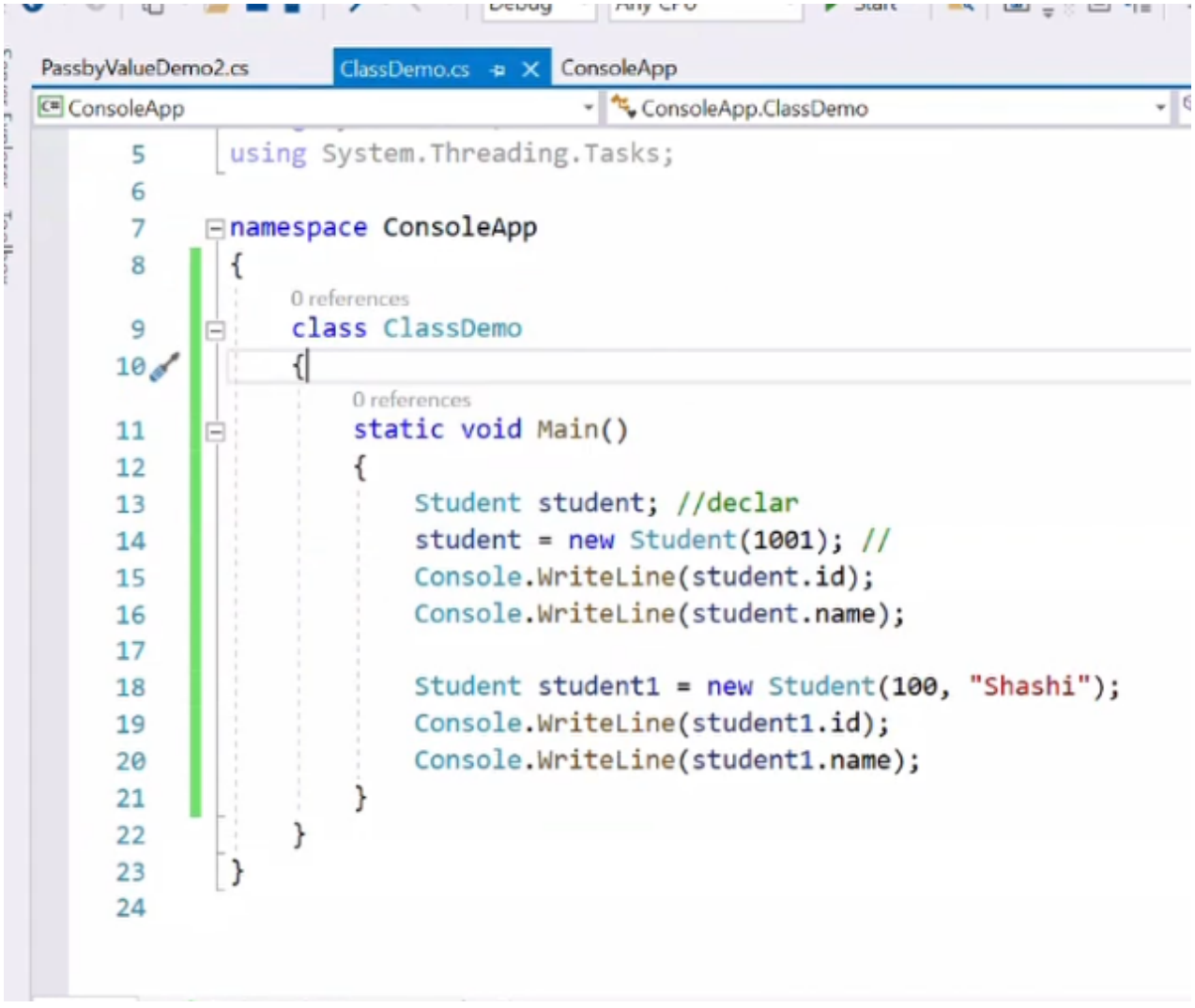
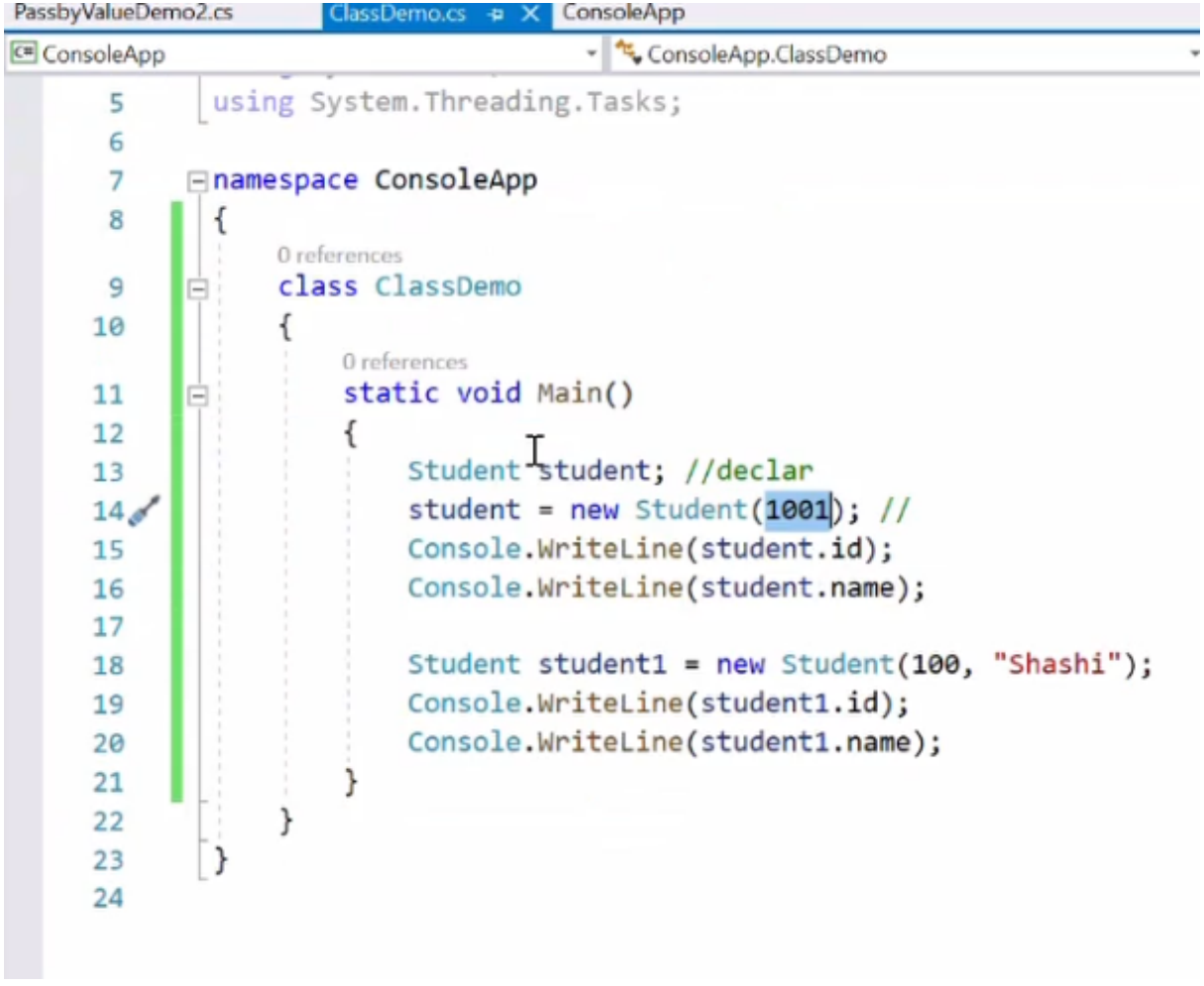
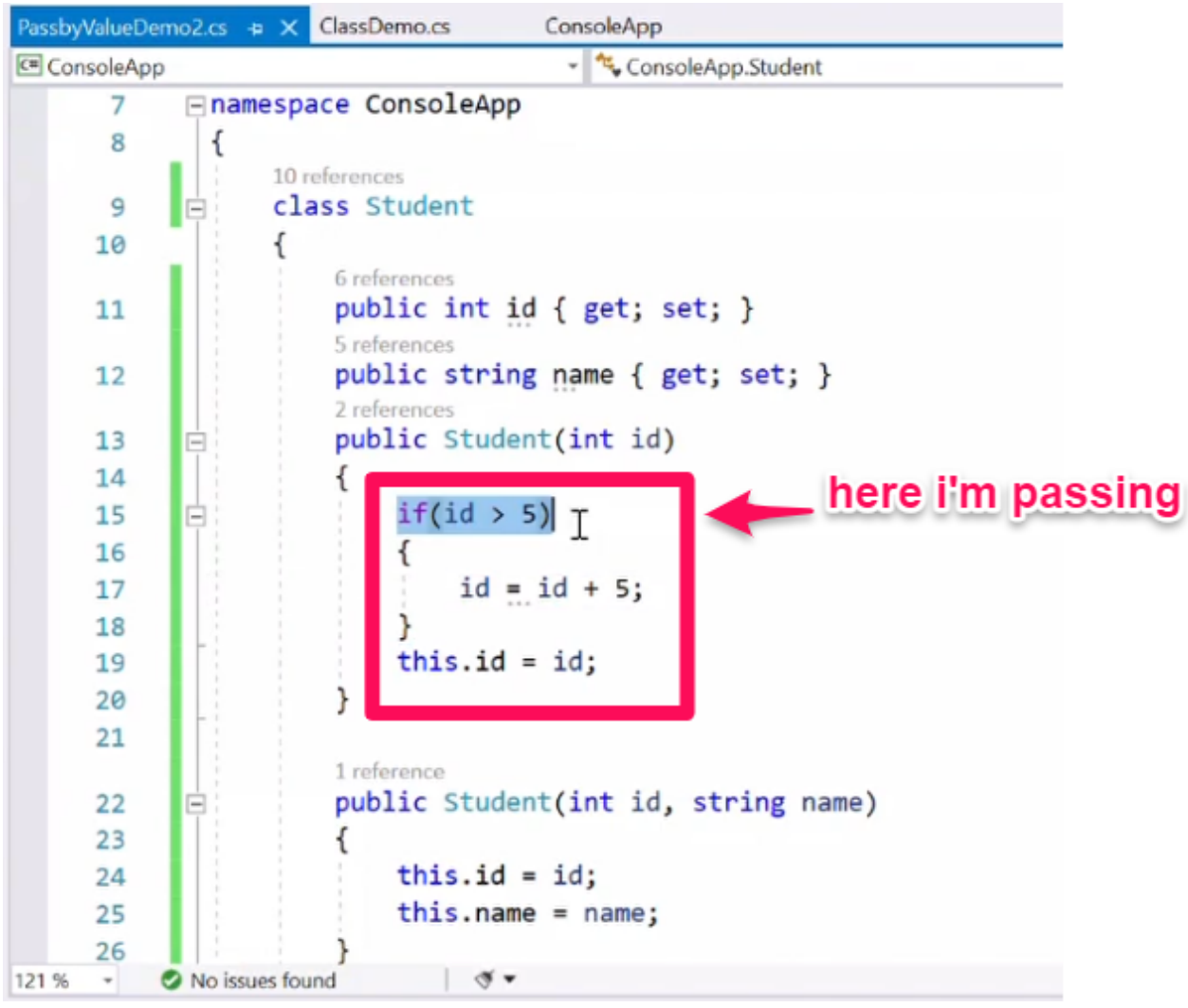
C#-7


PassbyValueDemo2.cs
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.ClassDemo
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class ClassDemo
{
0 references
static void Main()
{
Student student; //declar
student = new Student(1001); //
Console.WriteLine(student.id);
Console.WriteLine(student.name);
Student student1 = new Student(100, "Shashi");
Console.WriteLine(student1.id);
Console.WriteLine(student1.name);
}
}
}


PassbyValueDemo2.cs
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.ClassDemo
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class ClassDemo
{
0 references
static void Main()
{
Student student; //declar
student = new Student(1001); //
Console.WriteLine(student.id);
Console.WriteLine(student.name);
Student student1 = new Student(100, "Shashi");
Console.WriteLine(student1.id);
Console.WriteLine(student1.name);
}
}
}



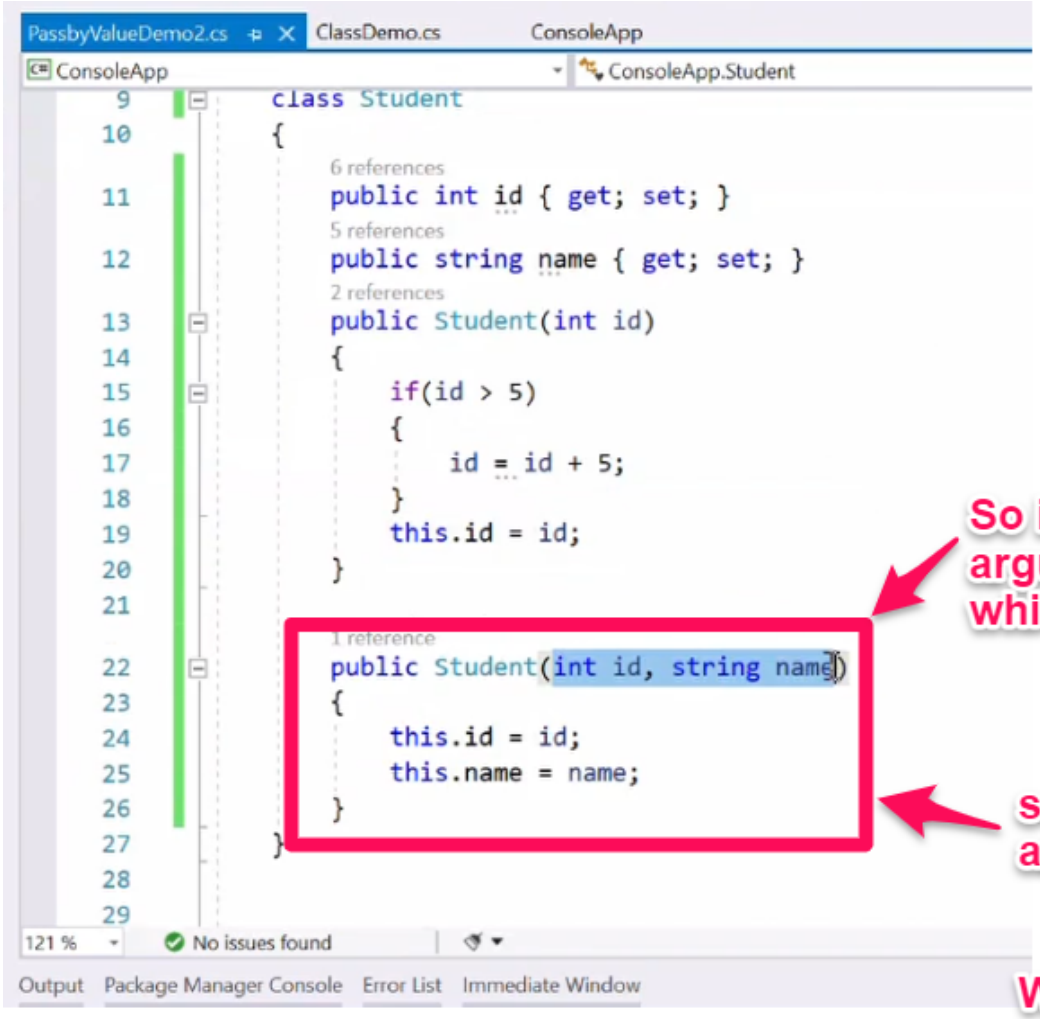
Now what will be the output ?

so i'm passing 1001
so as it is greater than 5, so it will add 5 to it and 1006 gets stored in the id

```
namespace ConsoleApp
{
    0 references
    class ClassDemo
    {
        0 references
        static void Main()
        {
            Student student; //declar
            student = new Student(1001); //
            Console.WriteLine(student.id);
            Console.WriteLine(student.name);

            Student student1 = new Student(100, "Shashi");
            Console.WriteLine(student1.id);
            Console.WriteLine(student1.name);
        }
    }
}
```

