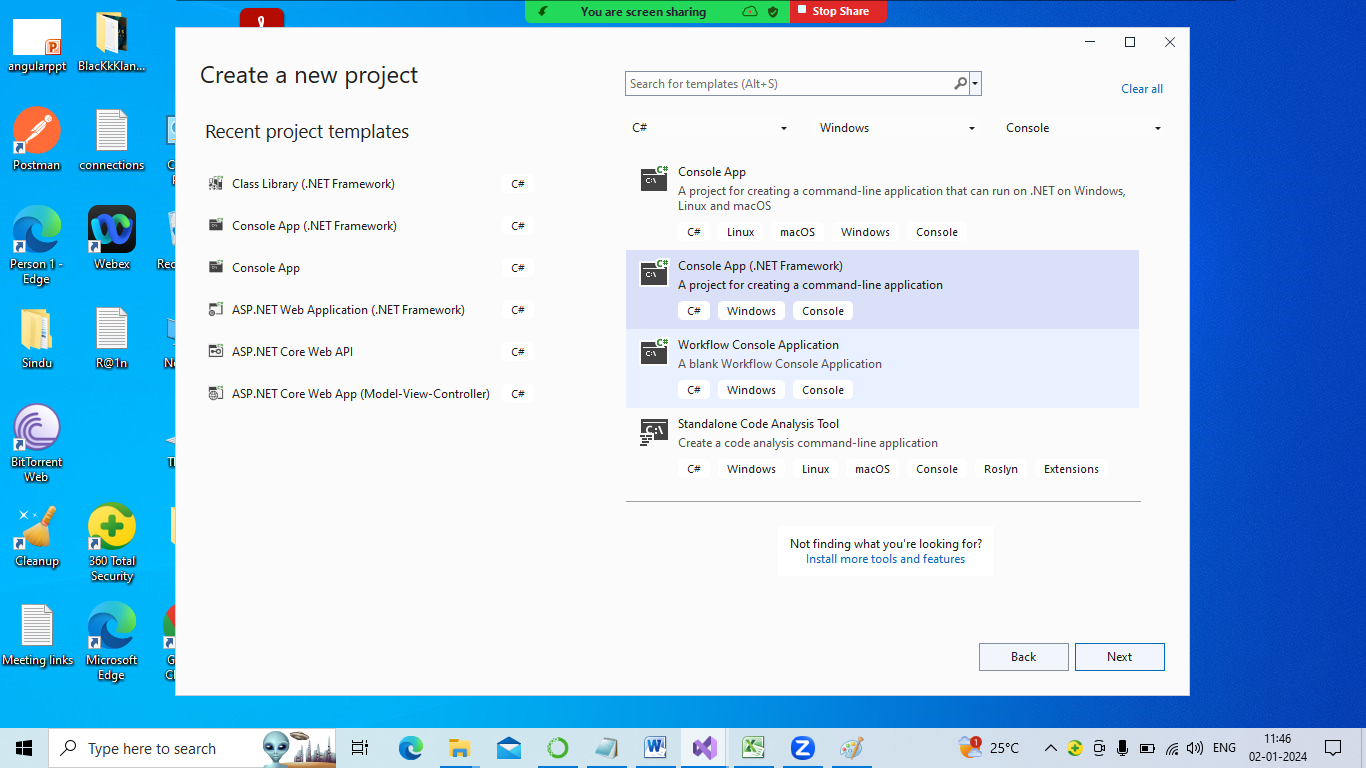
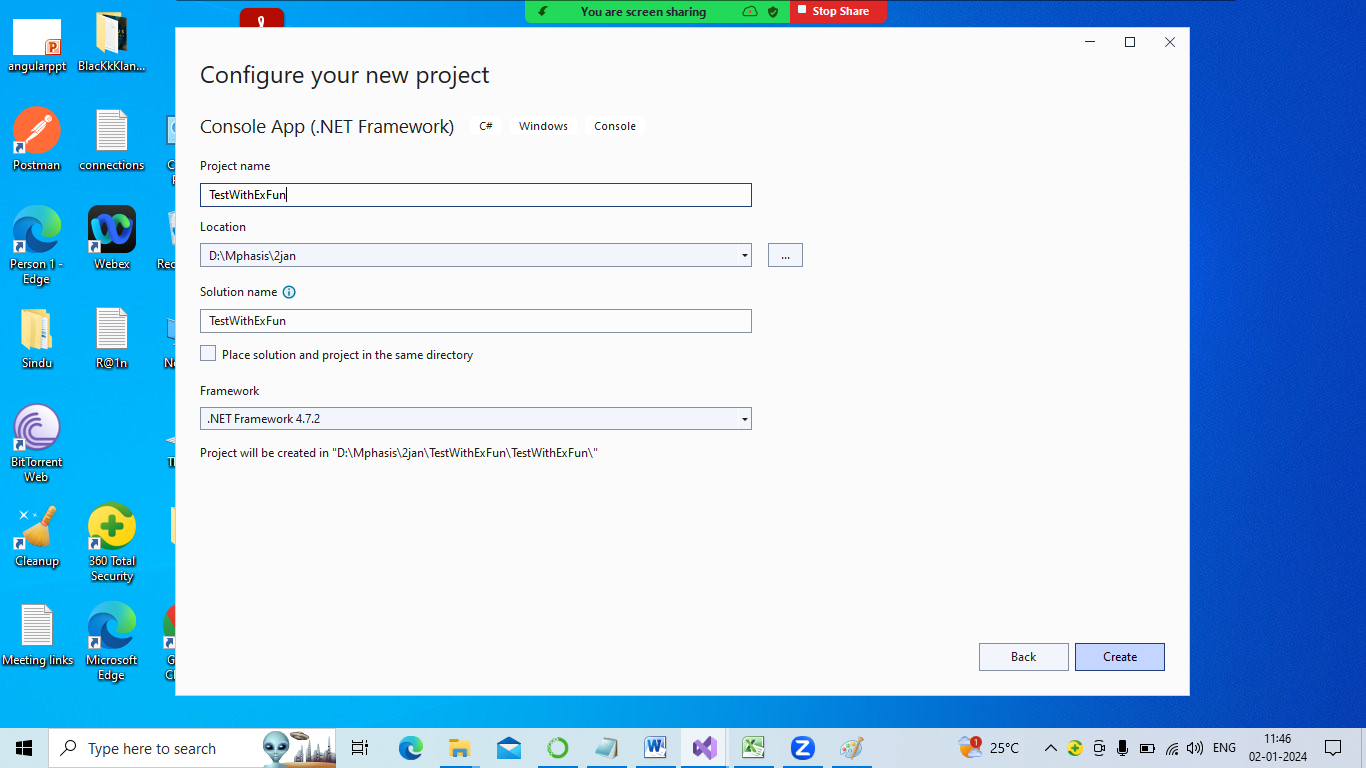


