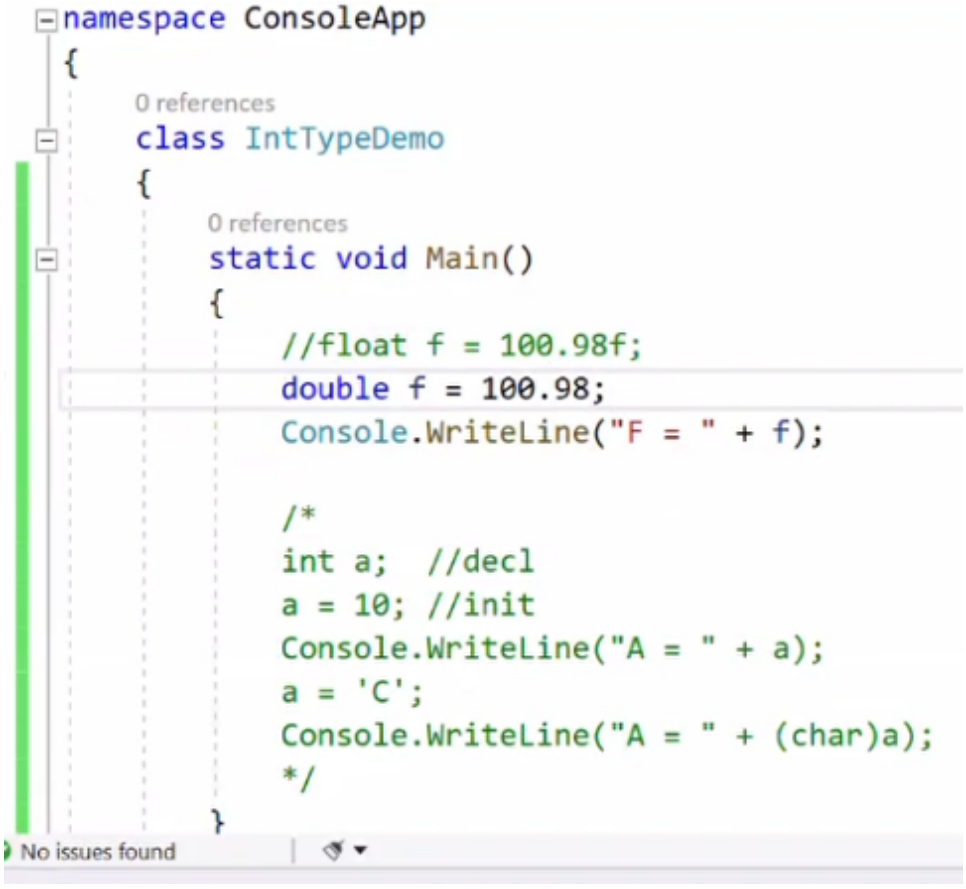
C#-3

but if we write double in place of float then there will be no error



because by default all your floating value will be treated as double

NameSpace, Class Name, MethodName, Interface should follow PascalCasing

almost every thing follow PascalCasing apart from local Variable Name

localVariableName should follow camelCasing

C# Keywords

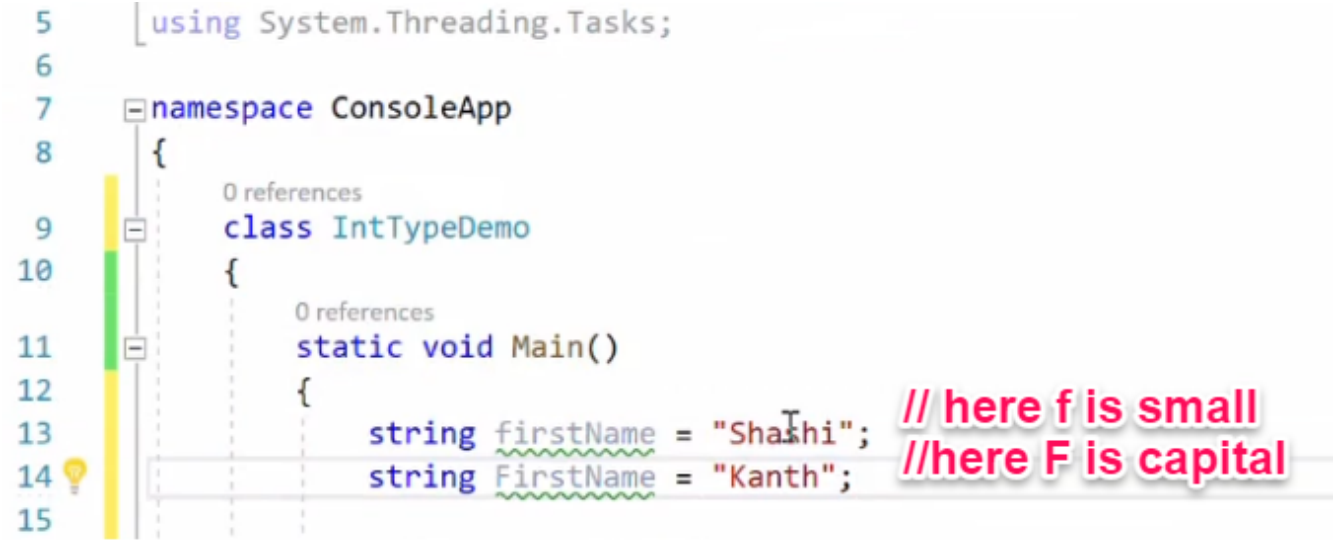
- Keywords are reserved identifiers

```
abstract, base, bool, default, if, finally
```

- Do not use keywords as variable names
  - Results in a compile-time error

- Avoid using keywords by changing their case sensitivity

```
int INT;  // Poor style
```



so my C# will not complaint here because these are 2 different variable even thoe they both have the first name but in first case f is small and in 2nd F is capital , so this is case sensitive

Declaring Local Variables

- Usually declared by data type and variable name:

```
int itemCount;
```

- Possible to declare multiple variables in one declaration:

```
int itemCount, employeeNumber;
```

--or--

```
int itemCount,
    employeeNumber;
```

Assigning Values to Variables

- Assign values to variables that are already

```
int employeeNumber;
employeeNumber = 23;
```

```
int employeeNumber = 23;
```

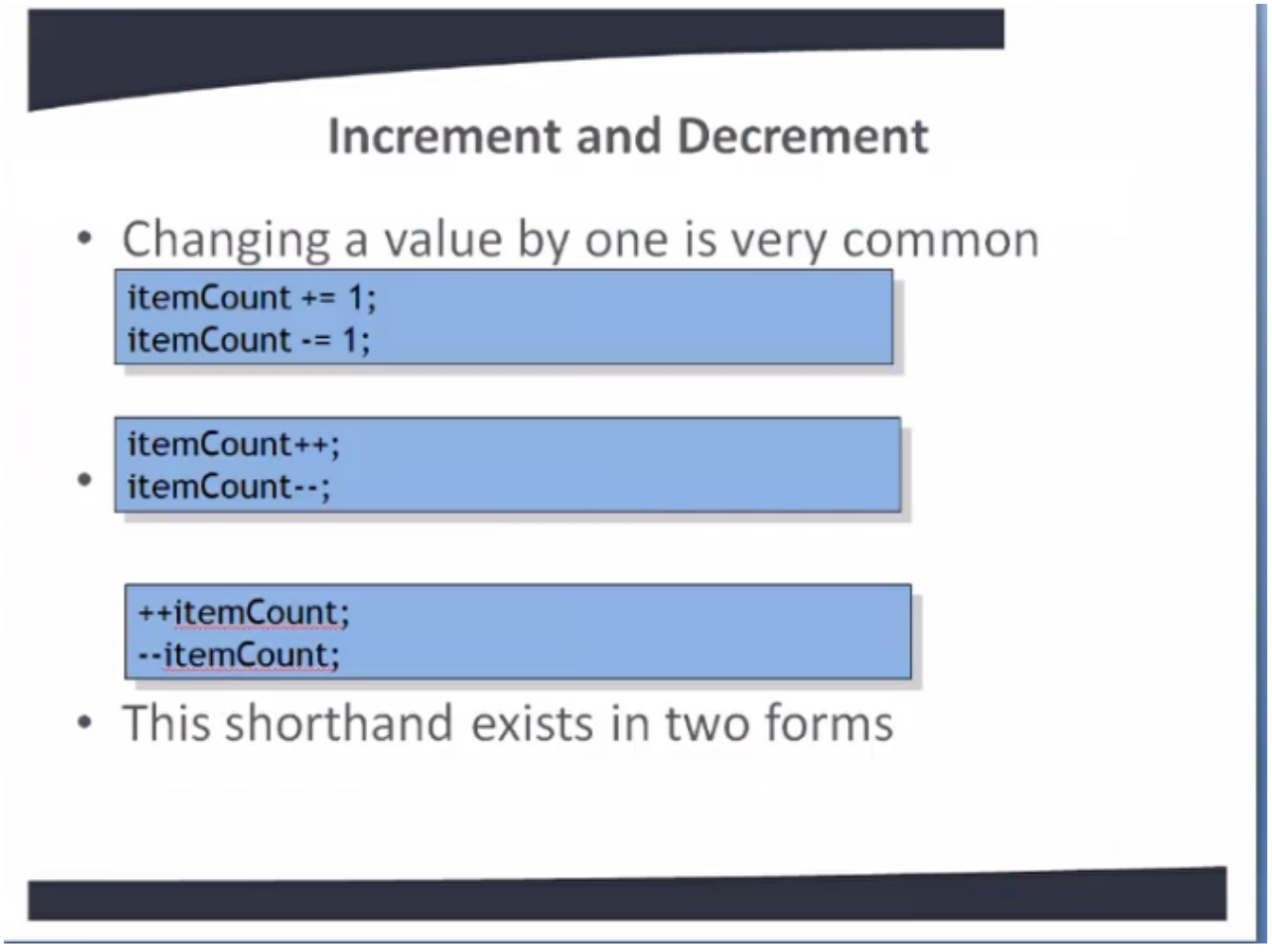
- Initialize a variable when you declare it:

```
char middleInitial = 'J';
```