But when i come down here
i'm passing 2 parameters here



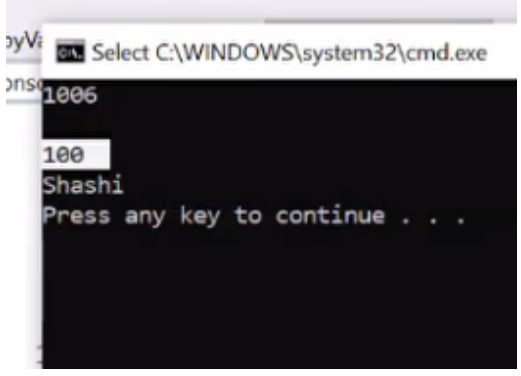
So it will come down to 2 argument constructor which is taking id and name

so here i'm directly taking id and name and storing it

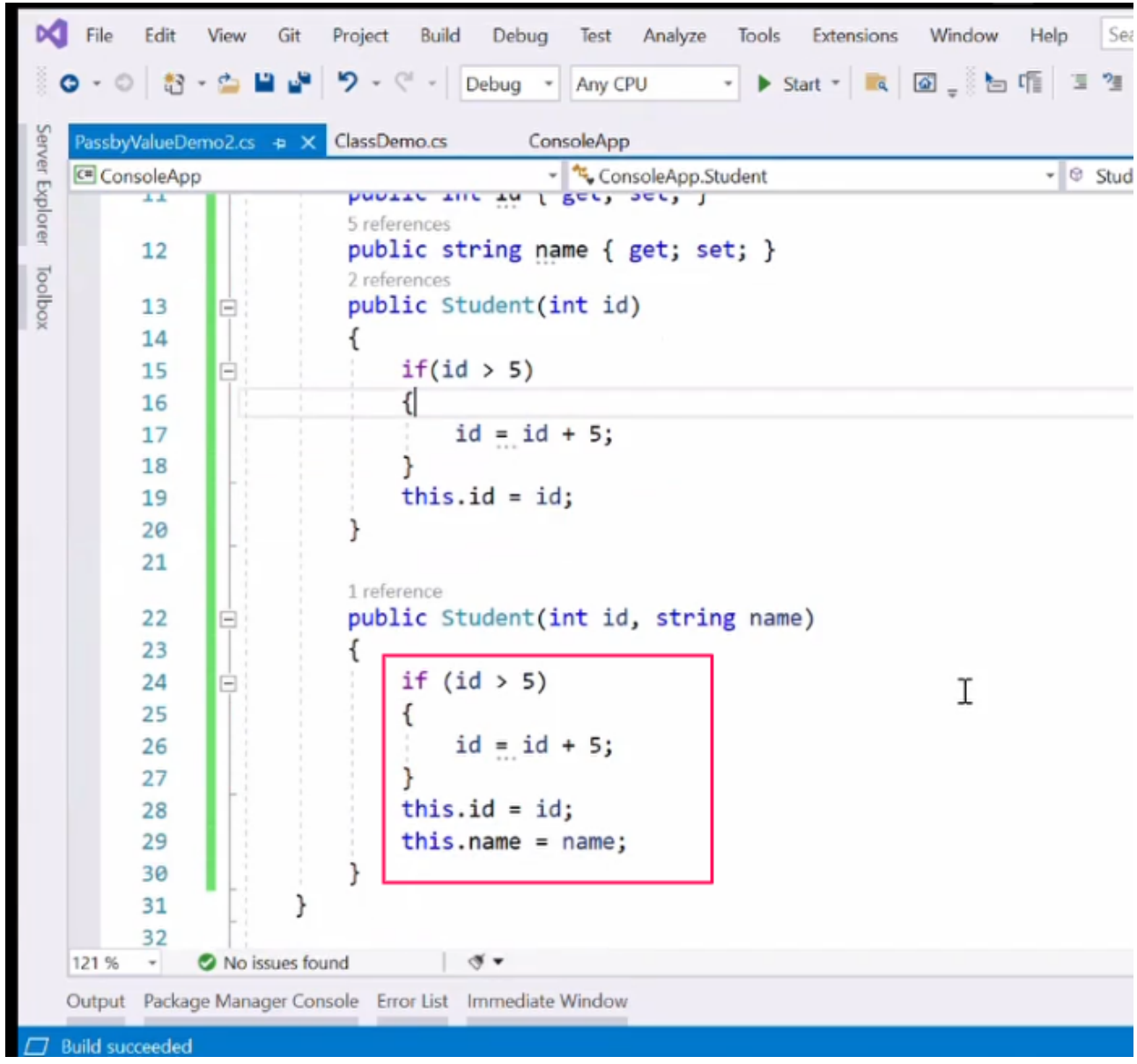
Where is the bussiness Rule ?

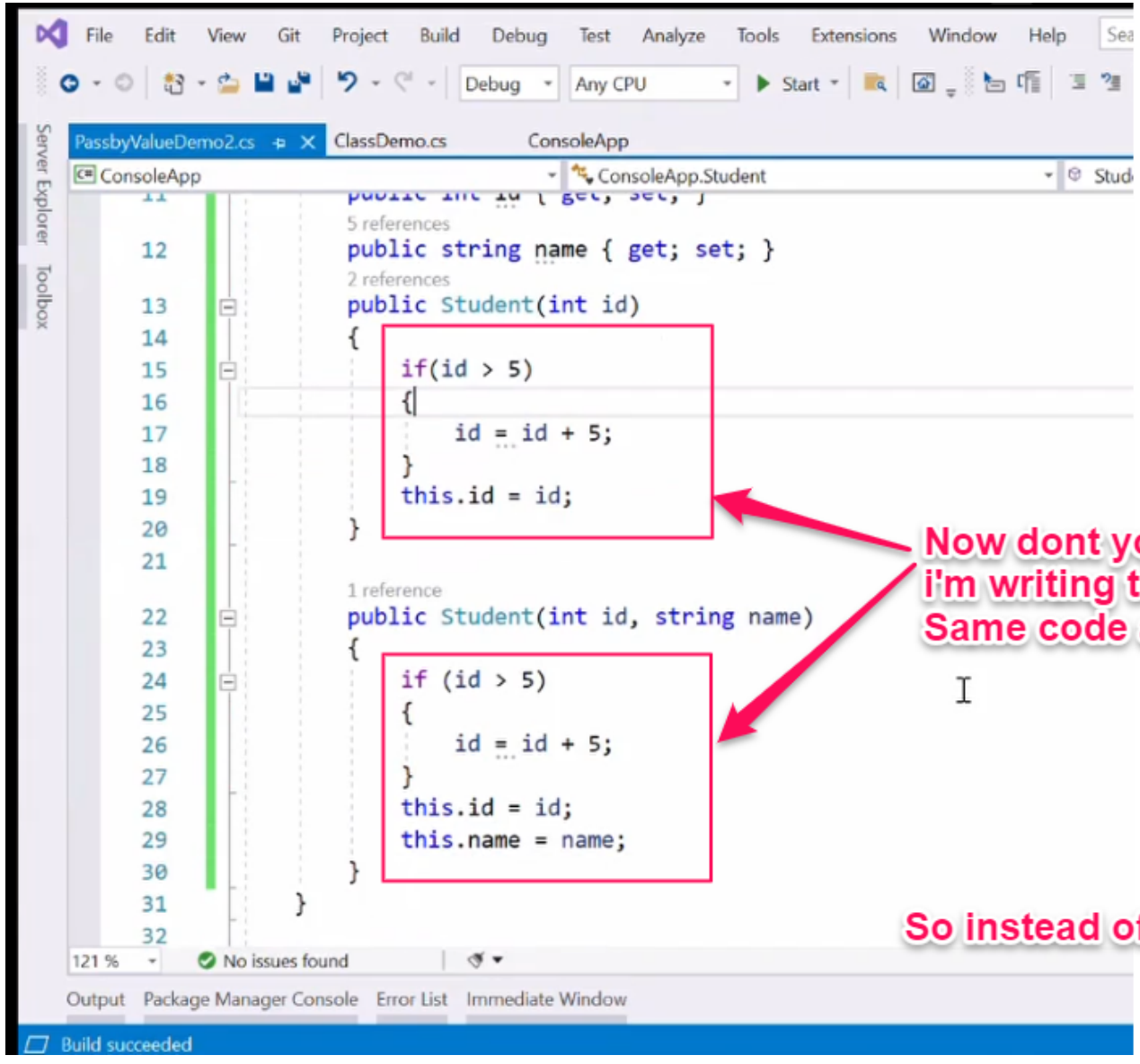
there is no bussiness rule

So, i was expecting it to be as 105 and shashi because of the fact that the above one has a bussiness rule but when i looked it in the output



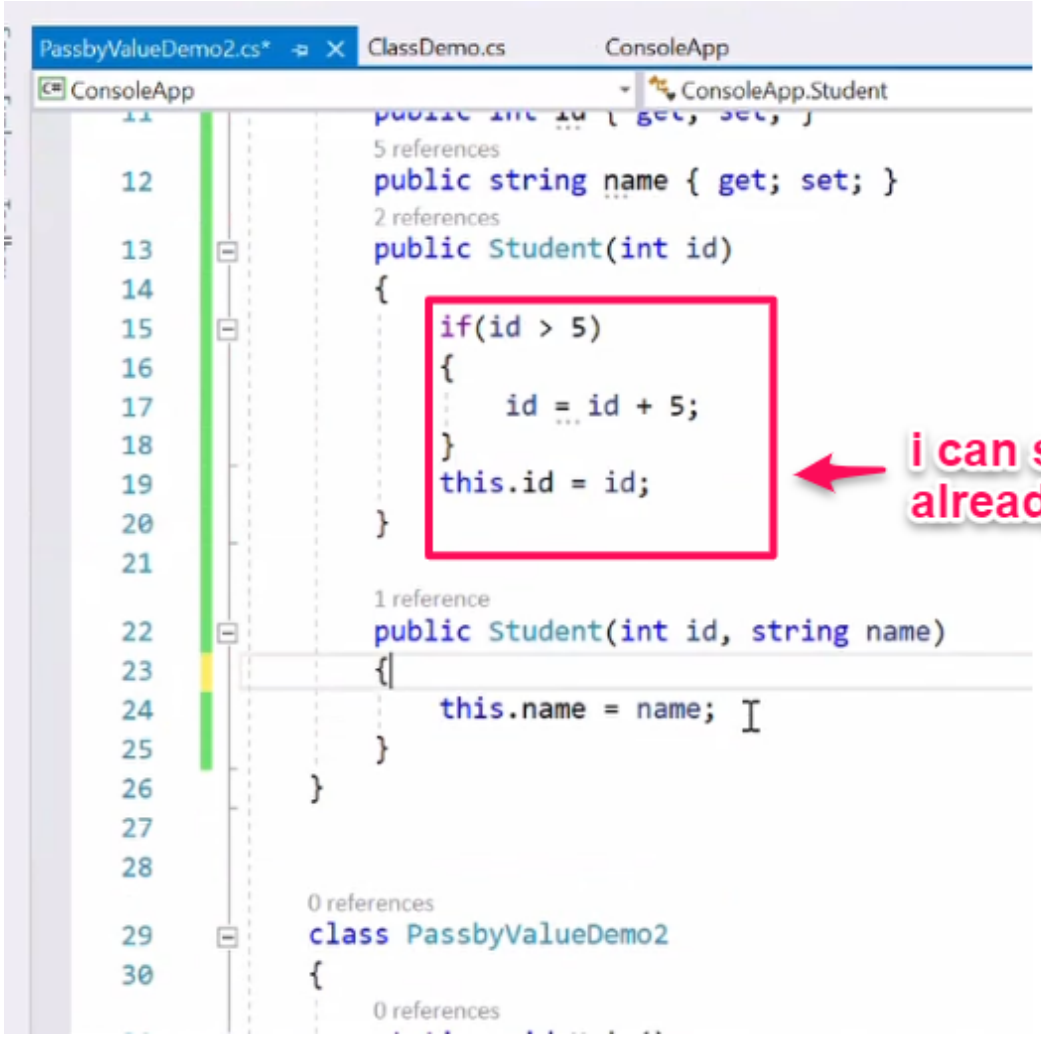
you can see it is still 100, because i have implemented the rule in one of the constructor , so i need to copy paste the same logic in 2nd constructor also, so that like same bussiness rule can be implemented


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Debug
Any CPU
Start
PassbyValueDemo2.cs
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.Student
5 references
public string name { get; set; }
2 references
public Student(int id)
{
if(id > 5)
{
id = id + 5;
}
this.id = id;
}
1 reference
public Student(int id, string name)
{
if (id > 5)
{
id = id + 5;
}
this.id = id;
this.name = name;
}
}
121 %
No issues found
Output
Package Manager Console
Error List
Immediate Window
Build succeeded



Now dont you think
i'm writing too much of code here
Same code again and again

So instead of doing this what i can do is

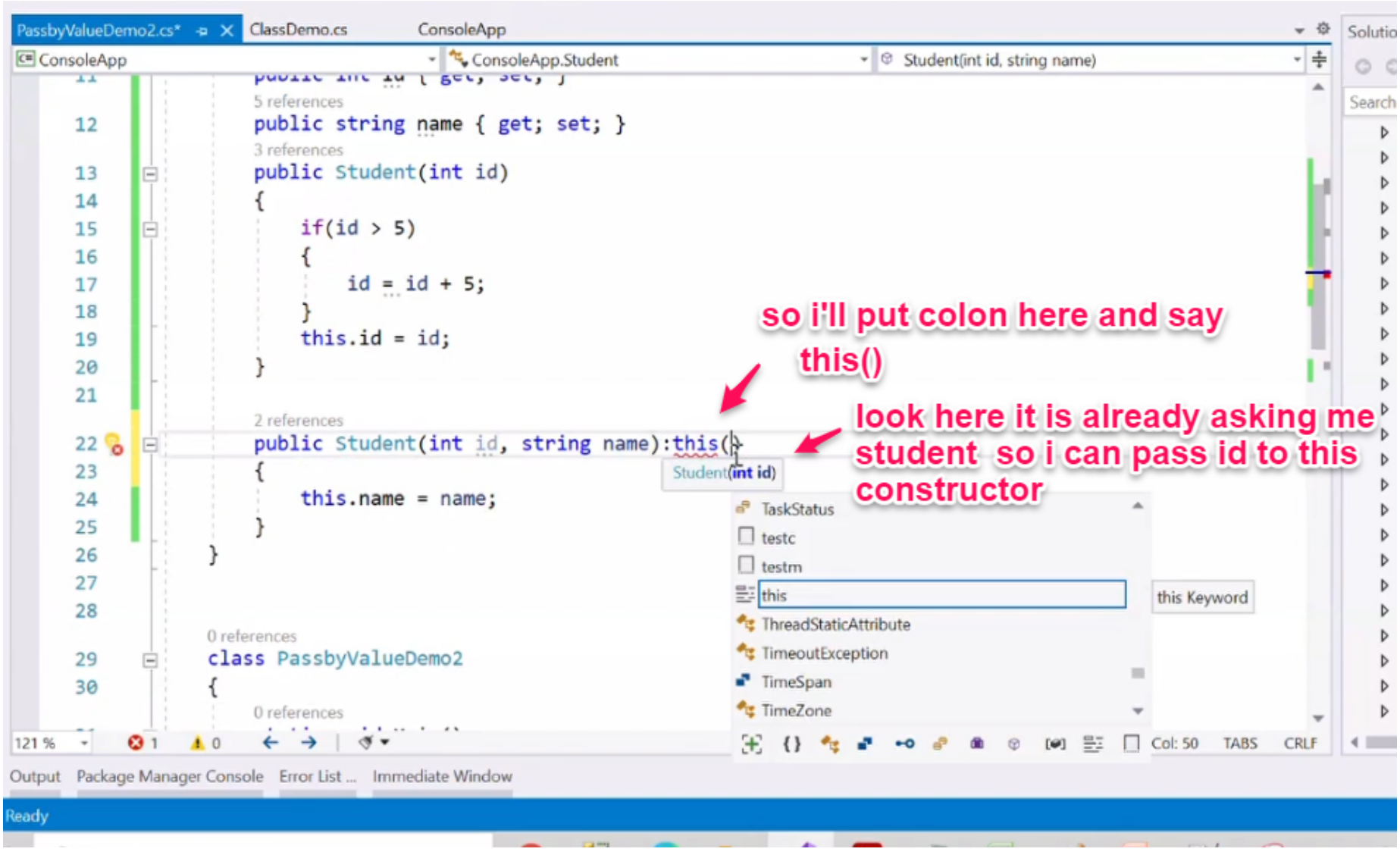


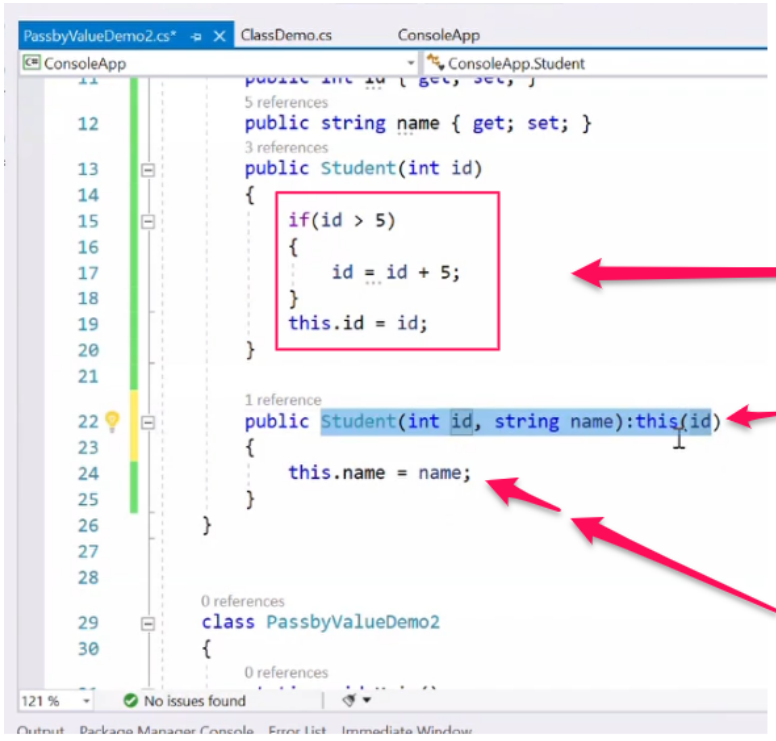
i can say that id initialization already this guy is doing