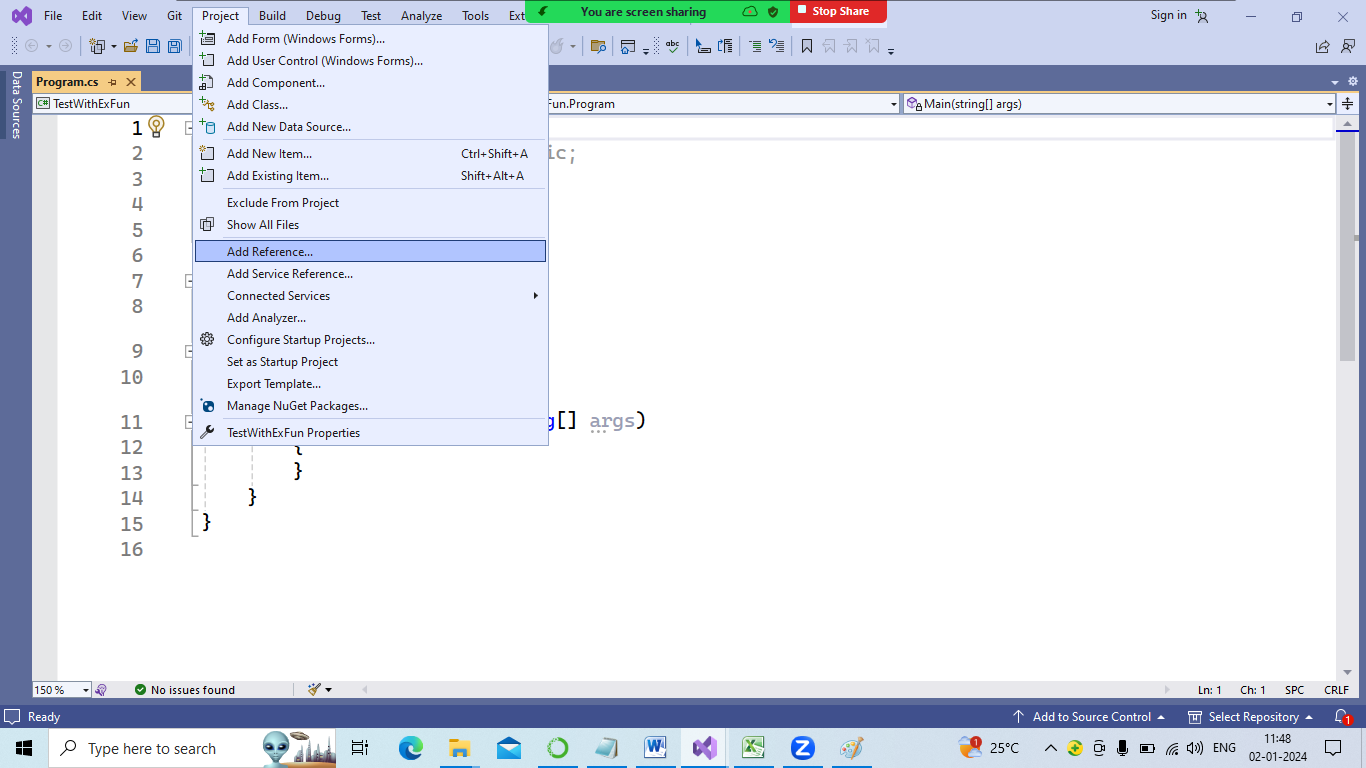
Path our compiled (build) app .dll file