- You can also initialize character values:

Common Operators

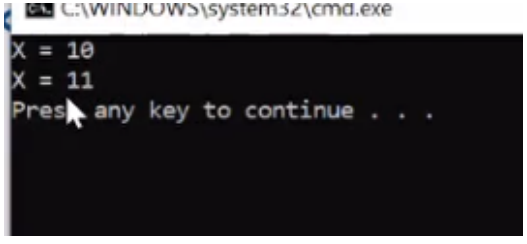
Common Operators	Example
• Equality operators	== !=
• Relational operators	< > <= >= is
• Conditional operators	&& || ?:
• Increment operator	++
• Decrement operator	- -
• Arithmetic operators	+ - * / %
• Assignment operators	= *= /= %= += -= <<= >>= &= ^= |=



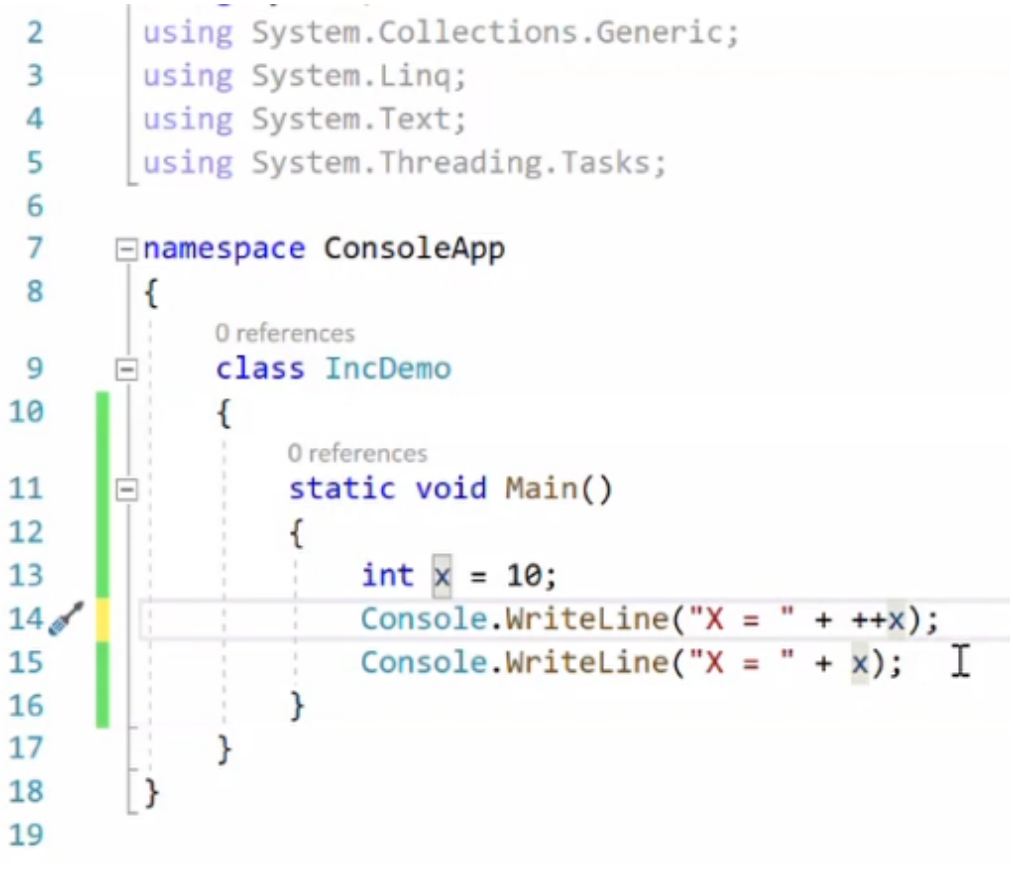
You should be little carefull about incrementing and decrementing

```
namespace ConsoleApp
{
    0 references
    class IncDemo
    {
        0 references
        static void Main()
        {
            int x = 10;
            Console.WriteLine("X = " + x++);
            Console.WriteLine("X = " + x);
        }
    }
}
```

What is the output of this code



because here ++ is a post increment so after printing it will increment



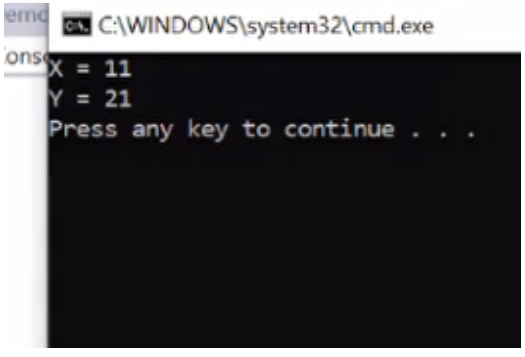
it will print me 11 11 because it is a pre increment

```
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace ConsoleApp
8   {
        0 references
9       class IncDemo
10      {
            0 references
11          static void Main()
12          {
13              int y = 10;
14              int x = 10;
15              y = y++ + ++x;
16
17              Console.WriteLine("X = " + x);
18              Console.WriteLine("Y = " + y);
19          }
20      }
```

No issues found

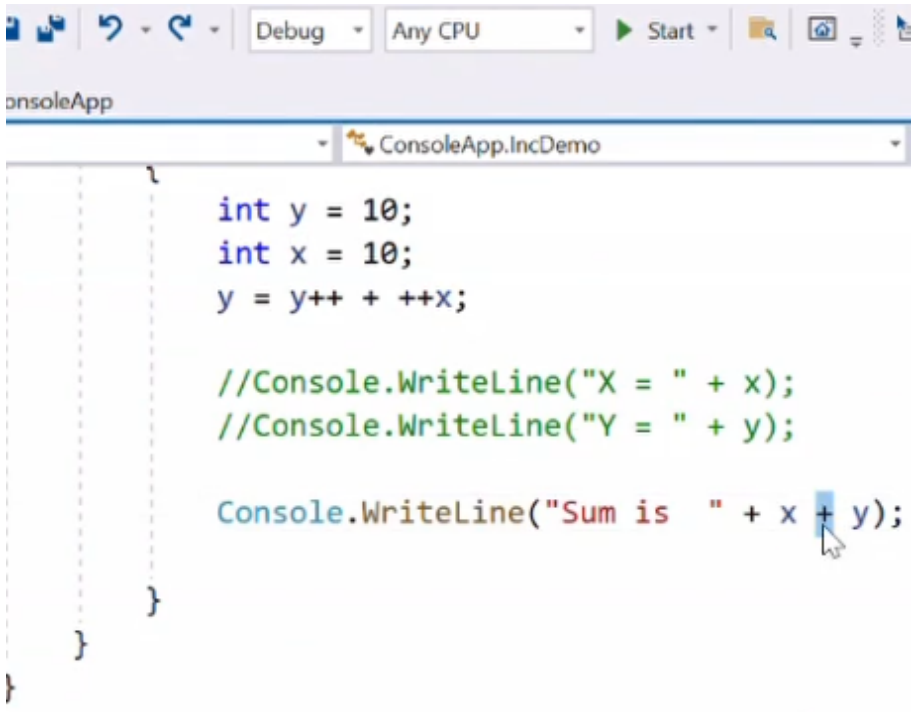
ackage Manager Console Error List Immediate Window