So i can call this Constructor

Now How will you call the object what represent a object here -----

this always represent the current object

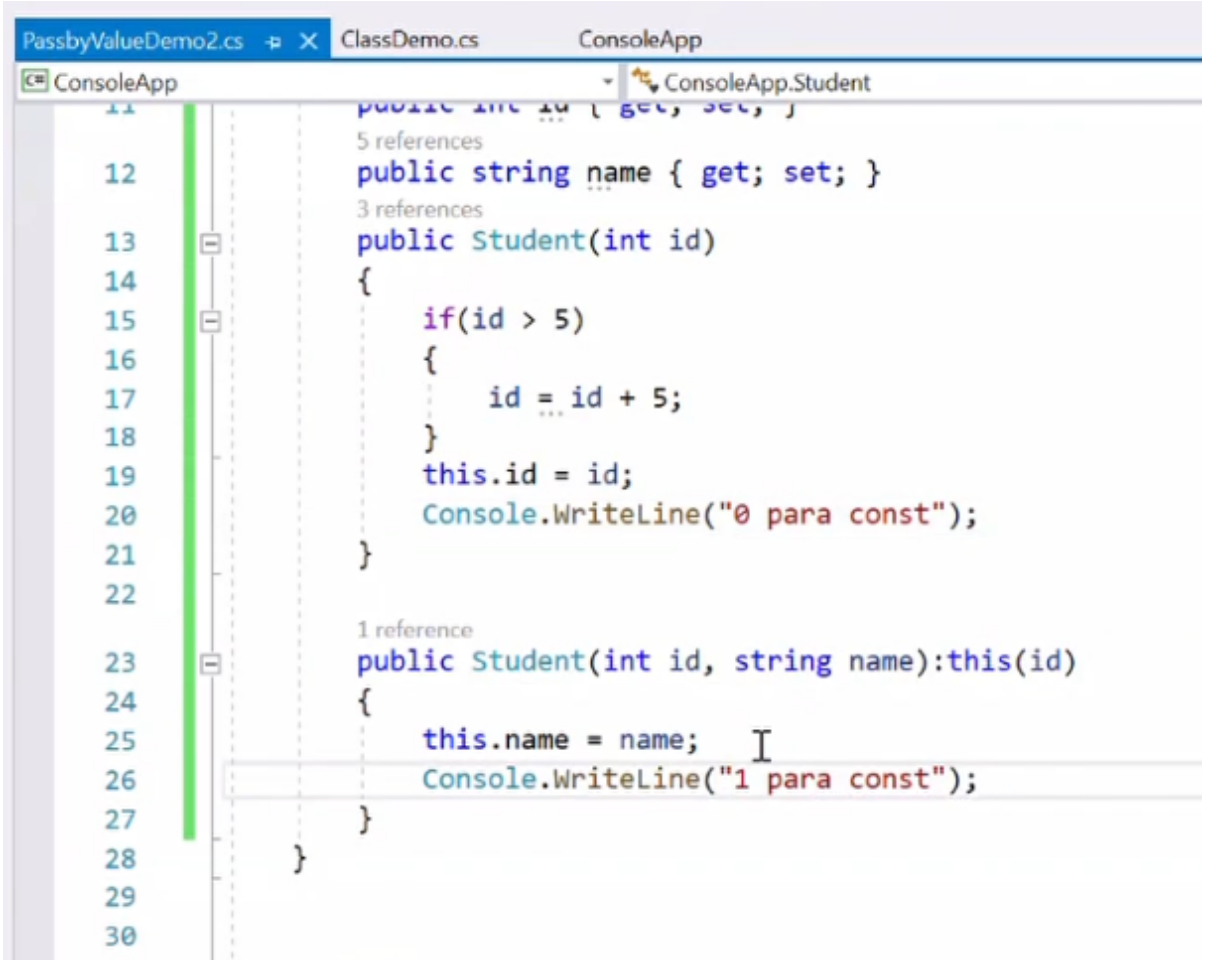

PassbyValueDemo2.cs*
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.Student
Student(int id, string name)
5 references
public string name { get; set; }
3 references
public Student(int id)
{
if(id > 5)
{
id = id + 5;
}
this.id = id;
}
2 references
public Student(int id, string name):this()
{
this.name = name;
}
}
0 references
class PassbyValueDemo2
{
0 references
Student(int id)
so i'll put colon here and say this()
look here it is already asking me student so i can pass id to this constructor
TaskStatus
testc
testm
this
this Keyword
ThreadStaticAttribute
TimeoutException
TimeSpan
TimeZone
121 %
Col: 50
TABS
CRLF
Output
Package Manager Console
Error List ...
Immediate Window
Ready

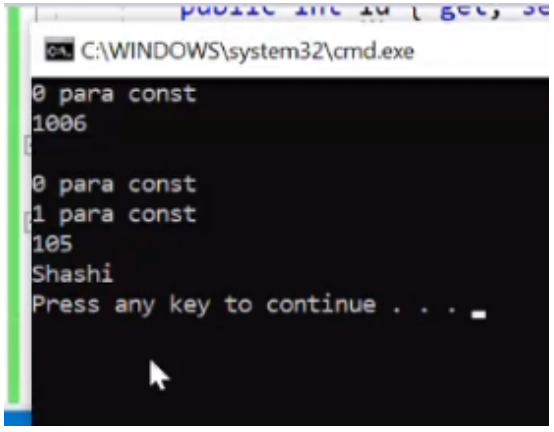


Now whenever i'm calling the second constructor
which is taking 2 parameter by default it will call the constructor
which is taking 1 parameter

here
you can see
this is executing this particular thing

after the execution of this
this particular constructor get called


PassbyValueDemo2.cs
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.Student
5 references
public string name { get; set; }
3 references
public Student(int id)
{
if(id > 5)
{
id = id + 5;
}
this.id = id;
Console.WriteLine("0 para const");
}
1 reference
public Student(int id, string name):this(id)
{
this.name = name;
Console.WriteLine("1 para const");
}
}



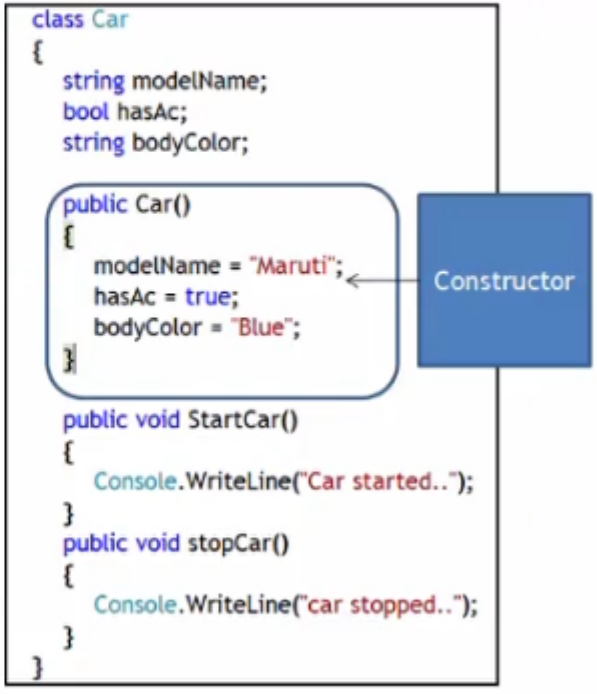
Now you can see as i told you, when i pass 2 parameter so it will call the 0 parameter 1st and then 1 parameter constructor gets called

So we are Chaining the Constructor here

So, by using this you can always call the object or the constructor of the same class

Constructor

- A constructor is a special method that is called to initialize default values to data members (fields) of a class
- It does not create object
- Constructor's name is same as that of the class.
- Constructor does not return anything



Default Constructor

- Features of a default constructor
  - Public accessibility
  - Same name as the class
  - No return type—not even **void**
  - Expects no arguments
  - Initializes all fields to **zero**, **false** or **null**
- Constructor Example:

```
class Date
{
        public Date( )
        { … }
}
```

Explicitly Providing the Default Constructor

- Default constructor will be supplied by compiler
- But, the default constructor might be inappropriate
  - If so, do not use it; write your own!
  - If you are providing default constructor, then compiler will not supply the default constructor

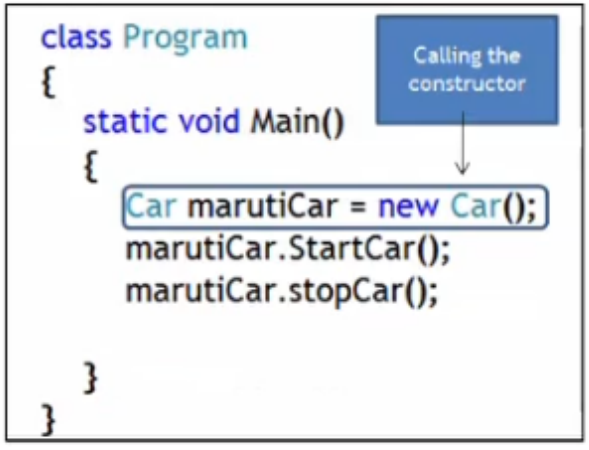
```
class Date
{
   private int ccyy;
   private int mm;
   private int dd;

   public Date( )
   {
      ccyy = 1970;
      mm = 1;
      dd = 1;
   }
}
```

Remember Constructors can be private

Calling Constructor

- Initialize the object's fields with default values by using a constructor
  - Use the name of the class followed by parentheses followed by 'new' keyword
  - As default value, zero is assigned to numerical fields (int, double, long etc.), null is assigned to reference types (string or any class variables) and false is assigned to boolean types



Access Specifiers

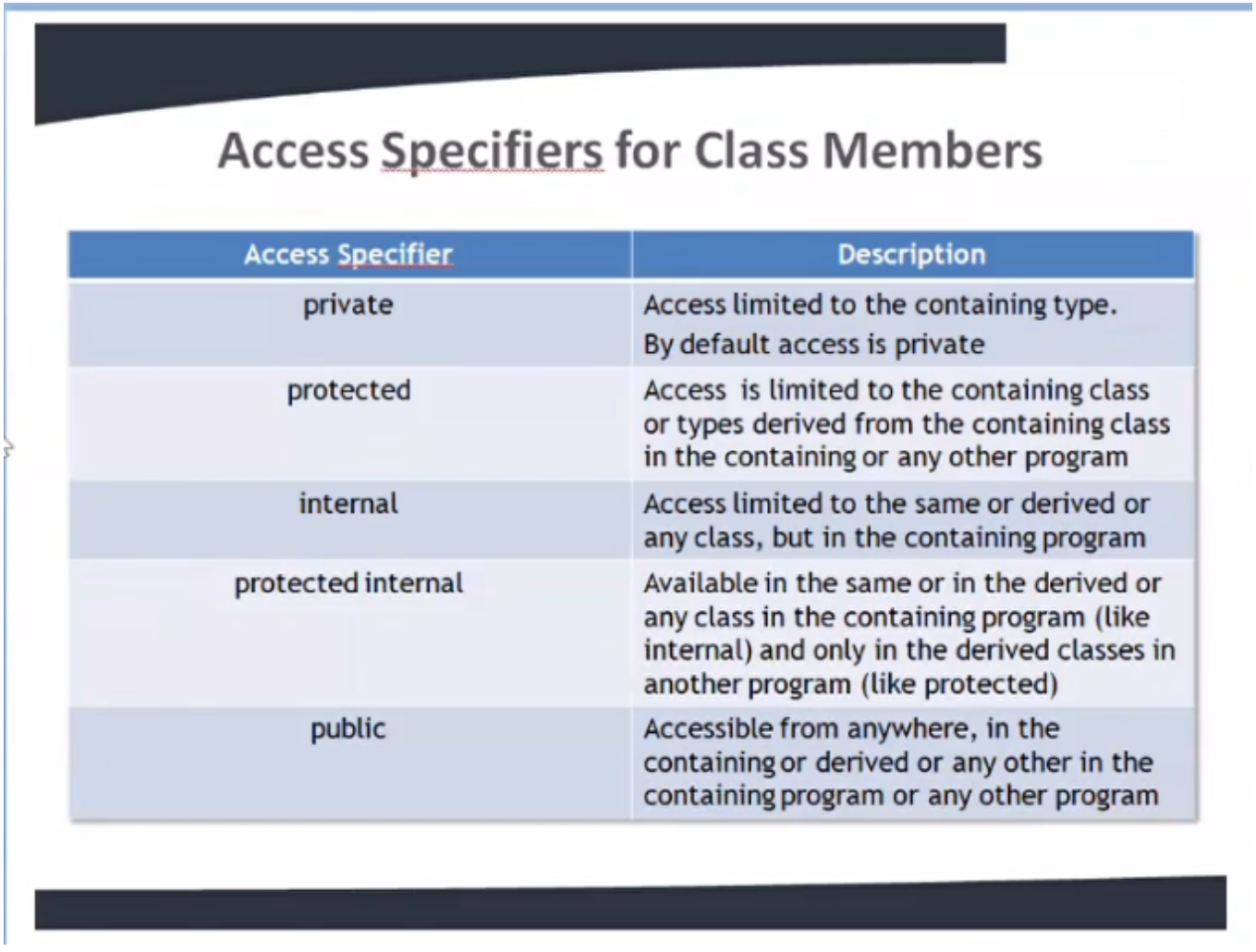
This id for the Class level

Access Specifiers for a Class

- Class has two access specifiers

Access Specifier	Description
internal	A class declared with this access specifier is accessible from anywhere within the containing program, but not accessible from outside that program
public	A class declared with this access specifier is accessible from anywhere within the containing program as well as accessible from outside that program