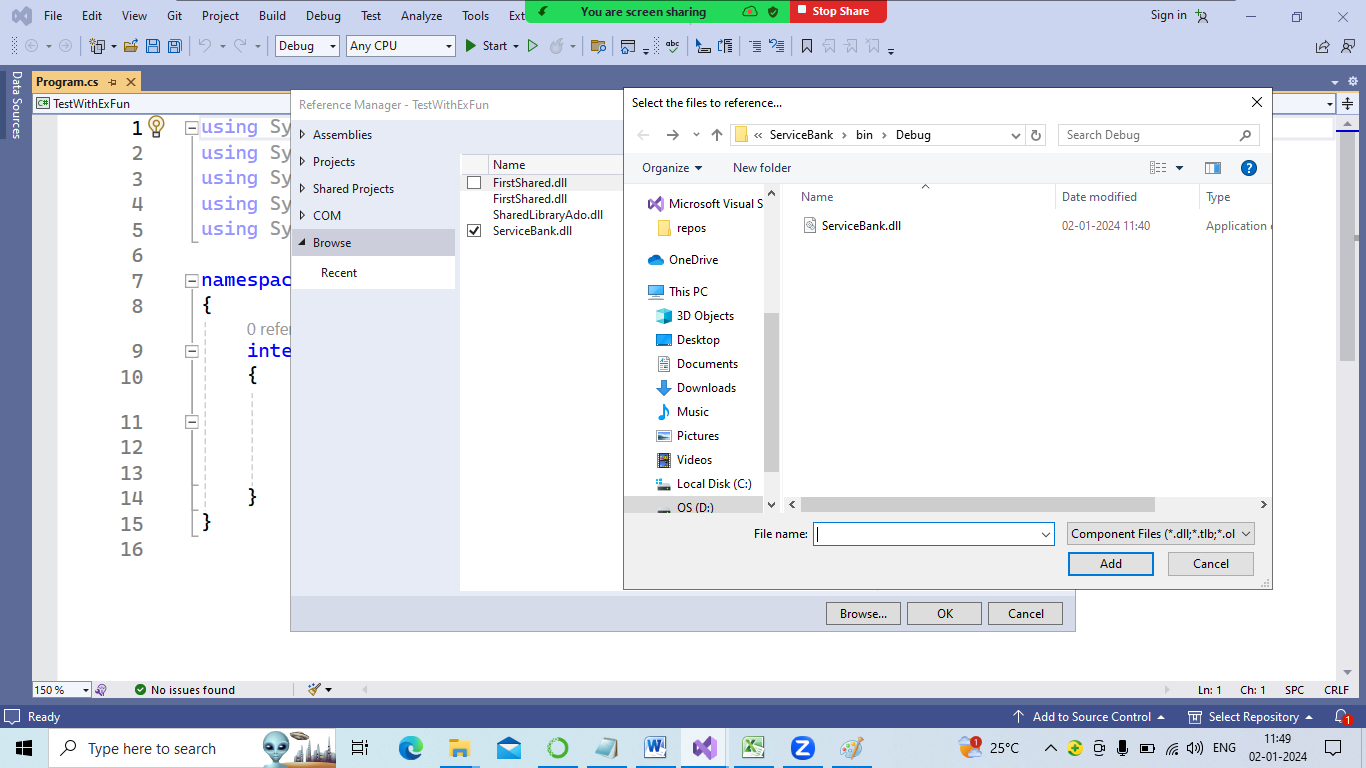
> D:\Mphasis\2jan\ServiceBank\ServiceBank\bin\Debug\ServiceBank.dll