here what will be the output

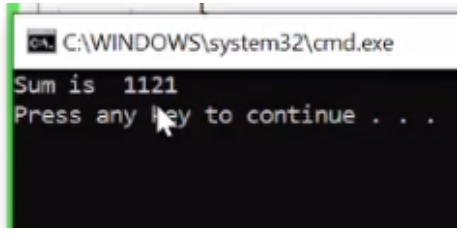


Operator Precedence

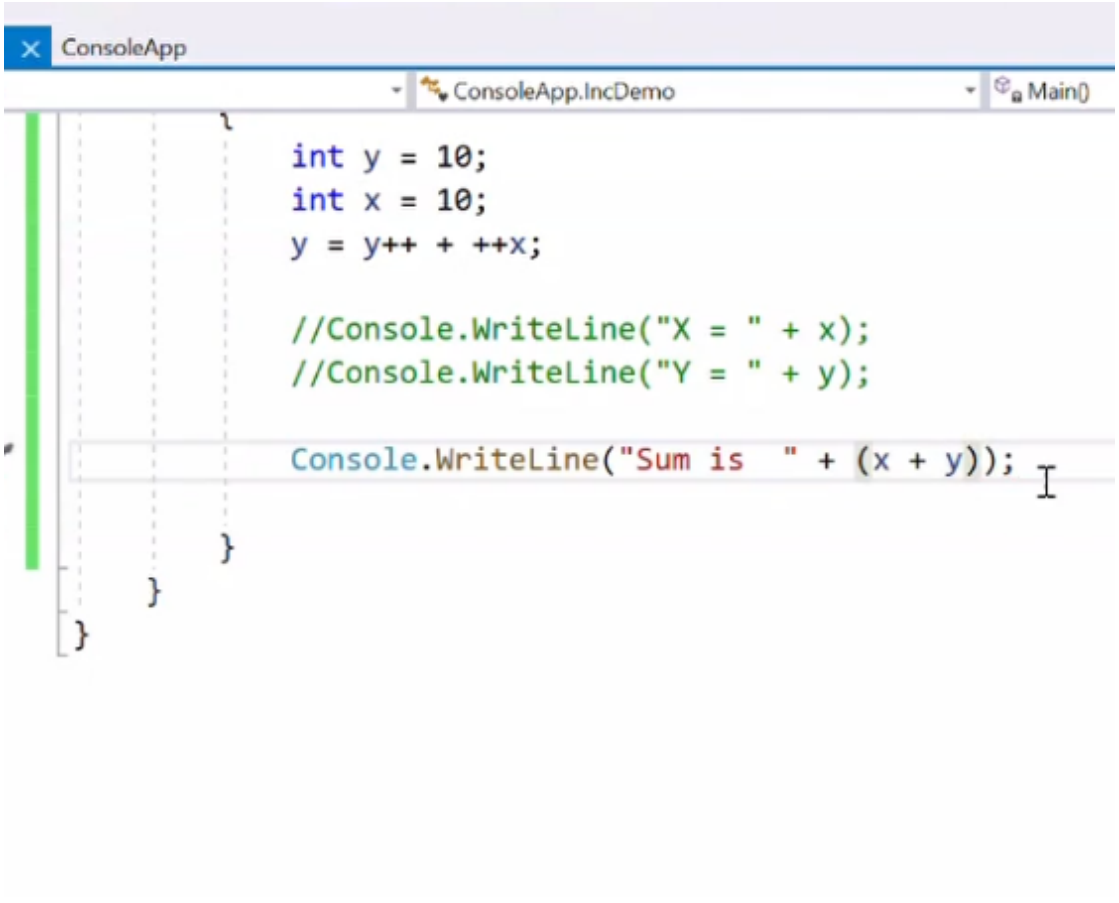
- Operator Precedence and Associatively
  - Except for assignment operators, all binary operators are left-associative
  - Assignment operators and conditional operators are right-associative

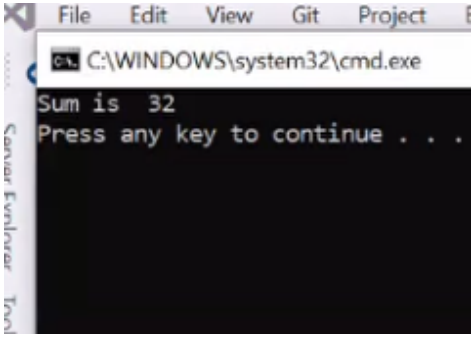


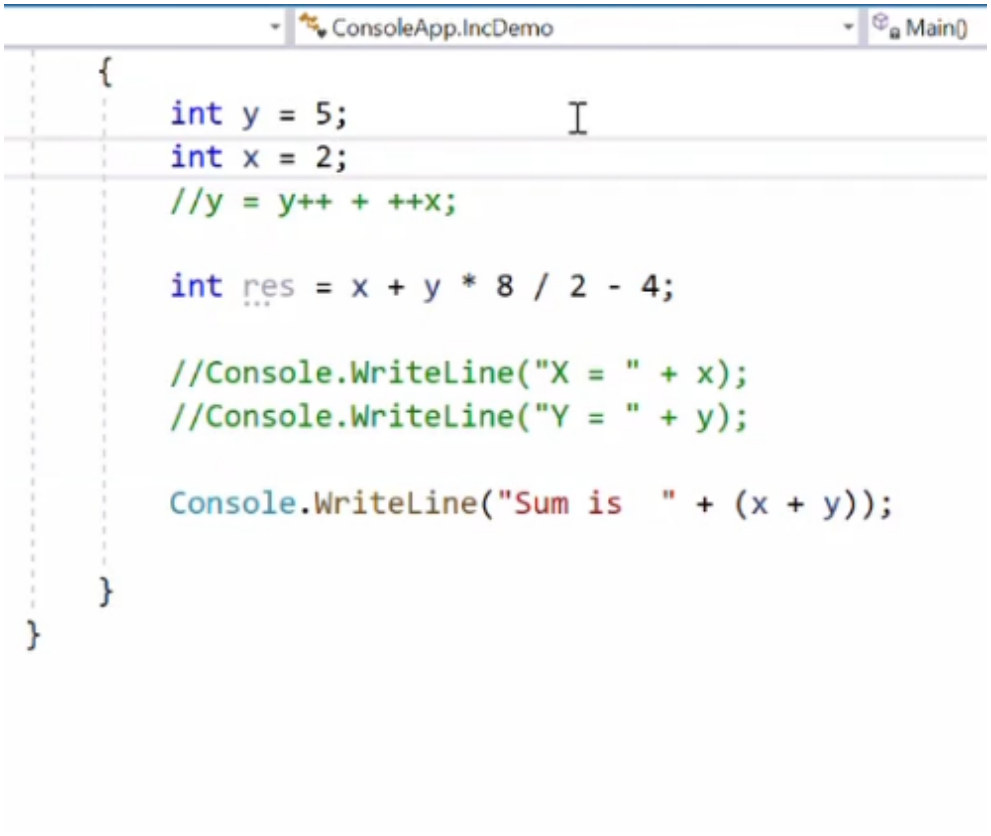
here it will concatinate it so output is



but i want to say baba add this baba don't concatinate do i'll put them in brackets

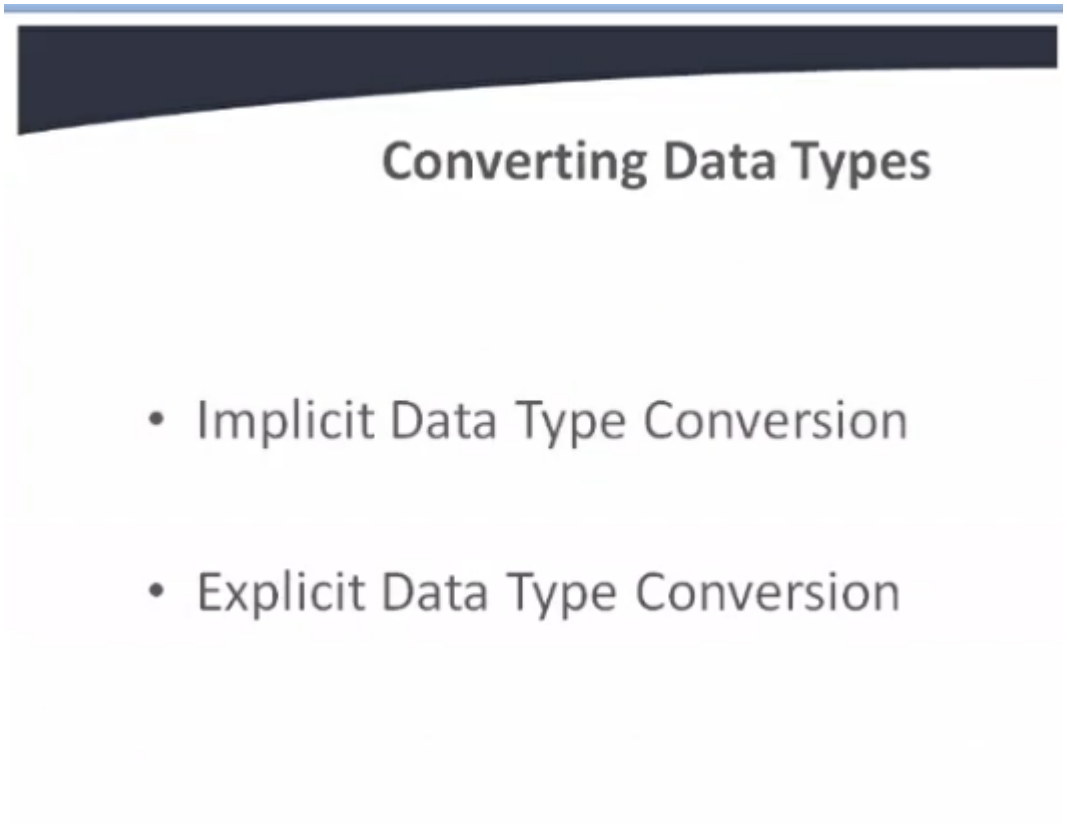

ConsoleApp
ConsoleApp.IncDemo
Main()
int y = 10;
int x = 10;
y = y++ + ++x;
//Console.WriteLine("X = " + x);
//Console.WriteLine("Y = " + y);
Console.WriteLine("Sum is " + (x + y));


File
Edit
View
Git
Project
C:\WINDOWS\system32\cmd.exe
Sum is 32
Press any key to continue . . .