This id for the Method level

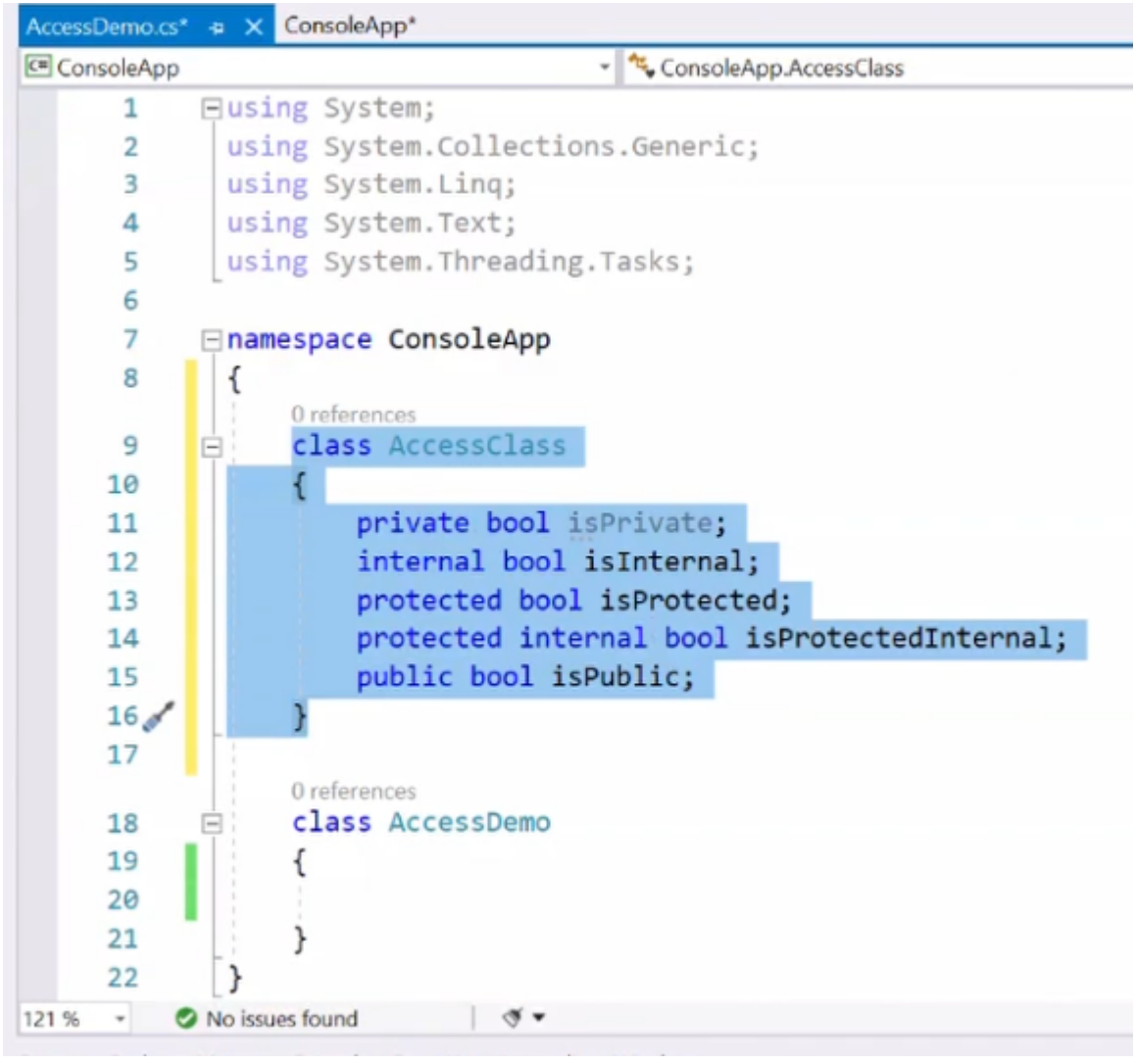


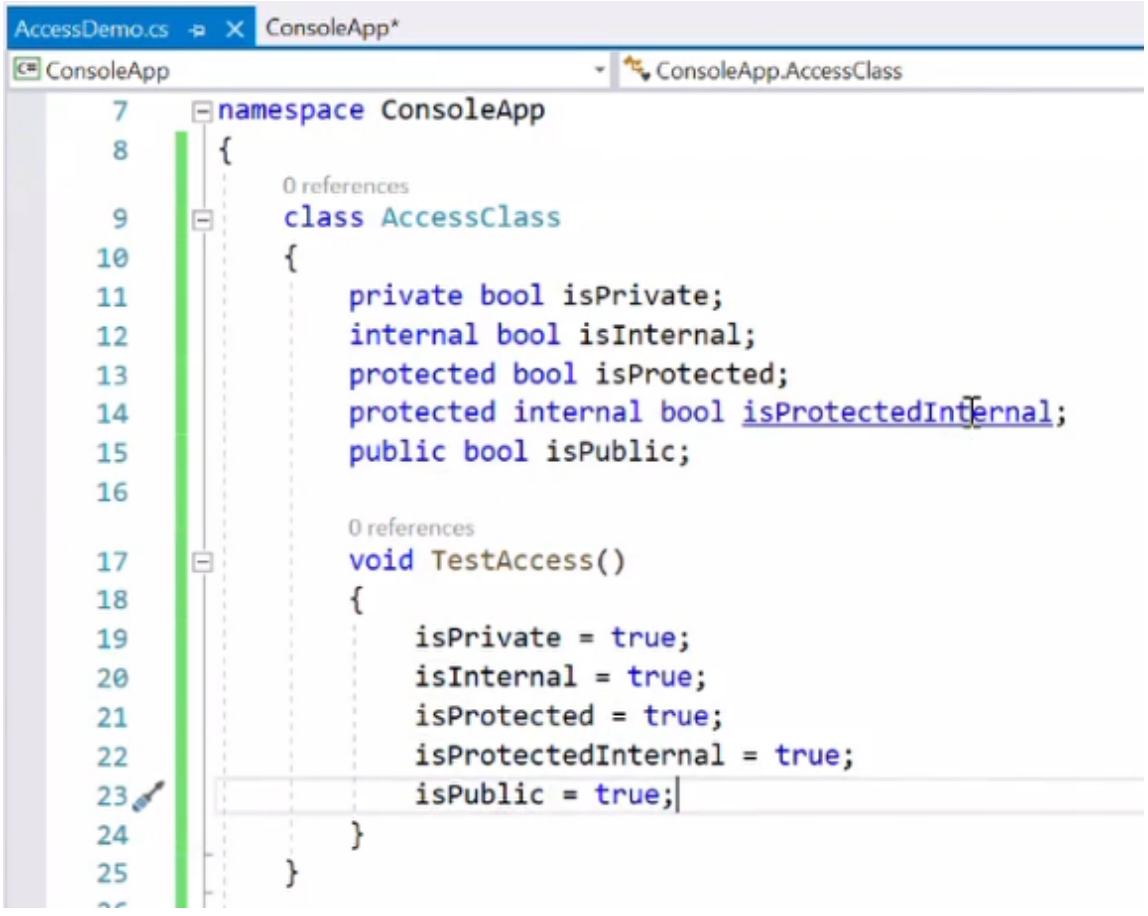
try this yourself, add new file AccessDemo.cs

i'm checking within the **same class** can i Access all of them, if i can access all of them you have to say yes if can't access you have to say no

So 1st

lets create a Access Class

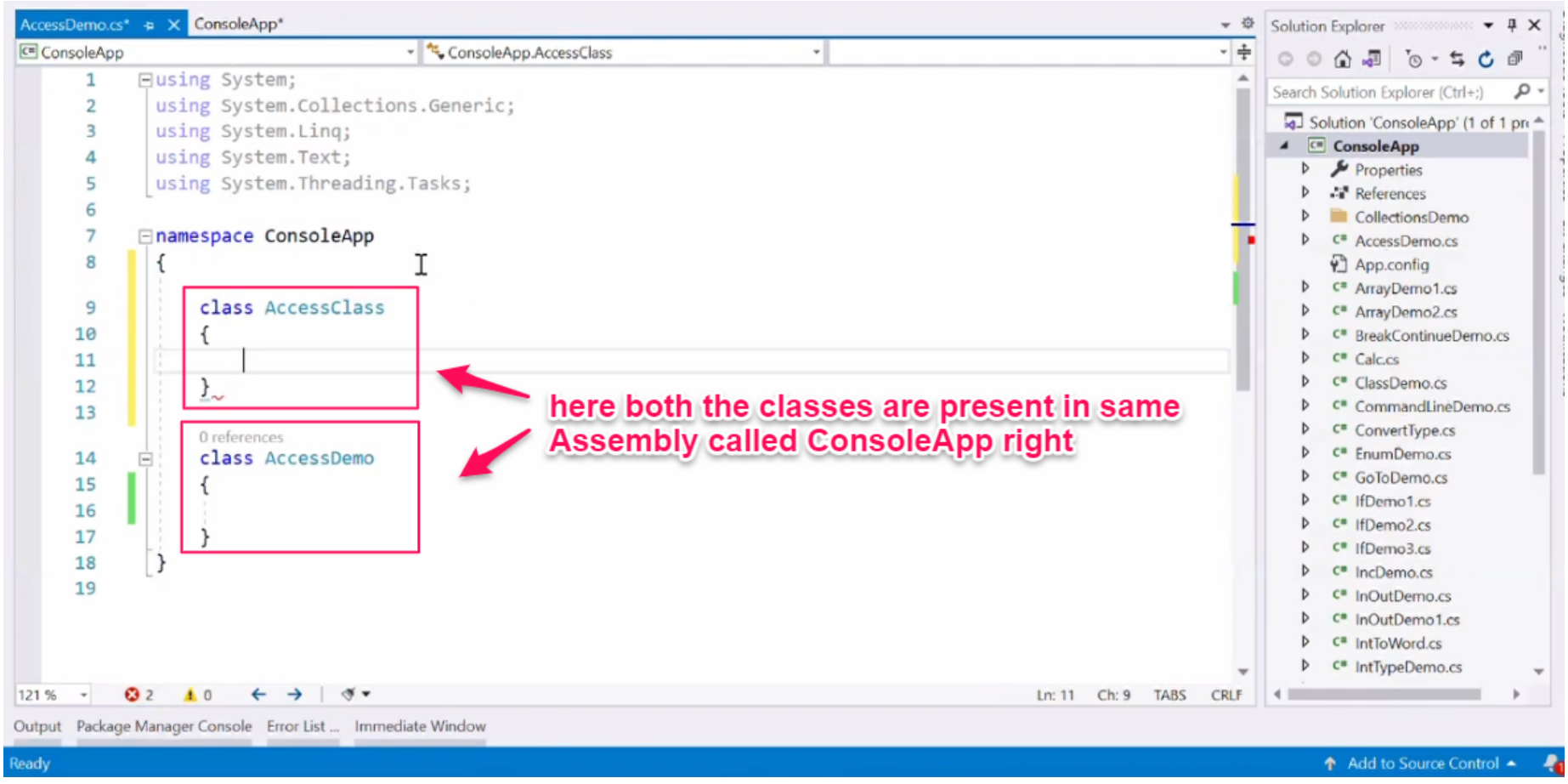

AccessDemo.cs*
ConsoleApp*
ConsoleApp
ConsoleApp.AccessClass
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class AccessClass
{
private bool isPrivate;
internal bool isInternal;
protected bool isProtected;
protected internal bool isProtectedInternal;
public bool isPublic;
}
0 references
class AccessDemo
{
}
}
121 %
No issues found

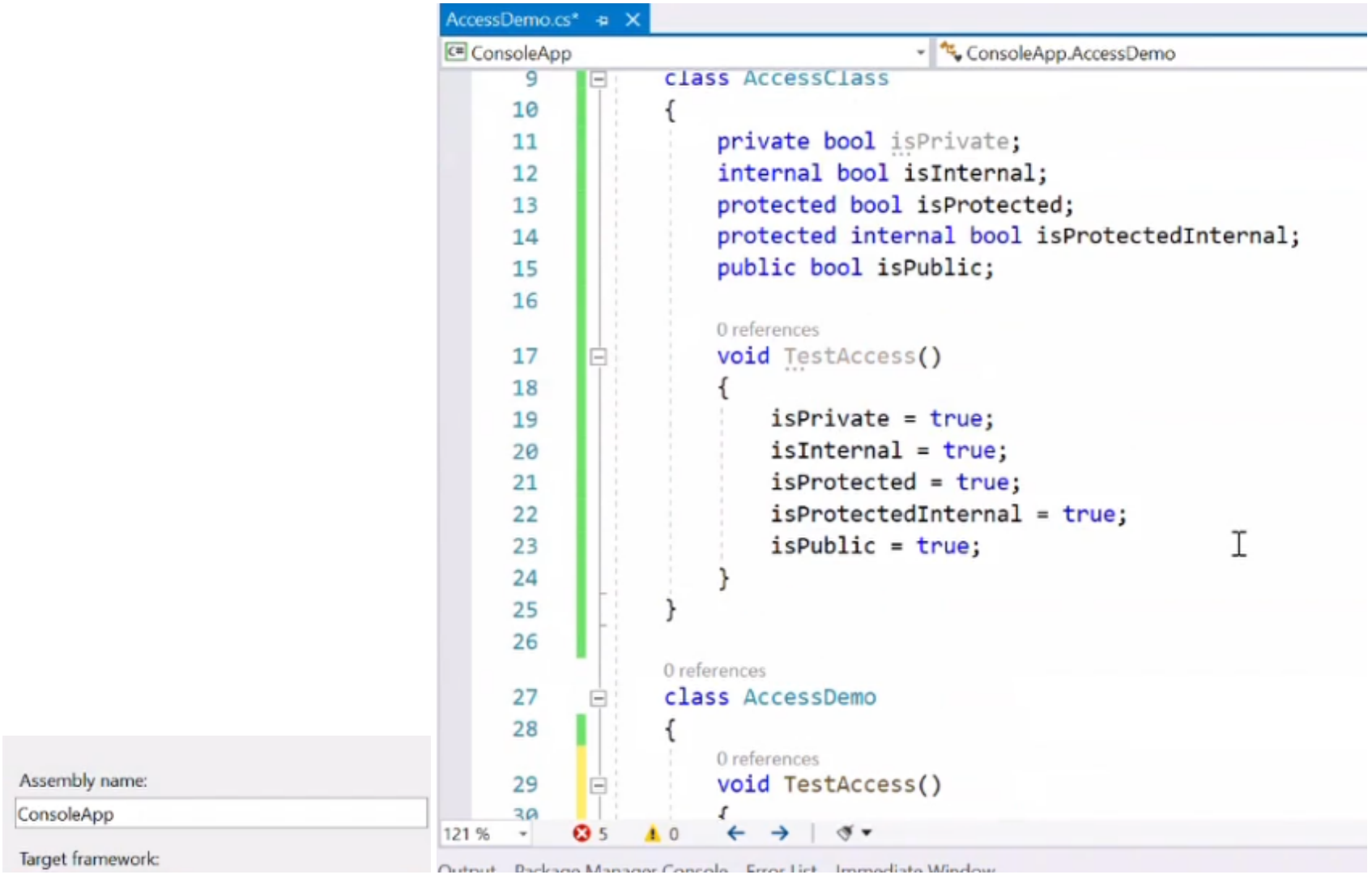


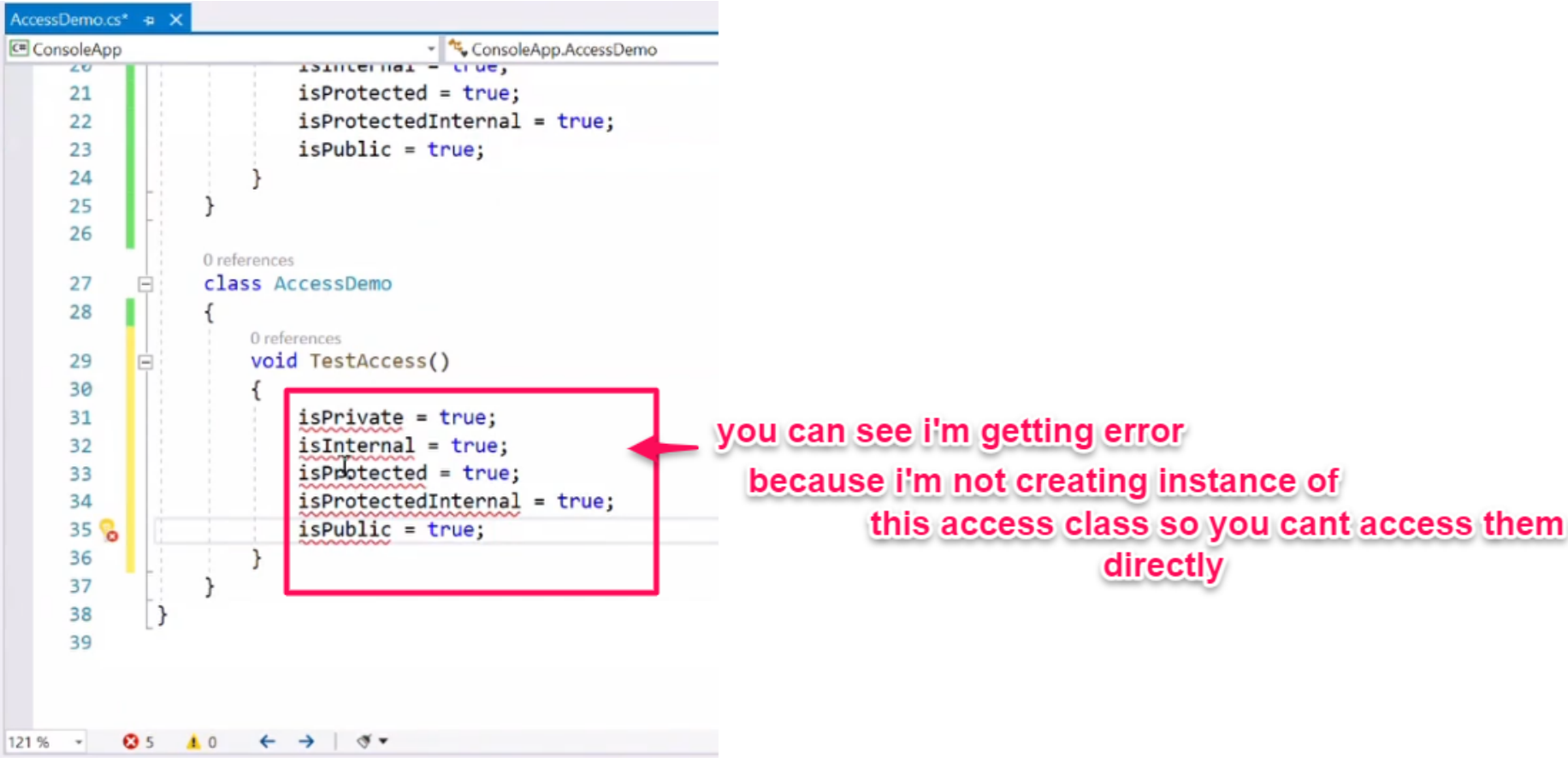
meaning within the same class irrespective of weather it is private or public or whatever i can access them