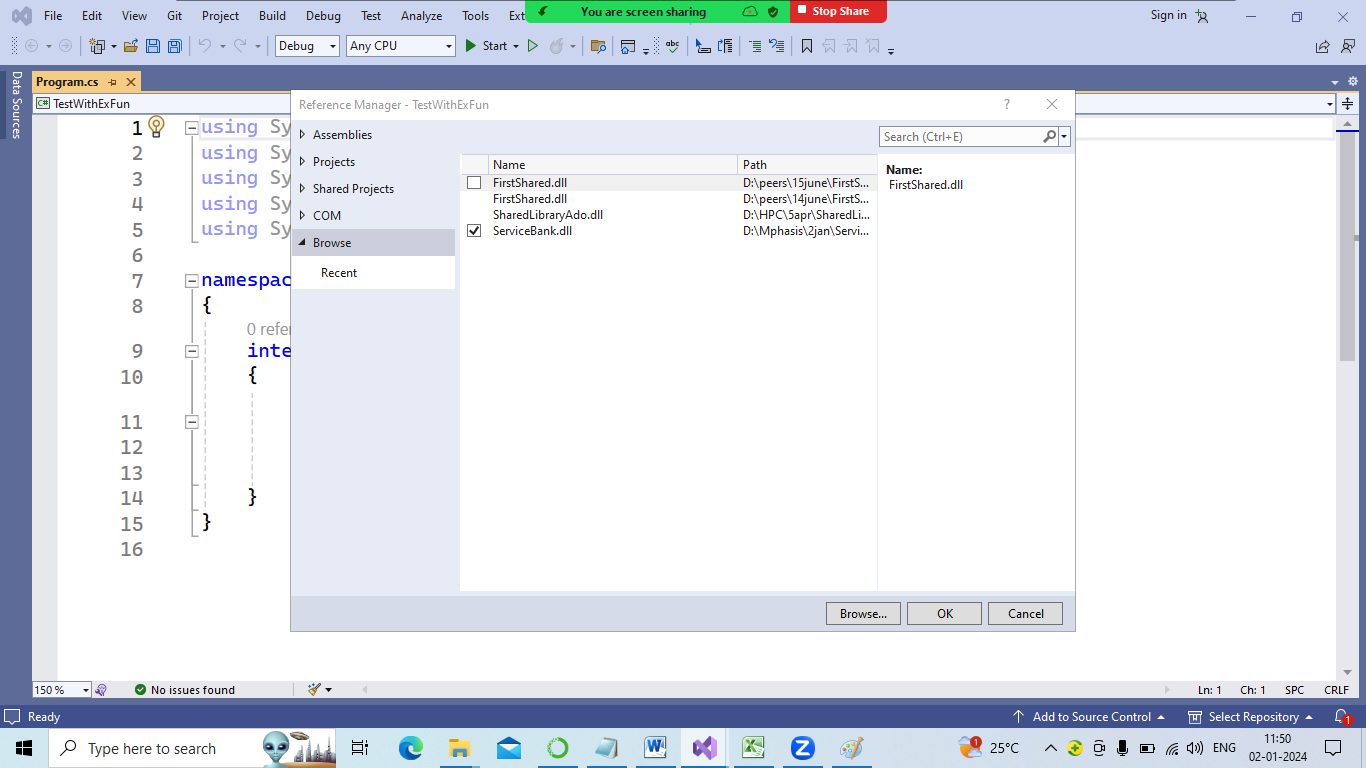
In new project we use dll file to add extention method , no harm to code or its running clients.