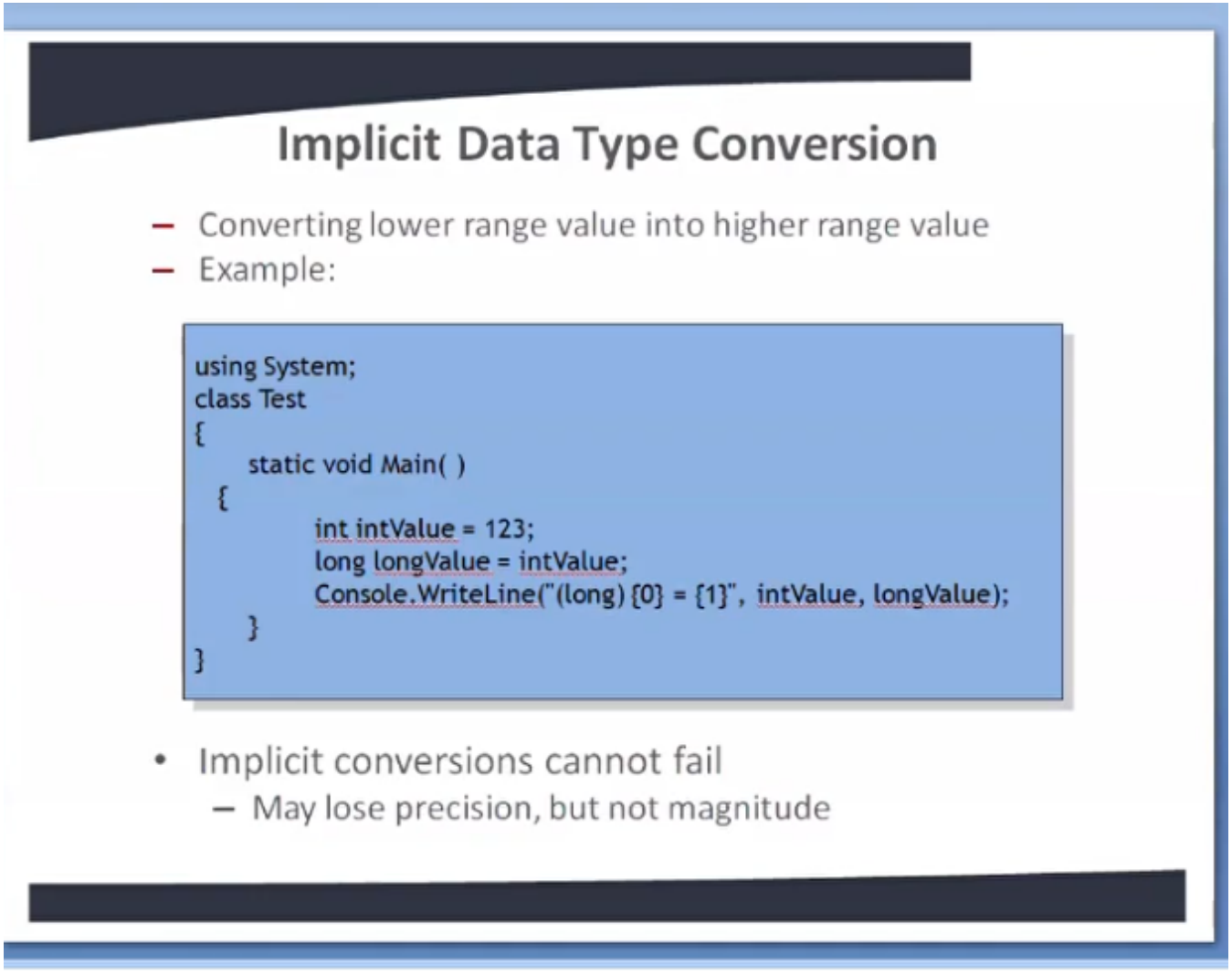


now what is the output

so it will always give preferences



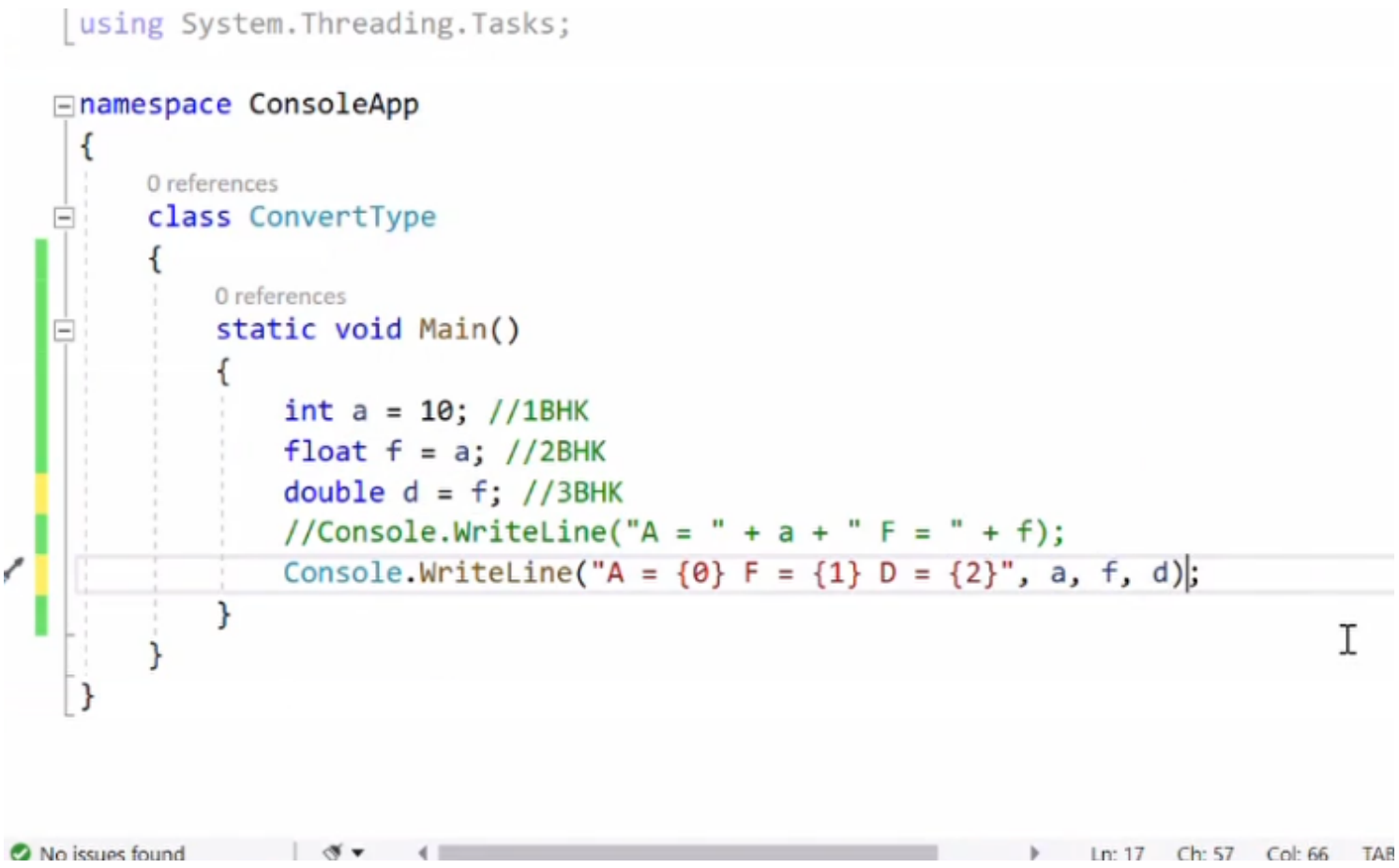
Remember when ever you talk about conversion it should be of the same type , you cannot convert from one type to other type



i can dump every thing from 1bhk to 2 bhk but i can't dump every thing from 2bhk to 1bhk

```
namespace ConsoleApp
{
    0 references
    class ConvertType
    {
        0 references
        static void Main()
        {
            int a = 10; //1BHK
            float f = a; //2BHK
            Console.WriteLine("A = " + a + " F = " + f);
            Console.WriteLine("A = {0} F = {1}", a, f);
        }
    }
}
```

either you can use this syntax
or you can write like this



i can pass float value to double

this is called implicit conversion i am going from lower to heigher where the conversions are there but it is converting from integer to float type but it is implicit

now i am taking 3BHK content to 2 BHK can we do it directly ---------**No**

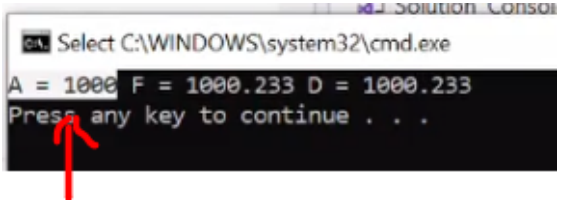
So, i need to filter i need to throw some of the unwanted things

```
namespace ConsoleApp
{
    0 references
    class ConvertType
    {
        0 references
        static void Main()
        {
            /*
            int a = 10; //1BHK
            float f = a; //2BHK
            double d = f; //3BHK
            */
            double d = 1000.233;
            float f = (float)d;

            //Console.WriteLine("A = " + a + " F = " + f);
            Console.WriteLine("A = {0} F = {1} D = {2}", a, f, d);
        }
```