	Class	Same Assembly	Same Assembly Inherited	Other Assembly Non-Inh	Other Assembly Inherited	Any
Private	Y					
Internal	Y					
Protected	Y					
Protected Internal	Y					
Public	Y					

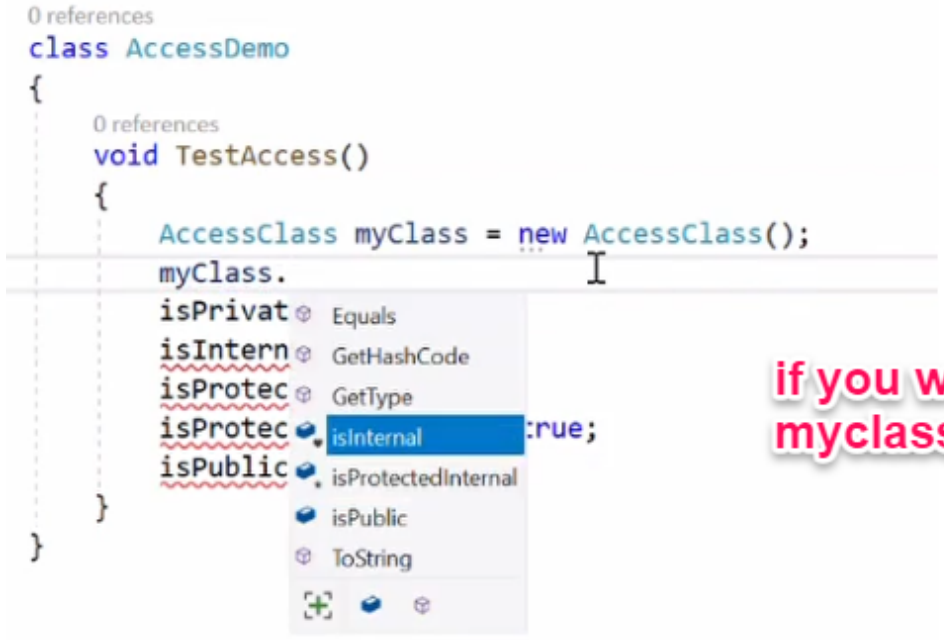
Same Assembly


AccessDemo.cs*
ConsoleApp*
ConsoleApp
ConsoleApp.AccessClass
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
class AccessClass
{
}
0 references
class AccessDemo
{
}
}
here both the classes are present in same
Assembly called ConsoleApp right
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ConsoleApp' (1 of 1 pr
ConsoleApp
Properties
References
CollectionsDemo
AccessDemo.cs
App.config
ArrayDemo1.cs
ArrayDemo2.cs
BreakContinueDemo.cs
Calc.cs
ClassDemo.cs
CommandLineDemo.cs
ConvertType.cs
EnumDemo.cs
GoToDemo.cs
IfDemo1.cs
IfDemo2.cs
IfDemo3.cs
IncDemo.cs
InOutDemo.cs
InOutDemo1.cs
IntToWord.cs
IntTypeDemo.cs
121 %
2
0
Ln: 11
Ch: 9
TABS
CRLF
Output
Package Manager Console
Error List ...
Immediate Window
Ready
Add to Source Control

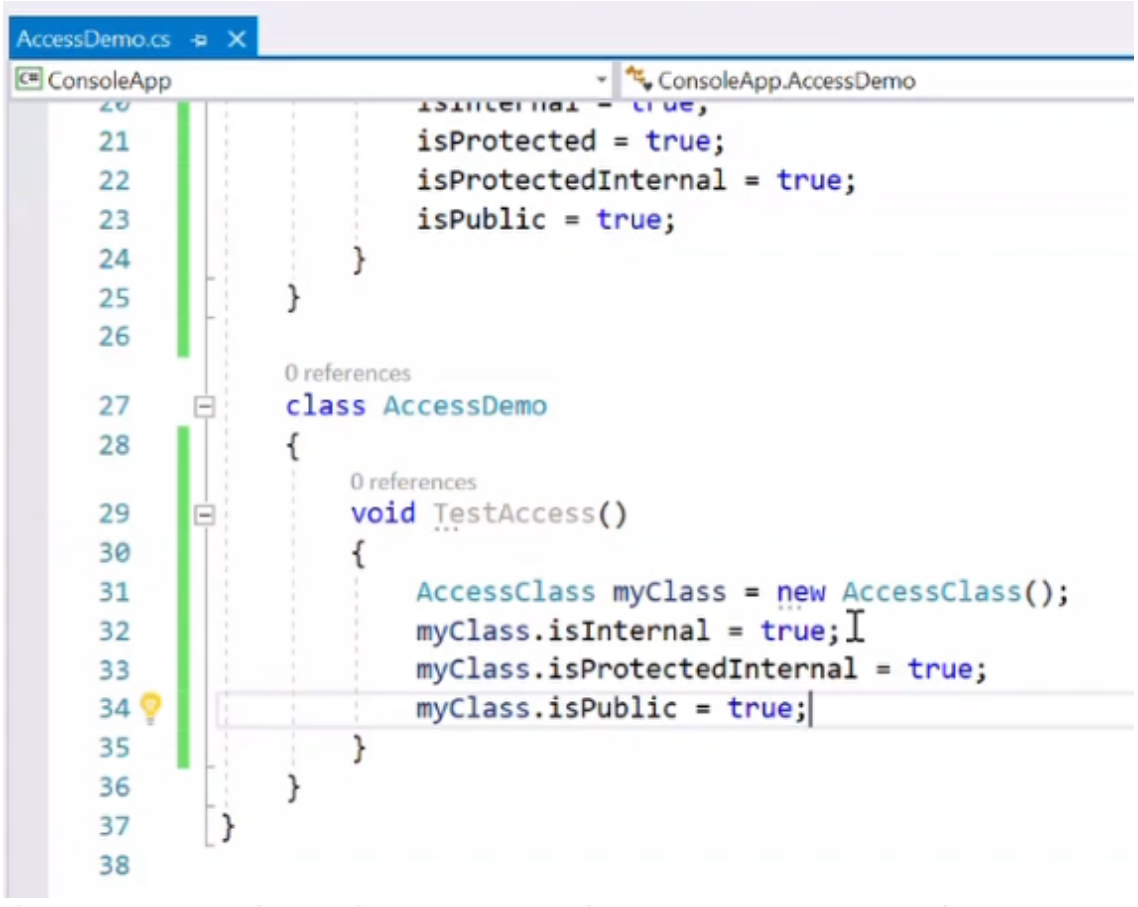

AccessDemo.cs*
ConsoleApp
ConsoleApp.AccessDemo
class AccessClass
{
private bool isPrivate;
internal bool isInternal;
protected bool isProtected;
protected internal bool isProtectedInternal;
public bool isPublic;
0 references
void TestAccess()
{
isPrivate = true;
isInternal = true;
isProtected = true;
isProtectedInternal = true;
isPublic = true;
}
}
0 references
class AccessDemo
{
0 references
void TestAccess()
121 %
5
0
Assembly name:
ConsoleApp
Target framework:



So if you want to access any of these you have to create instance of this particular access class

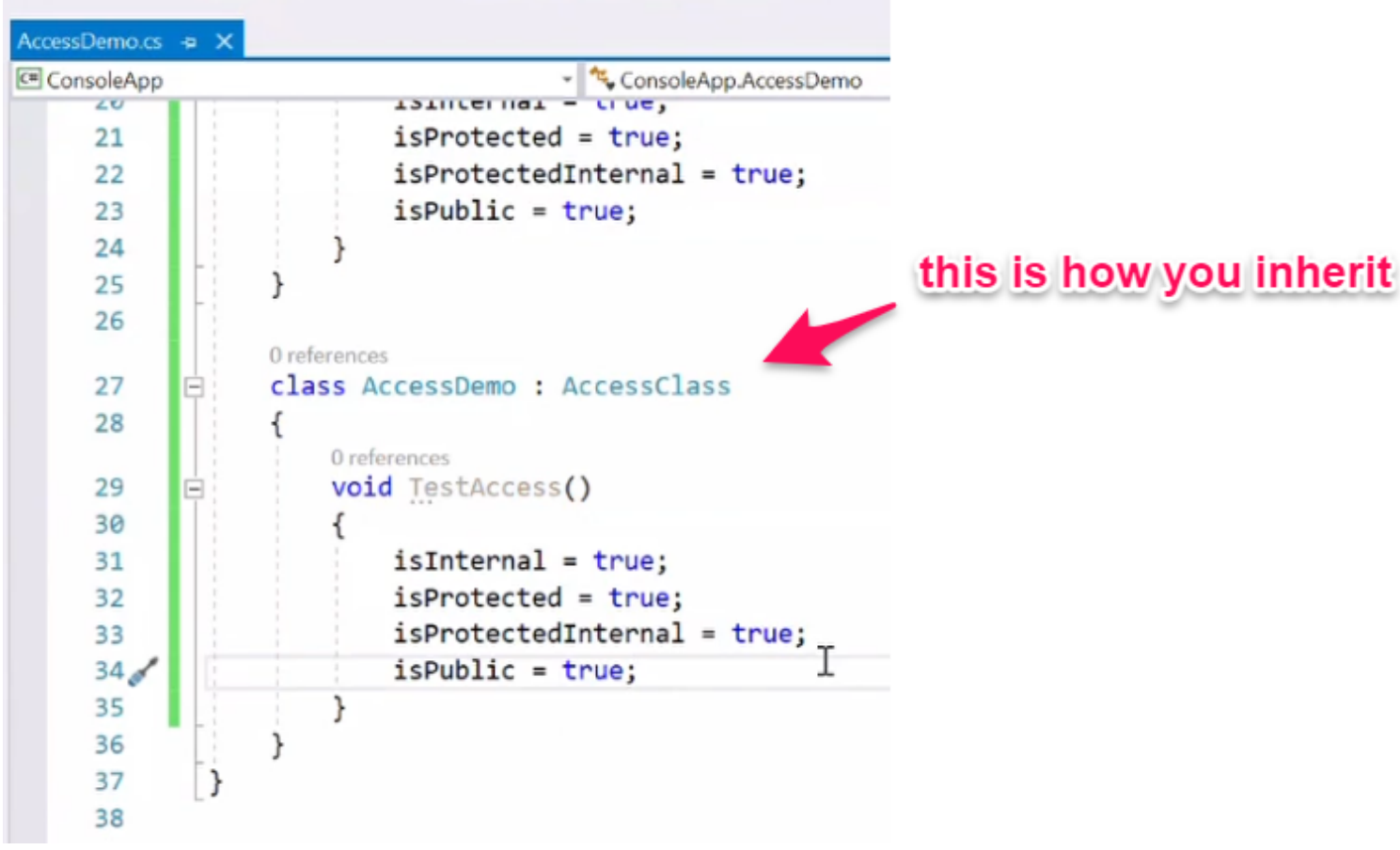


if you write myclass. you can check what you can access



	Class	Same Assembly	Same Assembly Inherited	Other Assembly Non-Inh	Other Assembly Inherited	Any
Private	Y	N				
Internal	Y	Y				
Protected	Y	N				
Protected Internal	Y	Y				
Public	Y	Y				

Same Assembly Inherited



So this is the advantage of Inheritance

If you are inheriting your class

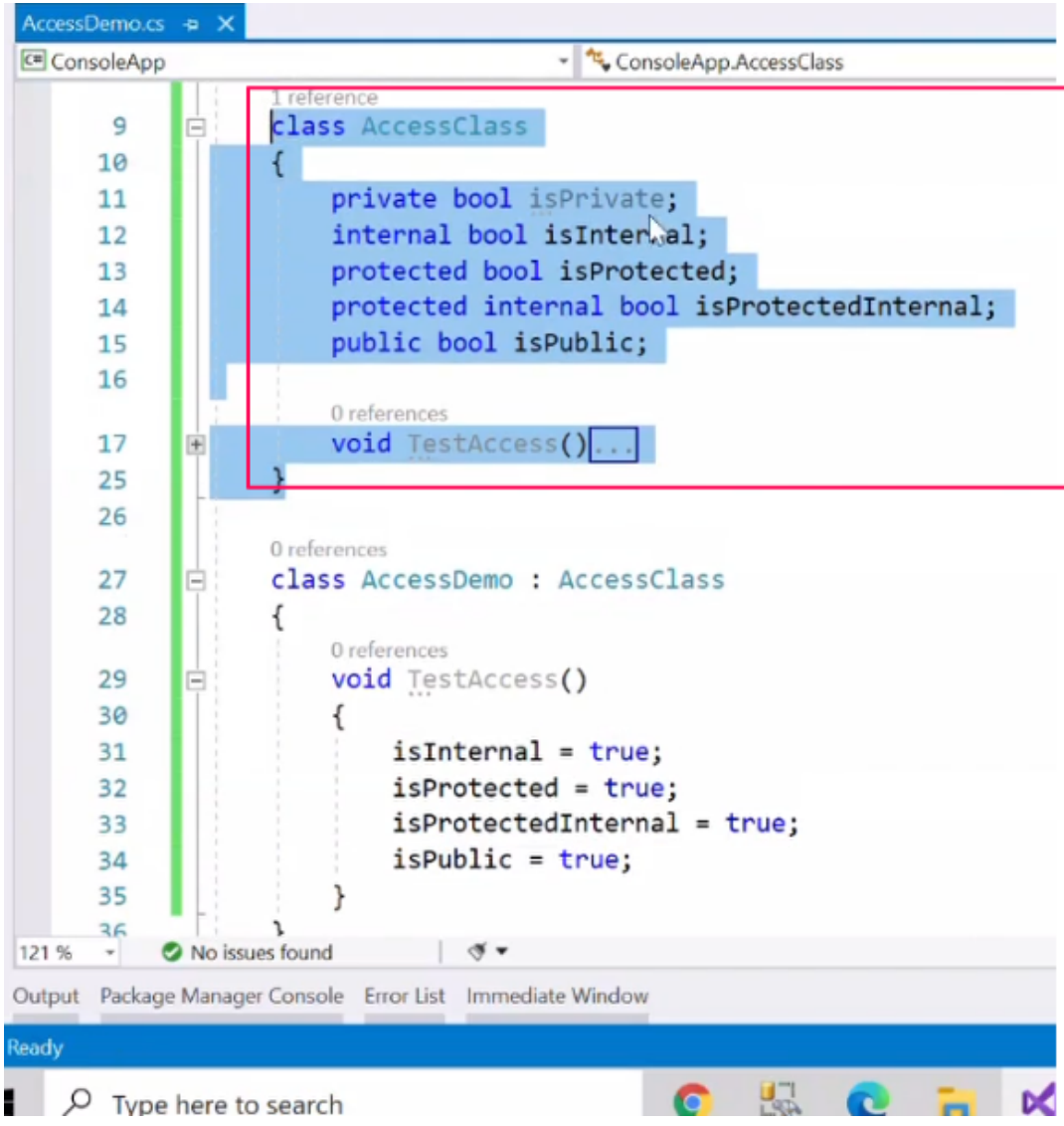
So from the base class what i can access -> internal members, protected members, protected internal members, and public members