Because we will recompile ServiceBank Appl

* Console app
* 
* 
* Add 🡪 reference to serviceBank.dll to access its functions







using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using ServiceBank;

namespace TestWithExFun {

static class BankExt {

public static void ExtTapCard(this Bank b) {

Console.WriteLine("this is extention method"); } }

internal class Program {

static void Main(string[] args) {

Bank b = new Bank();

//existing part old app

b.Current();

b.Savings();

b.Recurring();

// extention of bank with new feature

b.ExtTapCard();

}

}

}

Already project is having client

We using whatsup .

Every month or 6 months we get updates to install , features

Is our whatsup working with or without updates right .

Basic P we are learning how to improve of existing project , to increase client base(business)

.dll: dynamic link library has a class , methods

Console app we add new feature to the client

Extention method: support new features to existing product.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibrary1{

public class College {

public void Tenth() { Console.WriteLine("10 th"); }

public void Plus2() { Console.WriteLine("12 the class "); }

public void Graduation() { Console.WriteLine("Graduate"); }

}

}

Build🡪 build solution

D:\Mphasis\2jan\ClassLibrary1\ClassLibrary1\bin\Debug\ClassLibrary1.dll

Only 3 methods are declared above

Close app(2022)

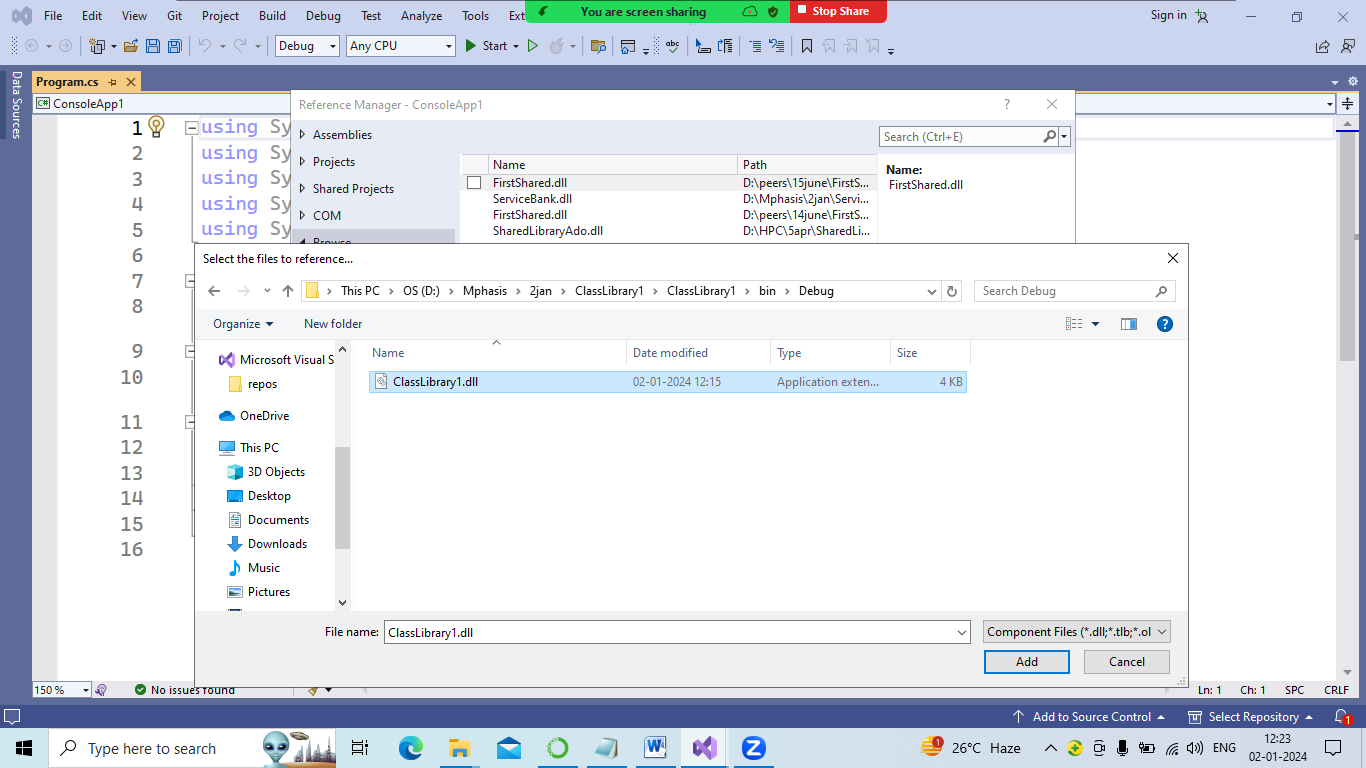
After taking new console app🡪 we need existing project , so we make new features . like append.