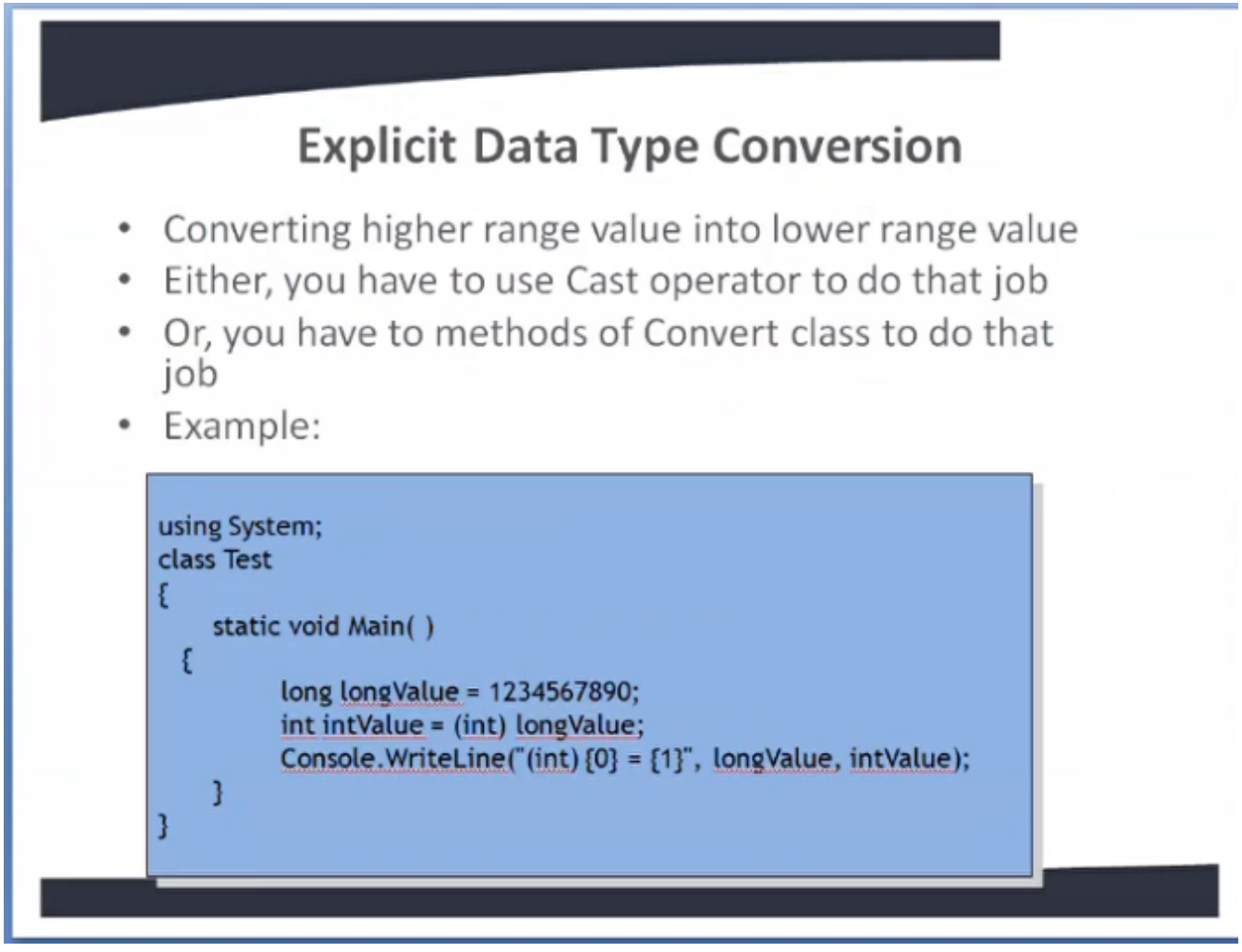
here i am throwing some of the unwanted things

Now because of some reason i have to move from 2BHK to 1 BHK now i cannot directly move from 2bhk to 1bhk so i need to filter it

this is called Explicit conversion



in case of integer it is eliminating the complete decimal value



this is with the same type, in the same type we are using conversion this are called **Type Casting**

Now we are discussing **Parsing**

in parsing we are converting the whole type from one type to another type

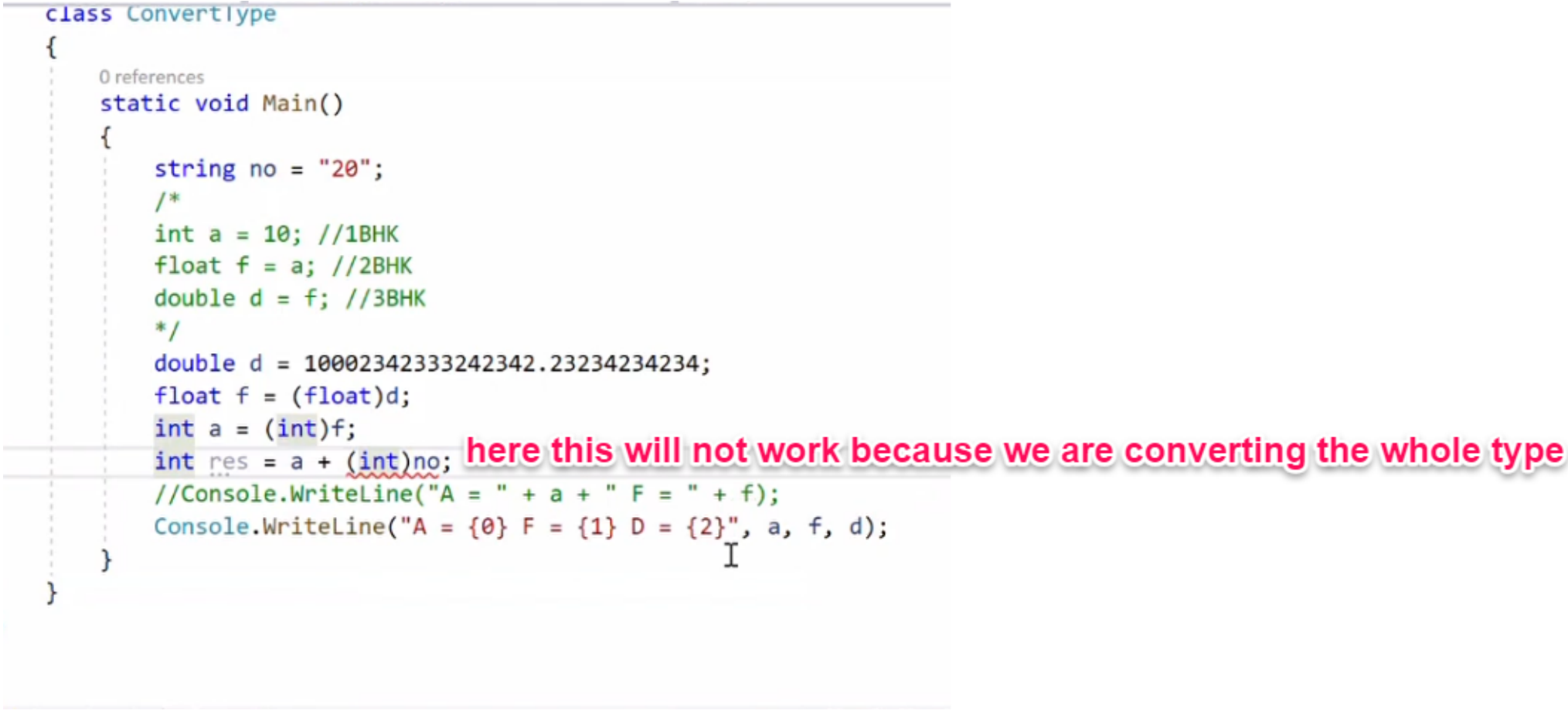
Using Convert Class

- Convert class contains methods to convert any data type into target data type
- Can be used for both Implicit and Explicit conversion
- Belongs to System namespace
- Example:

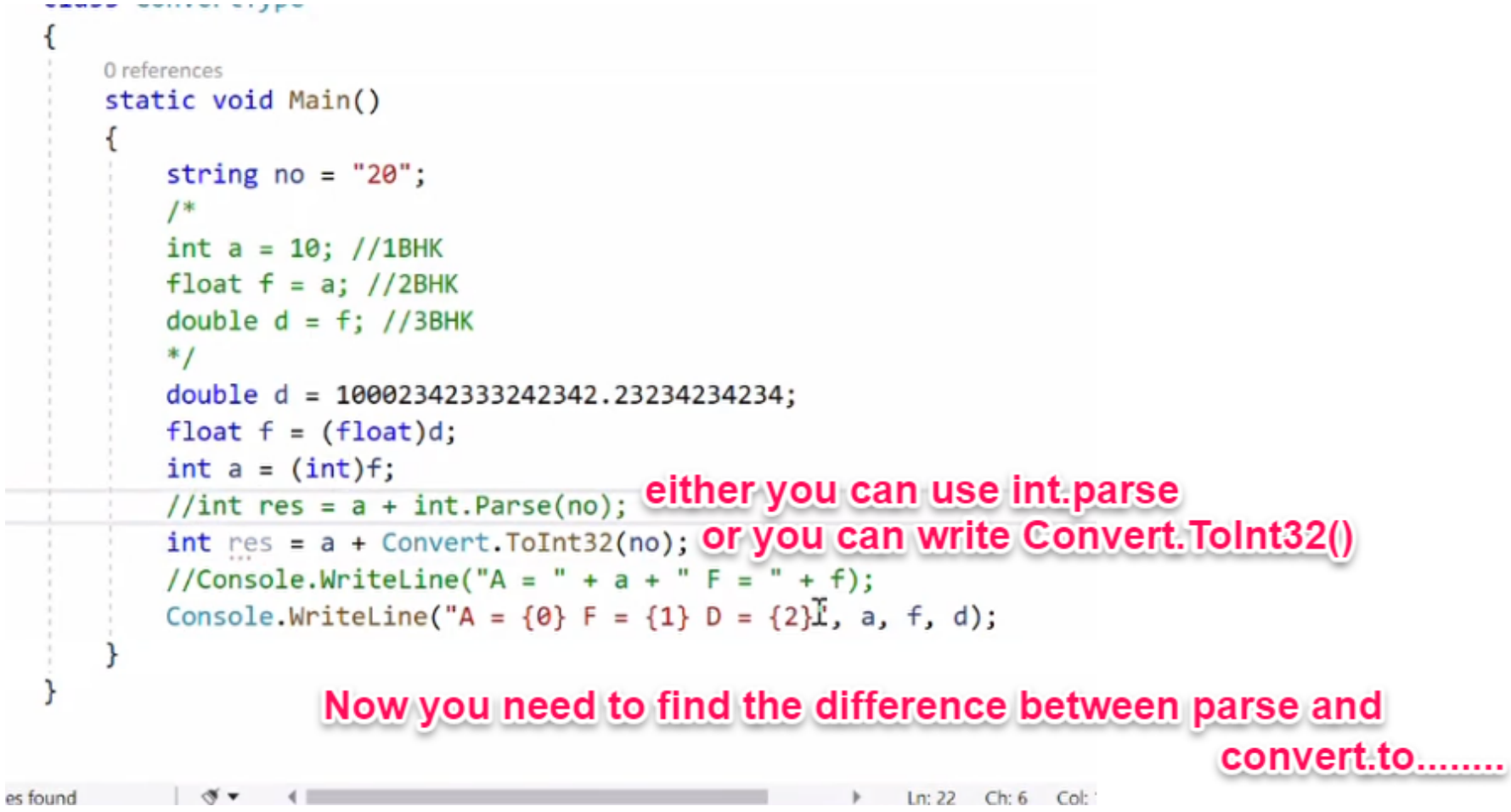
```
static void Main()
{
        Console.Write("Enter Your Identification Number: ");
        string strId = Console.ReadLine();
        int intId = convert.ToInt32(strId);
}
```

- User is asked to enter his/her id
- ReadLine() method returns id in string format
- ToInt32 method of Convert class is being used to convert that string representation into integer representation

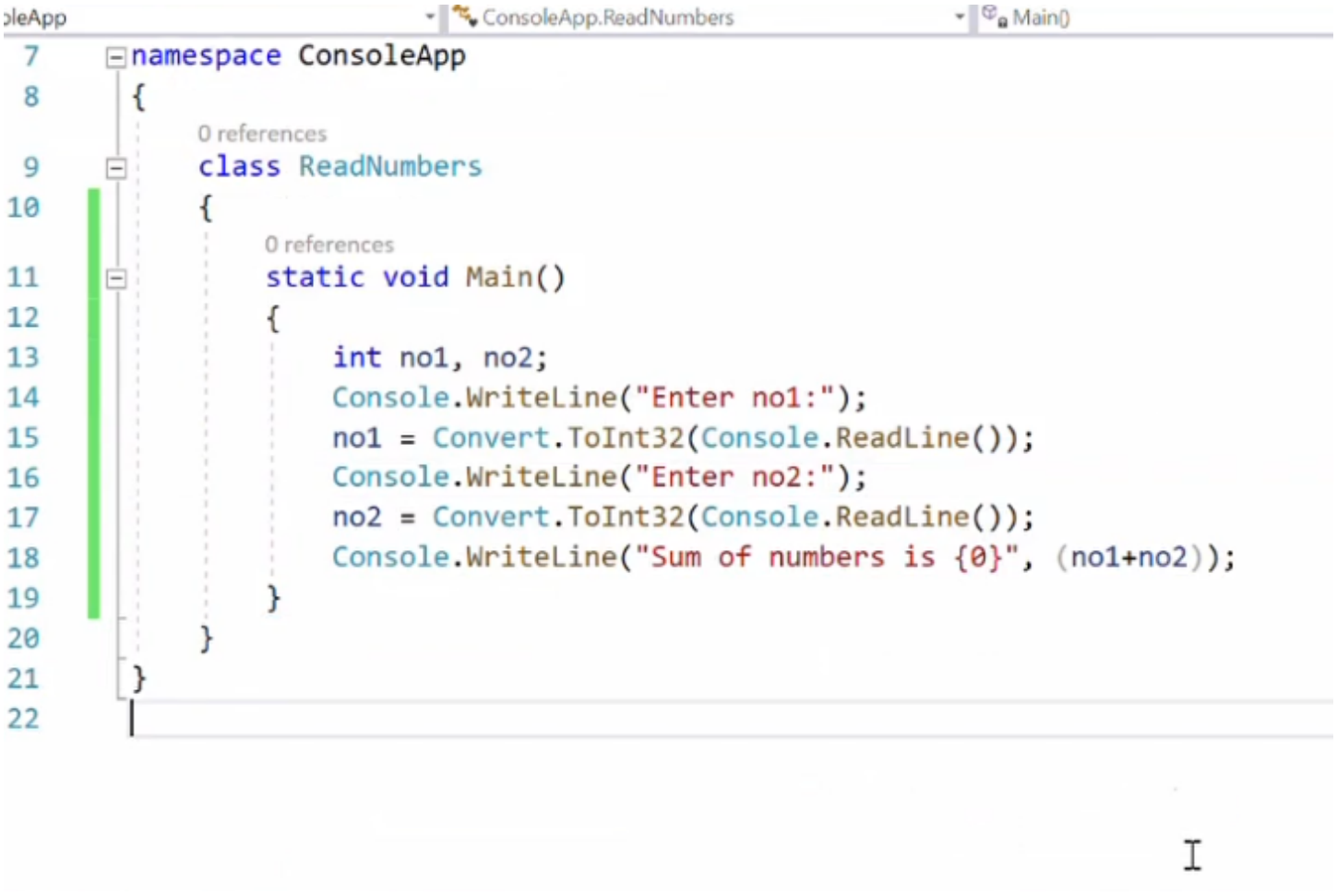
ToInt32() method converts any data type into 'int' data type.
Similarly, all other methods like, ToDecimal(), ToDouble() etc are used to convert any data type into 'decimal', 'double' data type respectively