You Cannot access your private members here

	Class	Same Assembly	Same Assembly Inherited	Other Assembly Non-Inh	Other Assembly Inherited	Any
Private	Y	N	N			
Internal	Y	Y	Y			
Protected	Y	N	Y			
Protected Internal	Y	Y	Y			
Public	Y	Y	Y			

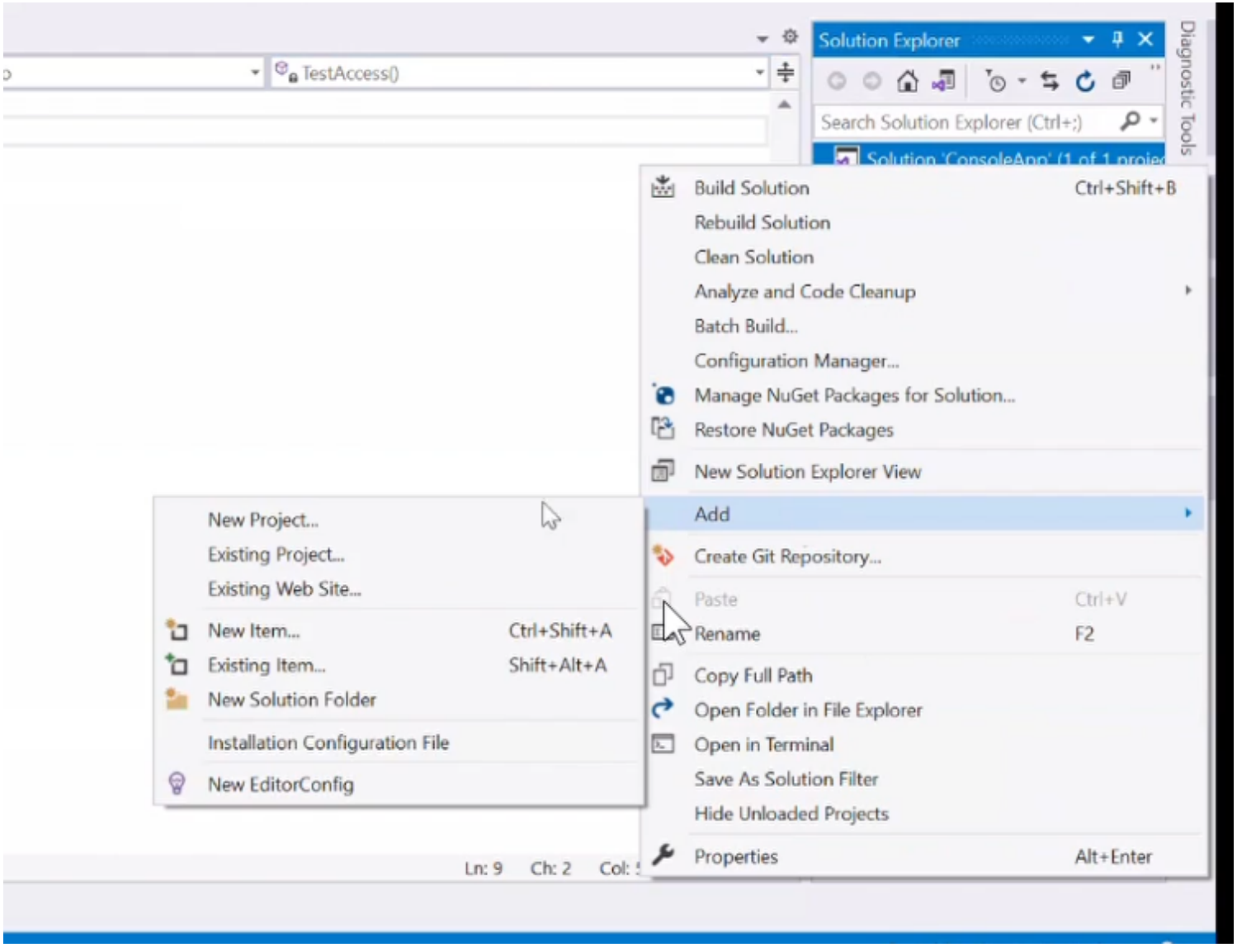
Other Assembly Non-Inherited

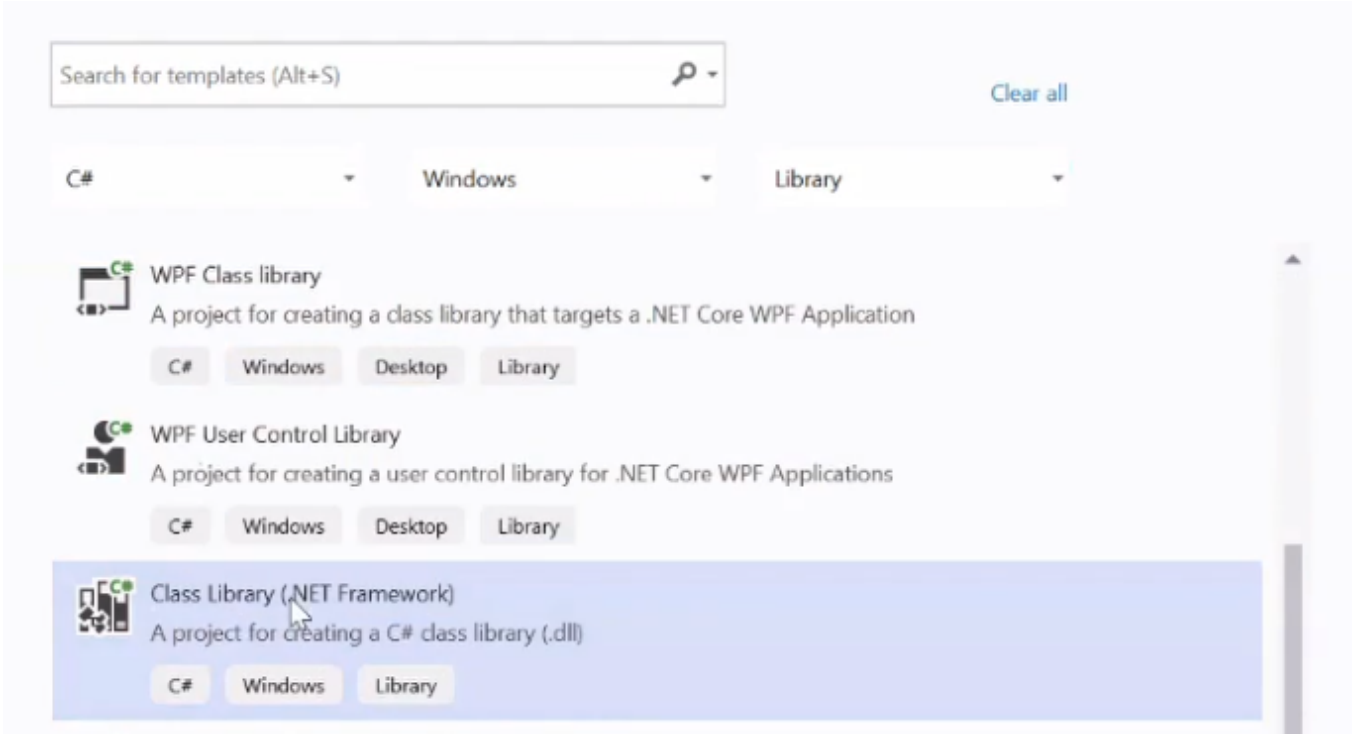
other Assembly means this class is not present here but whereas present in some other assembly

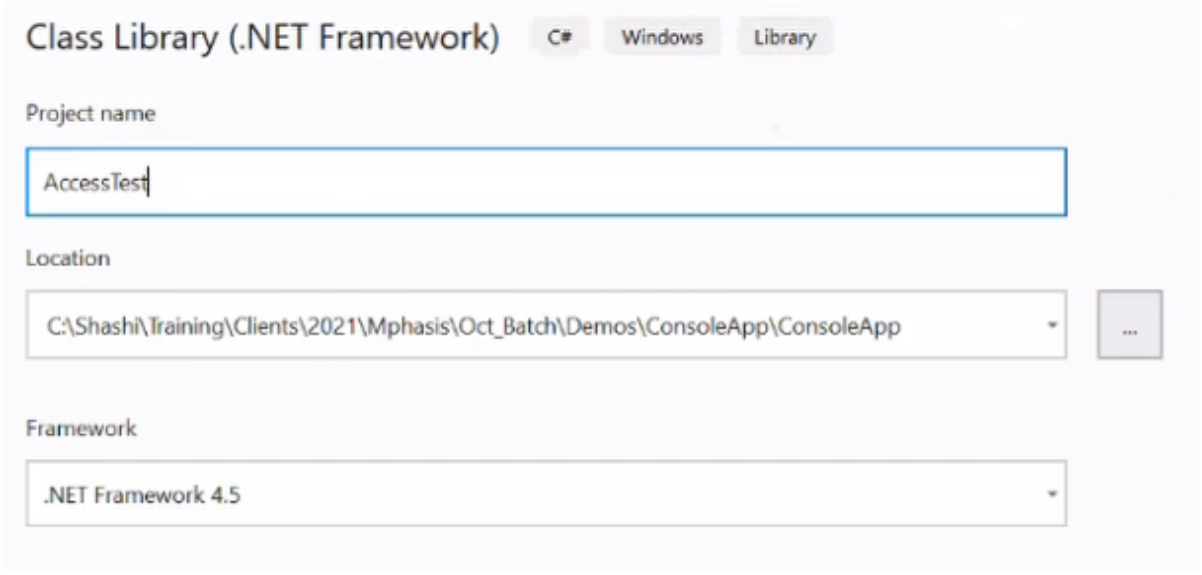


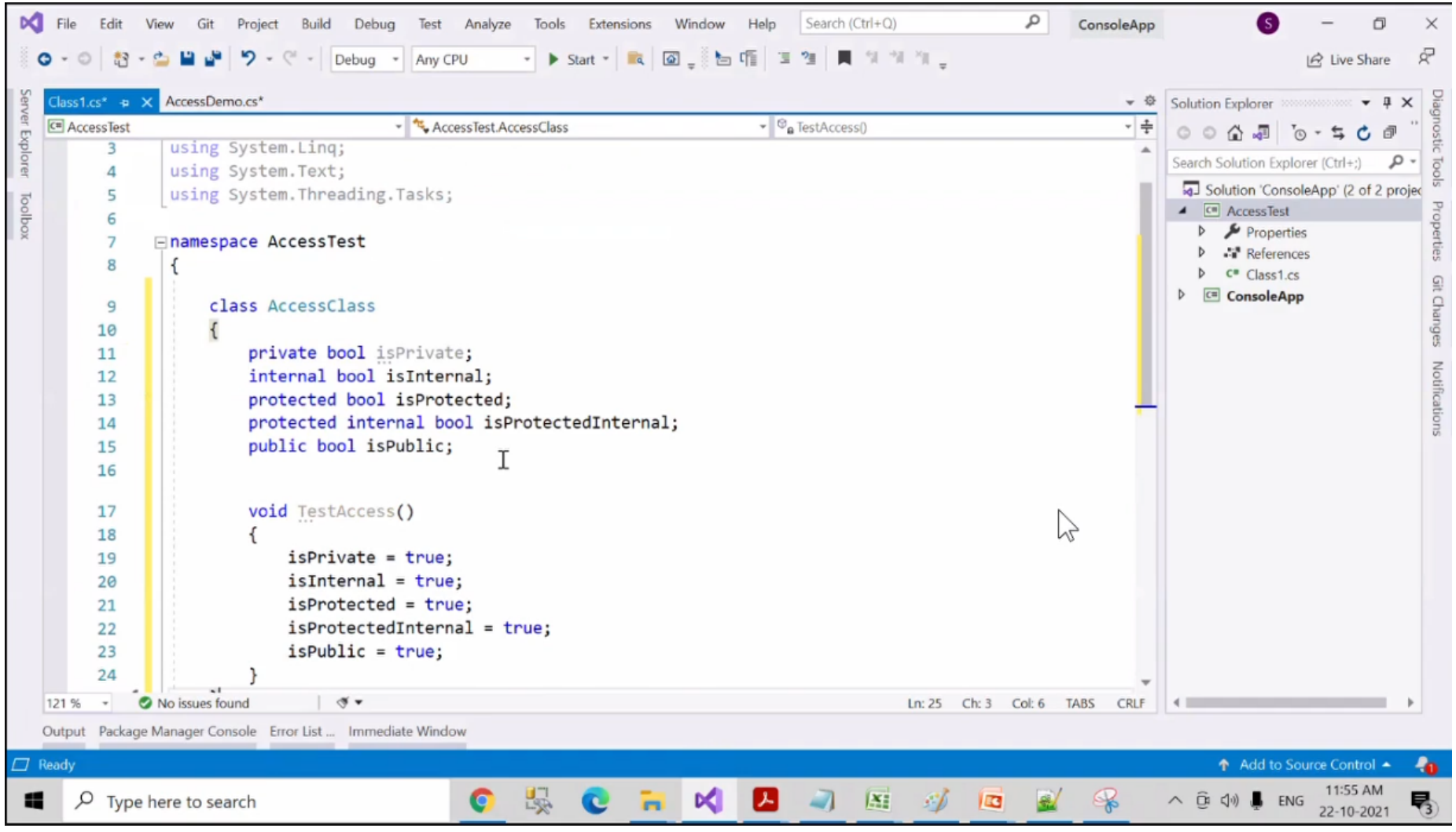
other Assembly means this class is not present here but whereas present in some other assembly / Project

right click on solution ---> add new project---->this time i'll not take it as a console rather i'll take it as a liabrary

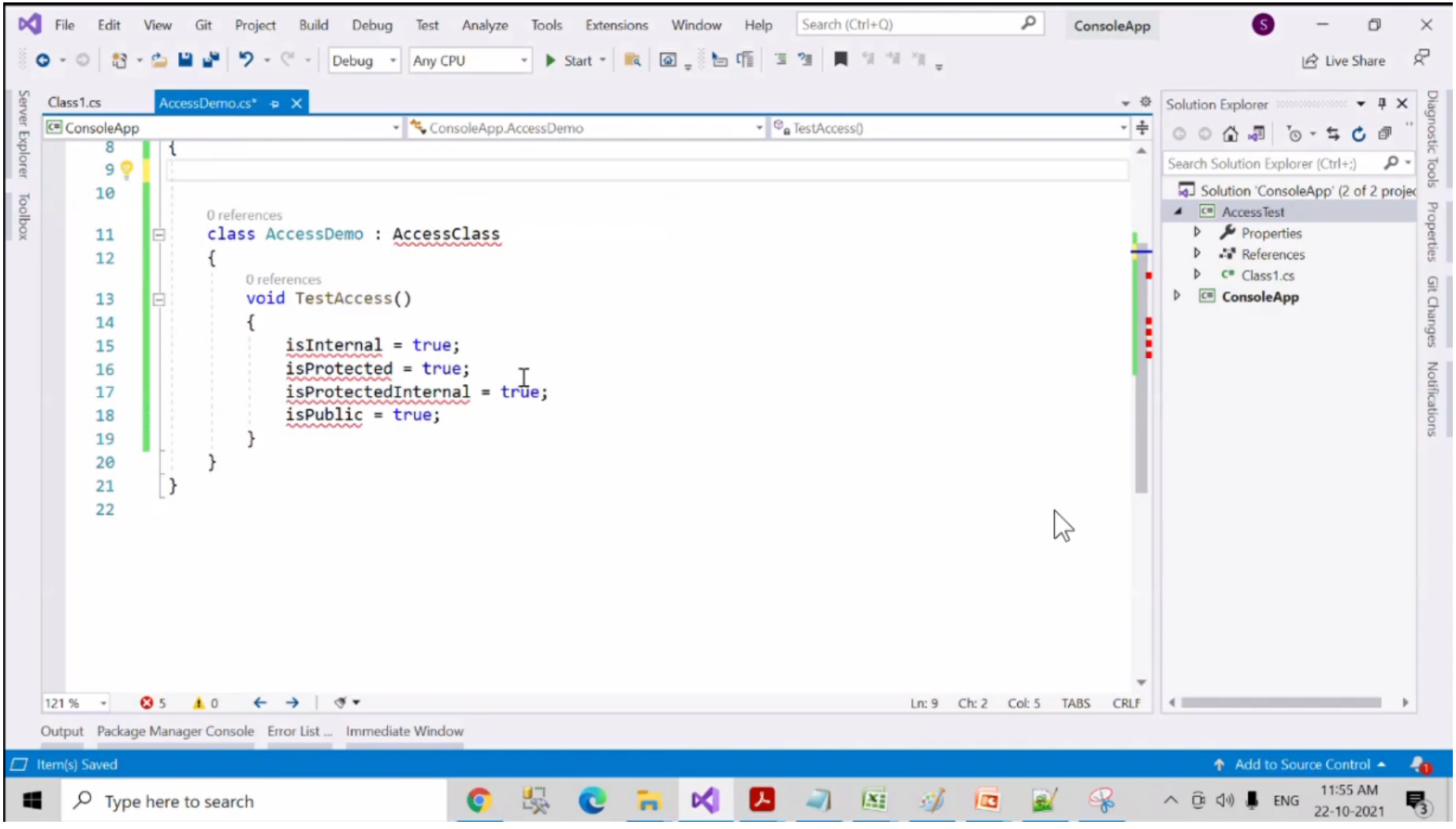

Solution Explorer
Search Solution Explorer (Ctrl+;)
Diagnostic Tools
TestAccess()
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Analyze and Code Cleanup
Batch Build...
Configuration Manager...
Manage NuGet Packages for Solution...
Restore NuGet Packages
New Solution Explorer View
Add
Create Git Repository...
Paste Ctrl+V
Rename F2
Copy Full Path
Open Folder in File Explorer
Open in Terminal
Save As Solution Filter
Hide Unloaded Projects
Properties Alt+Enter
New Project...
Existing Project...
Existing Web Site...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Solution Folder
Installation Configuration File
New EditorConfig
Ln: 9 Ch: 2