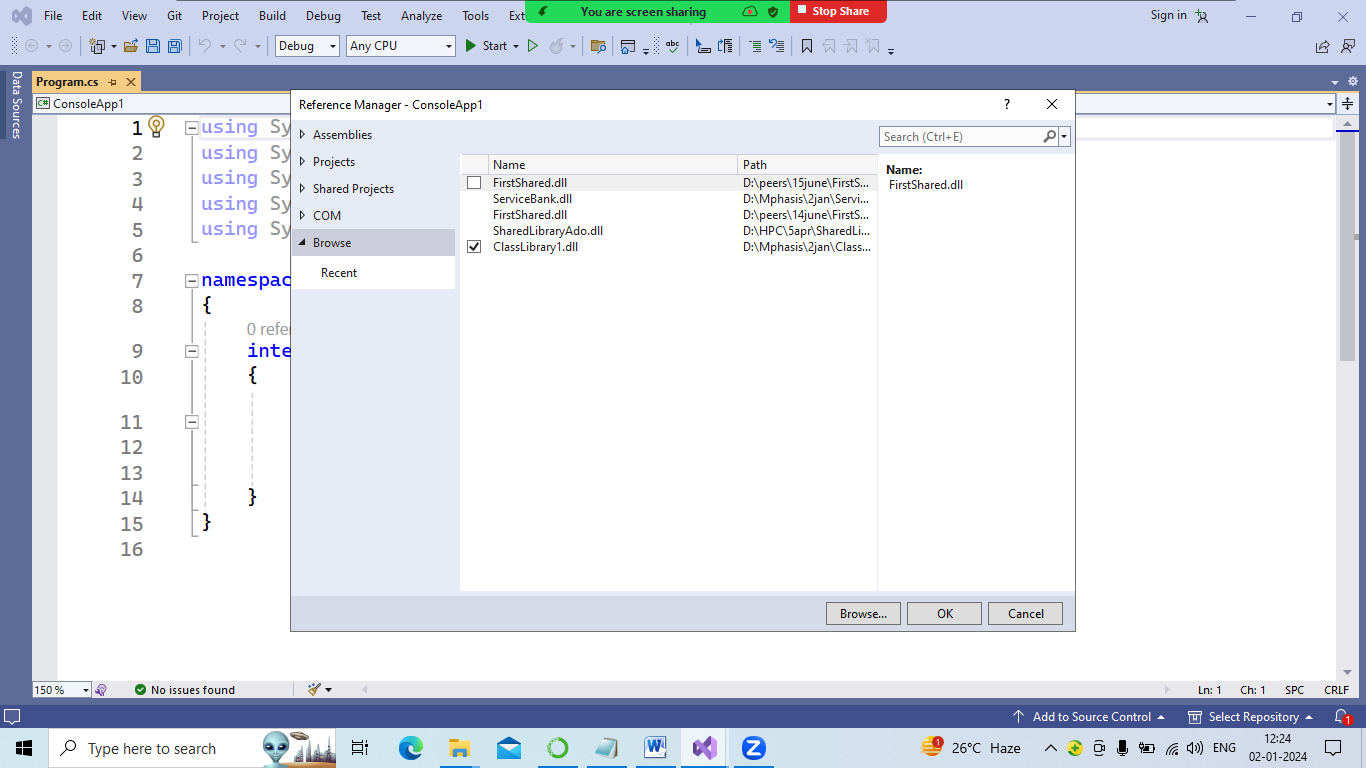
.

So current client able to access new features + old features .

3 + 1

4 features to new client who is paying more money to me.





IEnureable

IEnumerator

Why?

.net we have 2 interfaces for iteration.

It allows to access or traverse through all the index values(elements ) of a collection

A class that implement 2 interfaces can also be iterated(foreach).

We are allowed to iterate data items of collections such as

1. List
2. ArrayList
3. SortedList
4. HashTable
5. Dictionary
6. Queque
7. Stack

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Desc | Method | Property |
| IEnumerable | Used to iterate a given object | GetEnumerator()  Allows read-only access to the collection .  And its return type is “IEnumerator” | -- |
| IEnumerator | Is used to access the current element from a collection.by using its property “Current” | MoveNext()  Will return “false” if enumerator has reached the end of the collection. Or it returns “true” | Current |
|  |  | Reset()  To set enumerator to first element of collection |  |