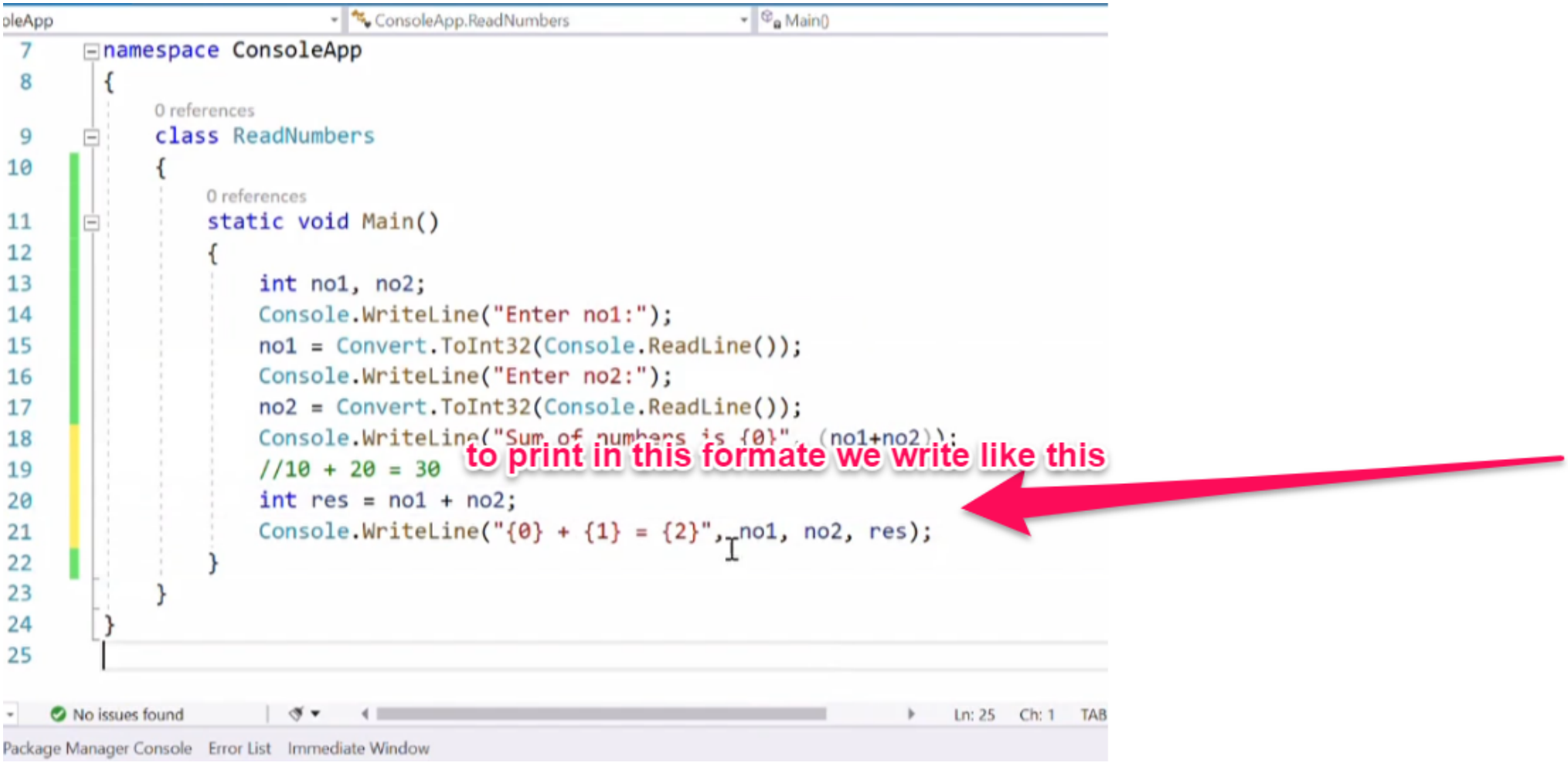


So if you want to do something like this we'll be using parsing here



What ever we take from Console.WriteLine() it always take input as a string


leApp
ConsoleApp.ReadNumbers
Main()
namespace ConsoleApp
{
0 references
class ReadNumbers
{
0 references
static void Main()
{
int no1, no2;
Console.WriteLine("Enter no1:");
no1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter no2:");
no2 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Sum of numbers is {0}", (no1+no2));
}
}
}


ConsoleApp.ReadNumbers
Main()
namespace ConsoleApp
{
0 references
class ReadNumbers
{
0 references
static void Main()
{
int no1, no2;
Console.WriteLine("Enter no1:");
no1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter no2:");
no2 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Sum of numbers is {0}", (no1+no2));
//10 + 20 = 30
int res = no1 + no2;
Console.WriteLine("{0} + {1} = {2}", no1, no2, res);
}
}
}
to print in this formate we write like this
No issues found
Ln: 25
Ch: 1
TAB
Package Manager Console
Error List
Immediate Window

Namespaces

- To provide conceptual grouping of your classes, we can provide namespaces.
- Similar to namespaces of C++.
- They can span across multiple files.
- Typically we use namespaces
  - To group our classes as per our logical groupings
  - To avoid naming conflicts.
- Units defined under namespaces are to be referred by namespace.UnitName.
- Example:
  - *System;*
  - *System.Collections;*
  - *System.IO;*

Statements in C#

Category	C# keywords
Selection statements	if, else, switch, case
Iteration statements	do, for, foreach, in, while
Jump statements	break, continue, default, goto, return, yield
Exception handling statements	throw, try-catch, try-finally, try-catch-finally
Checked and unchecked	checked, unchecked
fixed Statement	fixed
lock Statement	lock

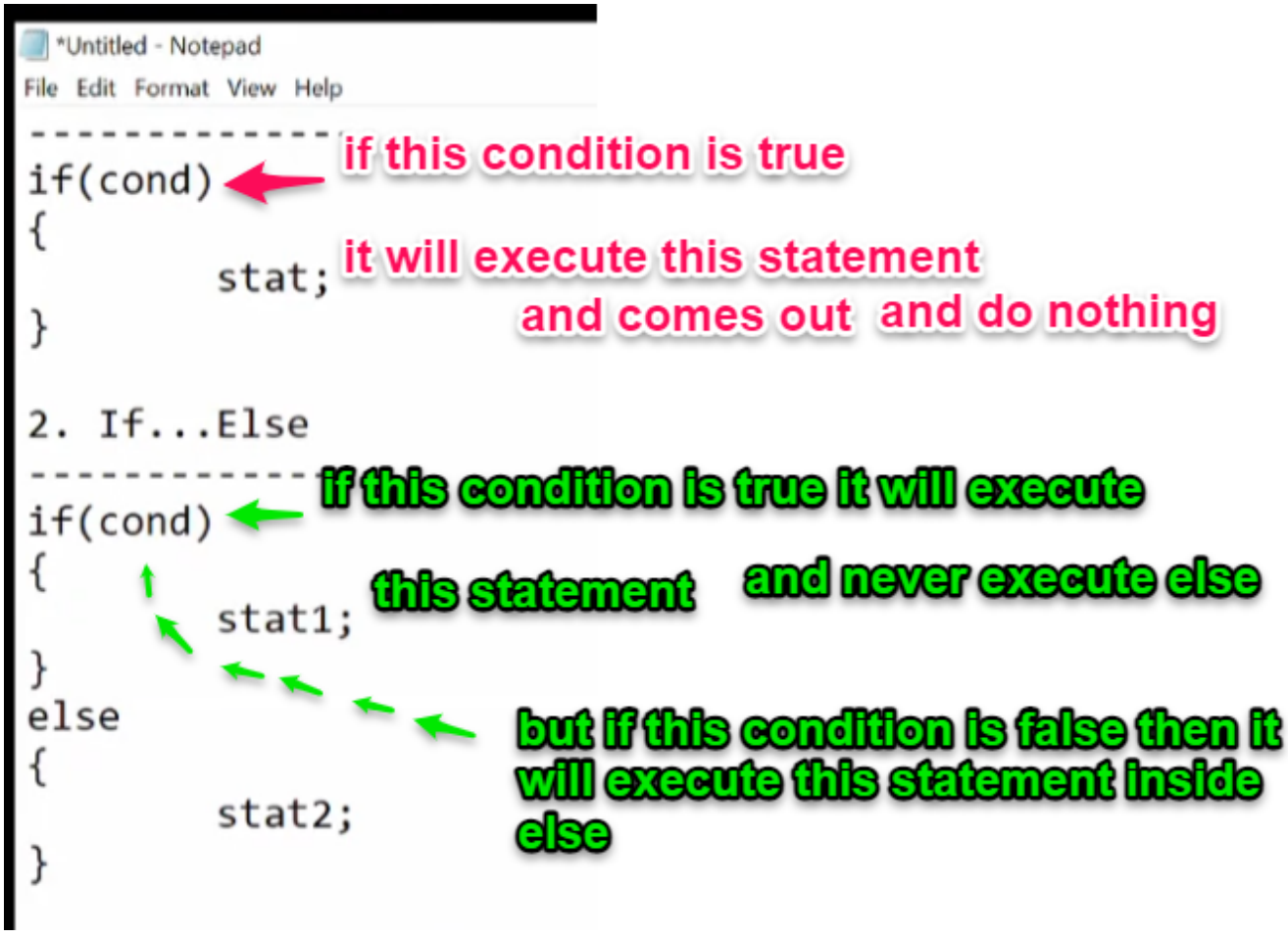
If..Else Statement

- Syntax:

```
if ( Boolean-expression )
  first-embedded-statement
else
  second-embedded-statement
```

- No implicit conversion from int to bool

```
int x;
...
if (x) ...          // Must be if (x != 0) in C#
if (x = 0) ...      // Must be if (x == 0) in C#
```


*Untitled - Notepad
File Edit Format View Help
if(cond)
{
stat;
}
if this condition is true
it will execute this statement
and comes out and do nothing
2. If...Else
if(cond)
{
stat1;
}
else
{
stat2;
}
if this condition is true it will execute
this statement
and never execute else
but if this condition is false then it will execute this statement inside else

3. if...elseif...else

```
if(cond1)
{
        stat1;
}
else if(cond2)
{
        stat2;
}
else
{
        statn;
}
```