Search for templates (Alt+S)
Clear all
C#
Windows
Library
WPF Class library
A project for creating a class library that targets a .NET Core WPF Application
C#
Windows
Desktop
Library
WPF User Control Library
A project for creating a user control library for .NET Core WPF Applications
C#
Windows
Desktop
Library
Class Library (.NET Framework)
A project for creating a C# class library (.dll)
C#
Windows
Library


Class Library (.NET Framework)
C#
Windows
Library
Project name
AccessTest
Location
C:\Shashi\Training\Clients\2021\Mphasis\Oct_Batch\Demos\ConsoleApp\ConsoleApp
...
Framework
.NET Framework 4.5


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
ConsoleApp
Debug
Any CPU
Start
Live Share
Class1.cs*
AccessDemo.cs*
AccessTest
AccessTest.AccessClass
TestAccess()
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace AccessTest
{
class AccessClass
{
private bool isPrivate;
internal bool isInternal;
protected bool isProtected;
protected internal bool isProtectedInternal;
public bool isPublic;
void TestAccess()
{
isPrivate = true;
isInternal = true;
isProtected = true;
isProtectedInternal = true;
isPublic = true;
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ConsoleApp' (2 of 2 projec
AccessTest
Properties
References
Class1.cs
ConsoleApp
121 %
No issues found
Ln: 25 Ch: 3 Col: 6 TABS CRLF
Output
Package Manager Console
Error List ...
Immediate Window
Ready
Add to Source Control
Type here to search
ENG
11:55 AM
22-10-2021

Now, this and

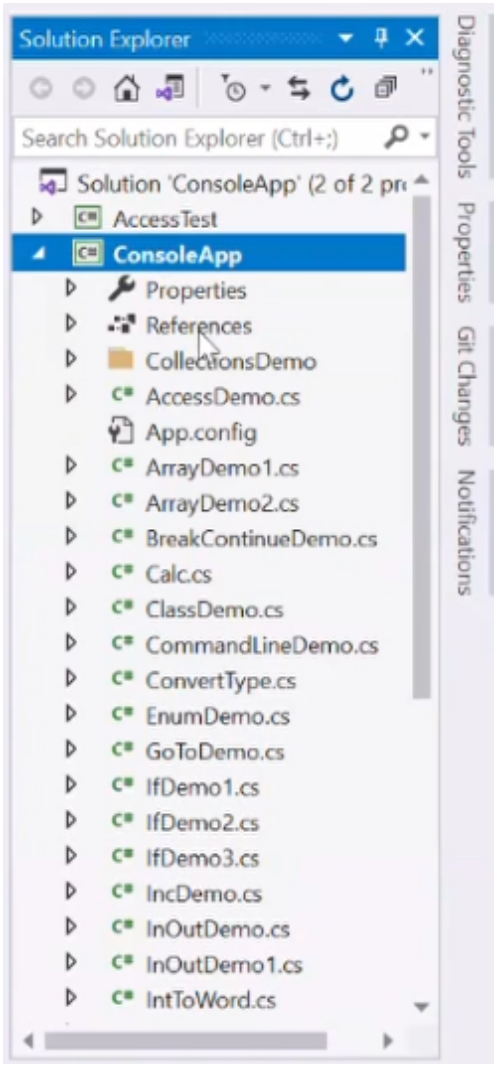


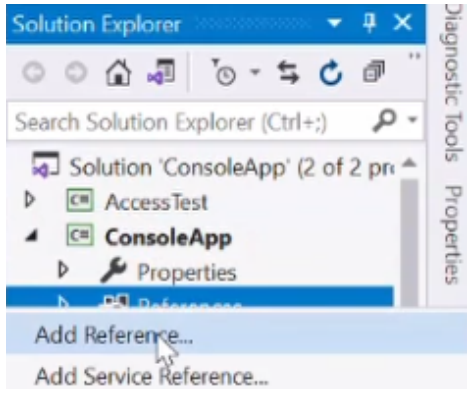
this,

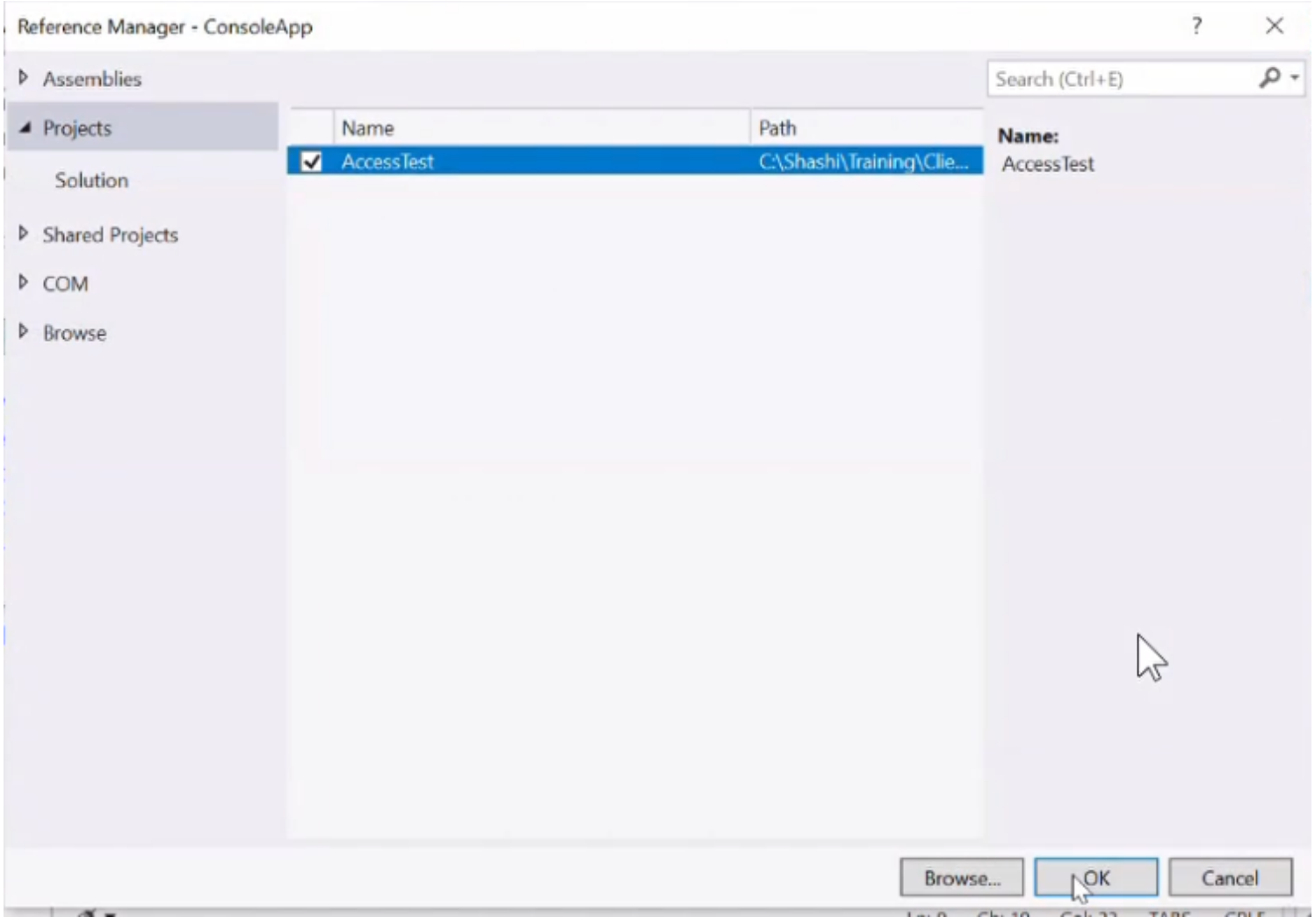
Both are 2 different places

so to access AccessTest in ConsoleApp we need to add reference to them

right click on reference

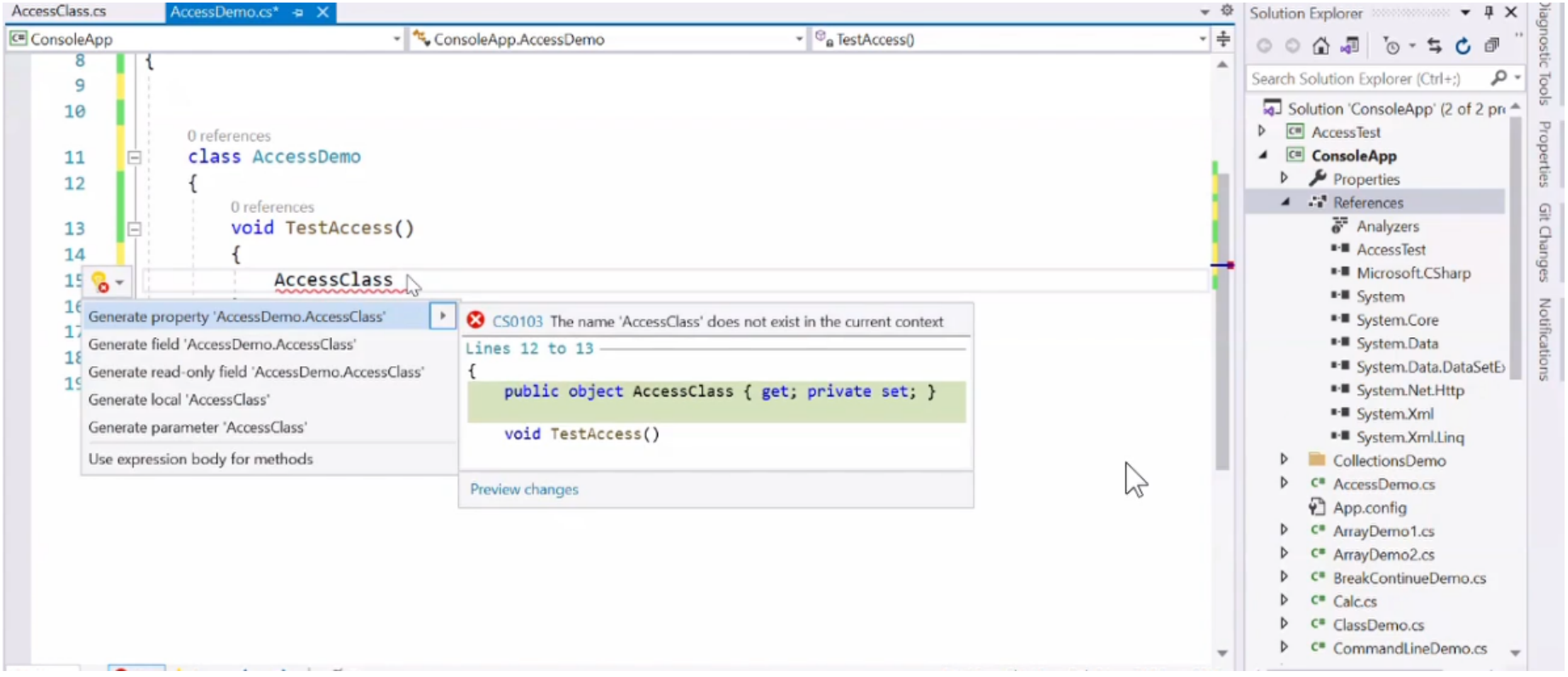

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ConsoleApp' (2 of 2 pr
AccessTest
ConsoleApp
Properties
References
CollectionsDemo
AccessDemo.cs
App.config
ArrayDemo1.cs
ArrayDemo2.cs
BreakContinueDemo.cs
Calc.cs
ClassDemo.cs
CommandLineDemo.cs
ConvertType.cs
EnumDemo.cs
GoToDemo.cs
IfDemo1.cs
IfDemo2.cs
IfDemo3.cs
IncDemo.cs
InOutDemo.cs
InOutDemo1.cs
IntToWord.cs
Diagnostic Tools
Properties
Git Changes
Notifications


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ConsoleApp' (2 of 2 pr
AccessTest
ConsoleApp
Properties
Add Reference...
Add Service Reference...
Diagnostic Tools
Properties



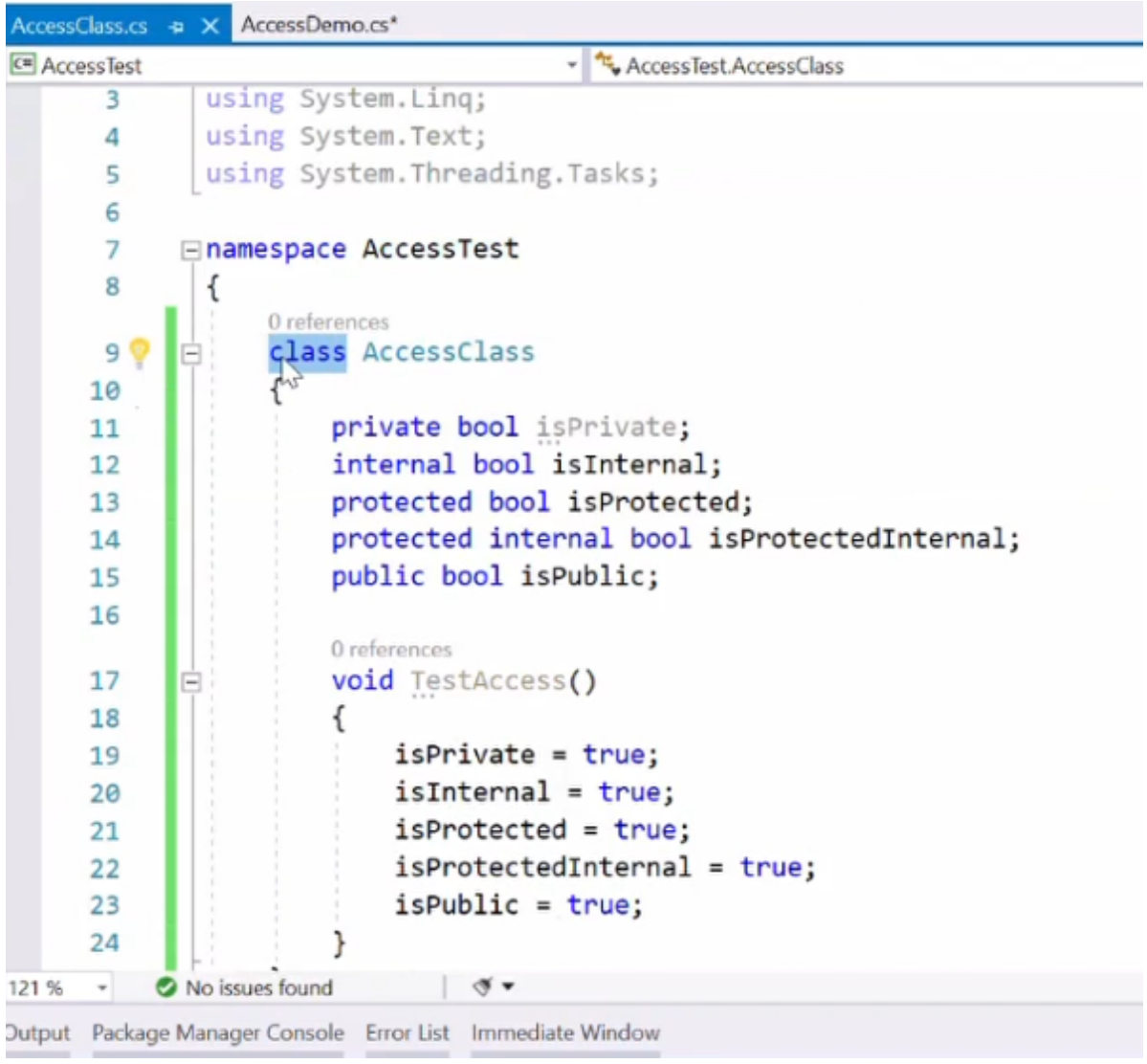
Now i have added a reference i'll come back, and create a reference of it , it will act just like another name space ,

Now my AccessClass is present inside AccessTest project

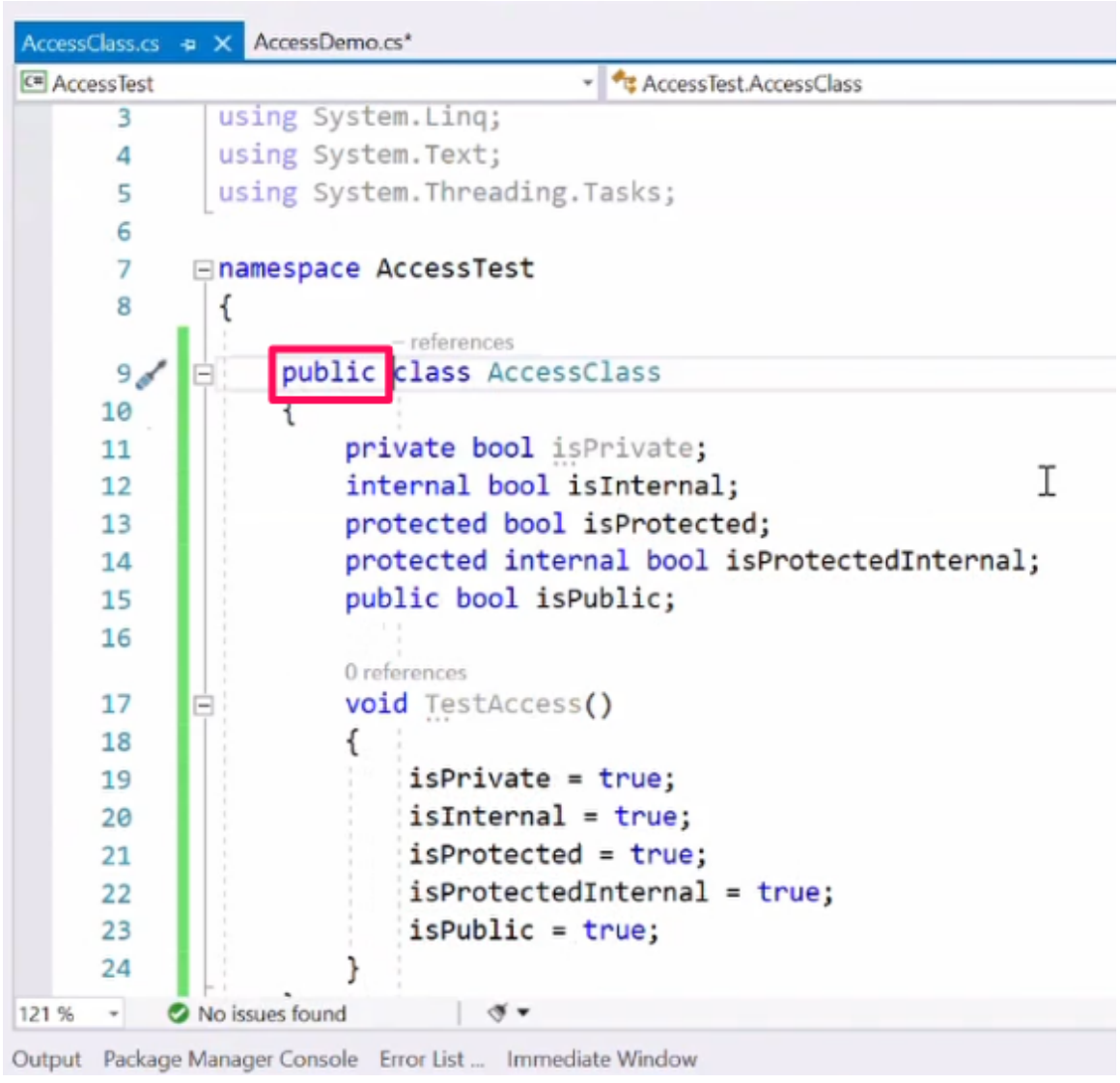


but it is saying to generate the class

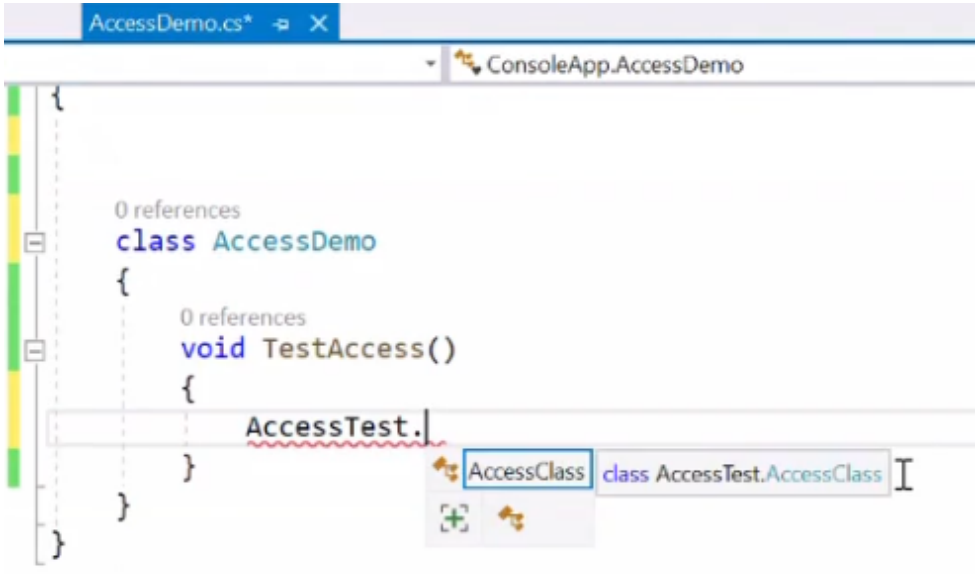
Why because in AccessClass i don't have any public class

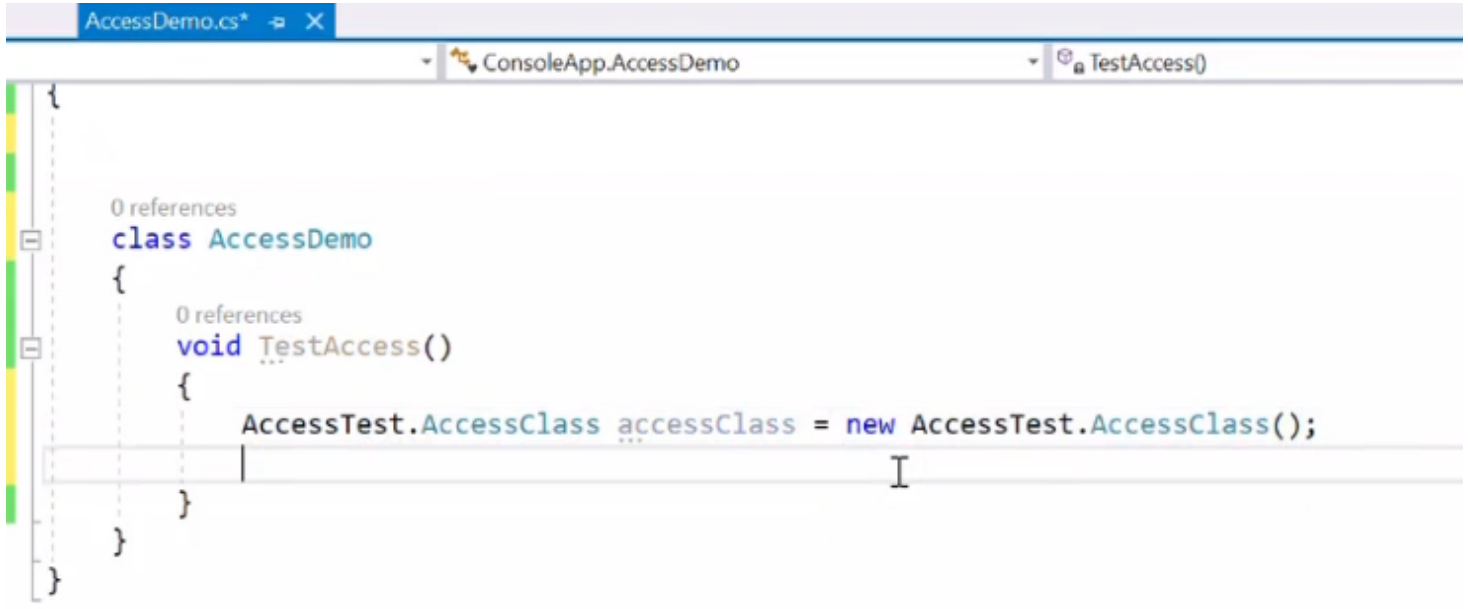


SO to access this AccessClass outside you need to make it public

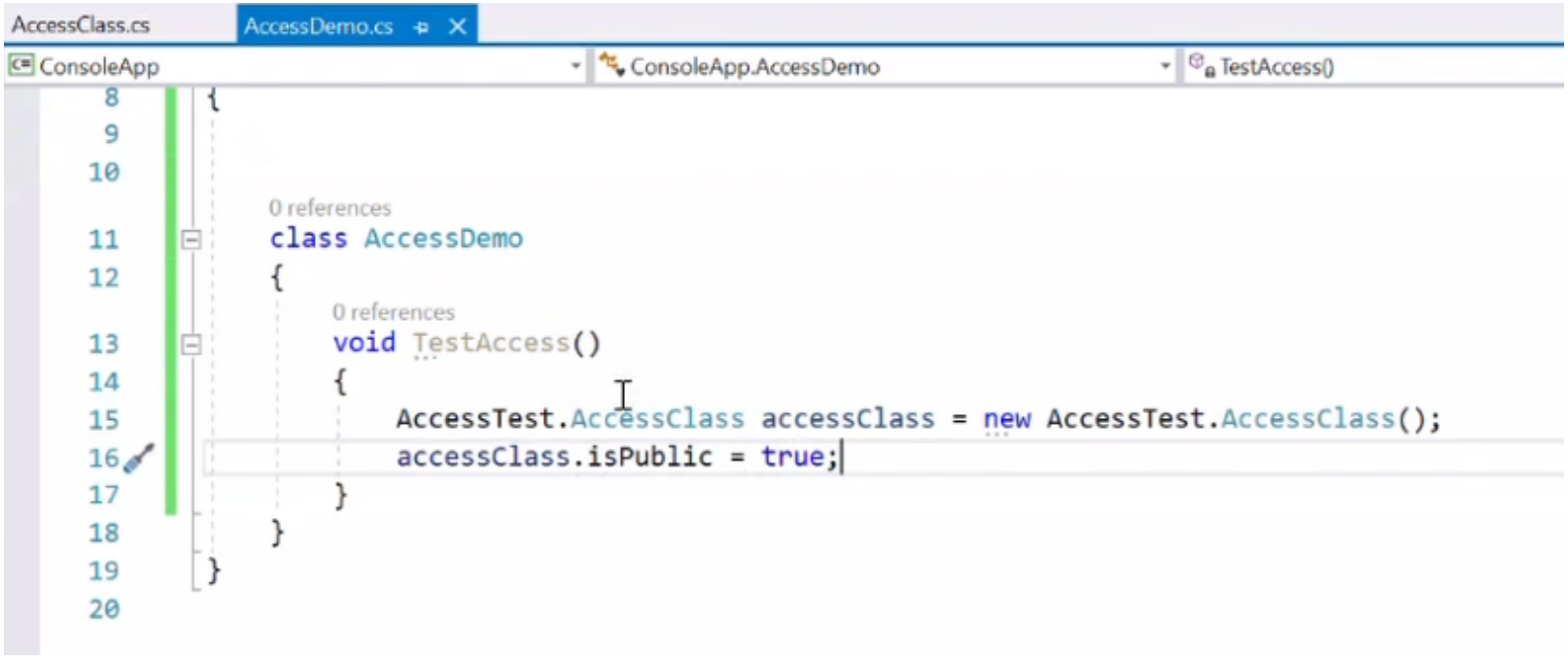


Now look at this i have AccessClass


AccessDemo.cs*
ConsoleApp.AccessDemo
{
0 references
class AccessDemo
{
0 references
void TestAccess()
{
AccessTest.
}
}
}
AccessClass
class AccessTest.AccessClass



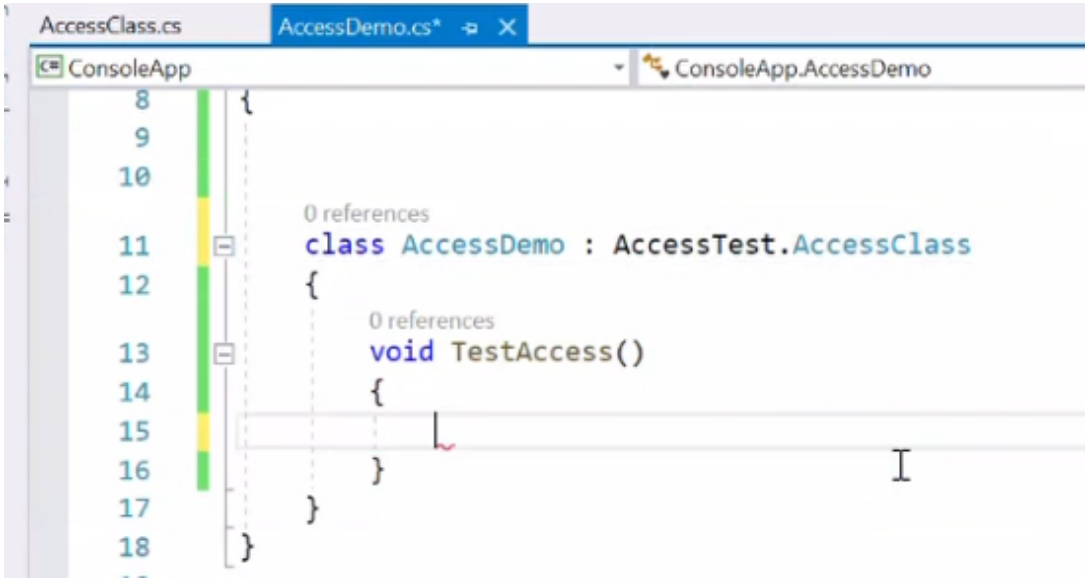
So if you are creating an instance and it is not inherited



	Class	Same Assembly	Same Assembly Inherited	Other Assembly Non-Inh	Other Assembly Inherited	Any
Private	Y	N	N	N		
Internal	Y	Y	Y	N		
Protected	Y	N	Y	N		
Protected Internal	Y	Y	Y	N		
Public	Y	Y	Y	Y		

Other Assembly Inherited

Now lets try to put colon and inherit



Now as soon as i try to inherit in some other assembly

These 3 things can be accessed

	Class	Same Assembly	Same Assembly Inherited	Other Assembly Non-Inh	Other Assembly Inherited	Any
Private	Y	N	N	N	N	
Internal	Y	Y	Y	N	N	
Protected	Y	N	Y	N	Y	
Protected Internal	Y	Y	Y	N	Y	
Public	Y	Y	Y	Y	Y	

now you have to check for any