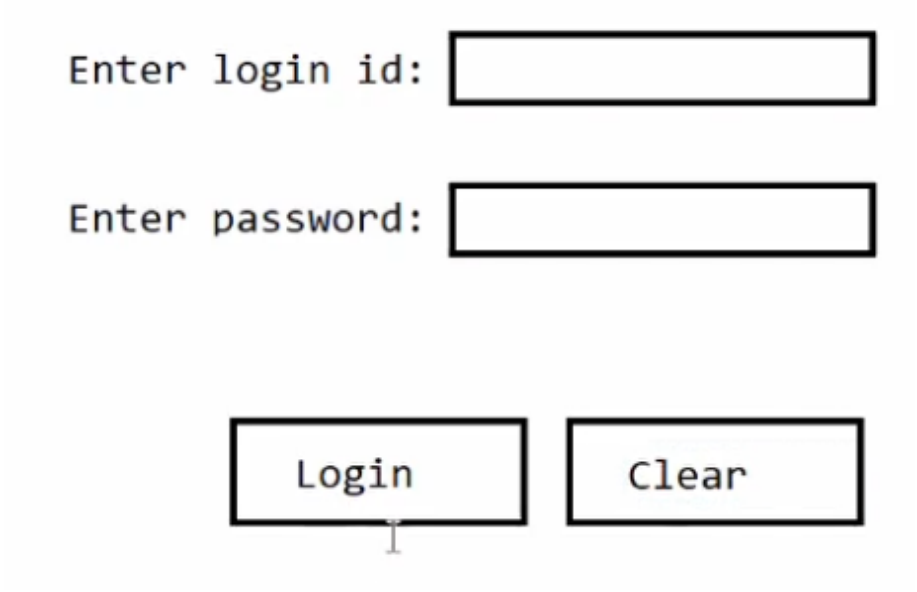
if this condition is true it executes this statements and never executes this elseif and else statements

but if condition1 is false then it will check for condition2 if this condition is true then it execute this statement and never execute statement inside else

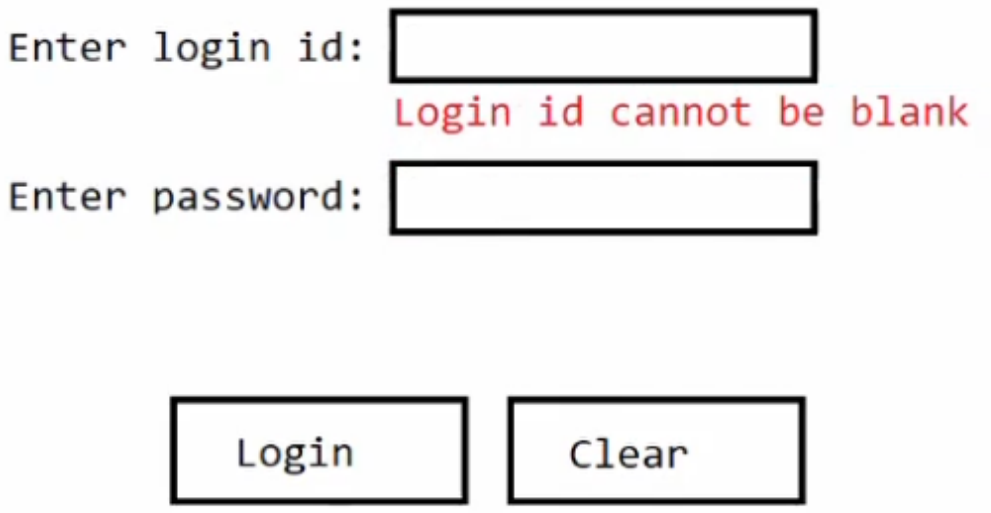
but if this 2 conditions are false then the condition inside else will be executed

```
4. Neseted if
--------------
if(cond1)
{
        if(cond1.1)
        {
                stat1.1;
        }
        else
        {
                stat1.2;
        }
}
else
{
        statn;
}
```

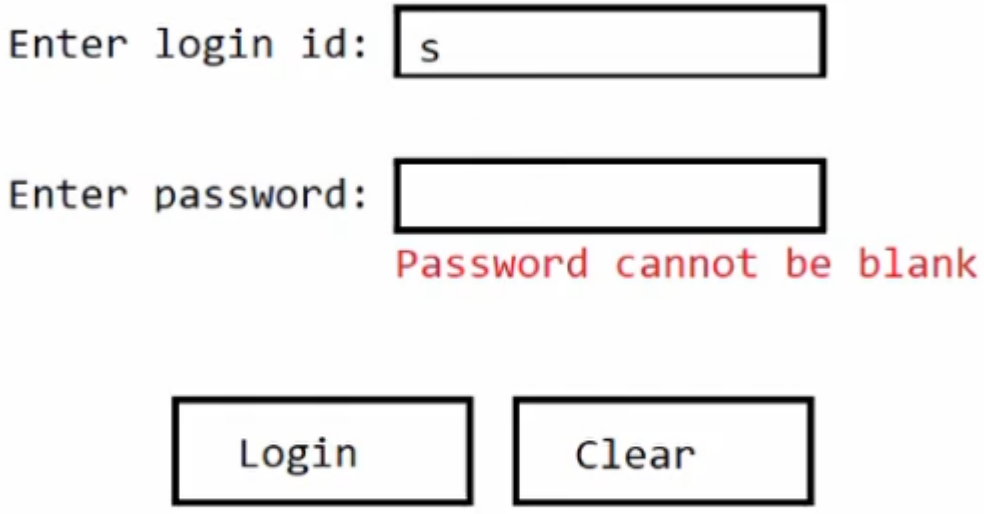
when ever you are using nested if
then here cond1 should have association with
condition1.1



When user clicks on this login if Loginid or password is blank i should get a message



if user enter something in login id but pass is blank so i should get err msg like this



So to achieve this which if should i use senario like this dosen't need nested if So, my simplest solution is

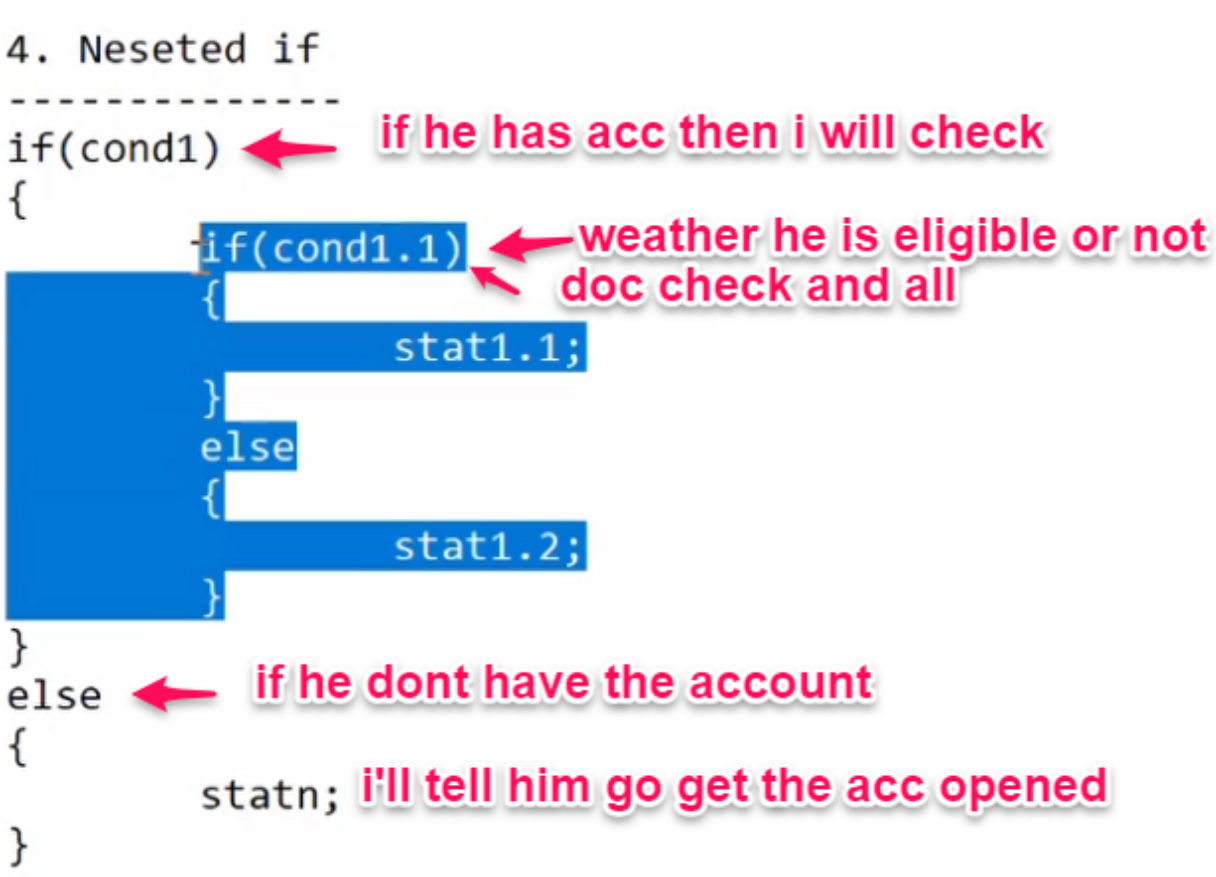
```
if(login == '')
{
        print("Login cannot be blank");
}
if(password == '')
{
        print("Password cannot be blank");
}
```

I Have a 3rd senario we have a banking application which you are developing in that we have a loan section and the form is been pass to the manager, and me being a manager i have to approve a loan so

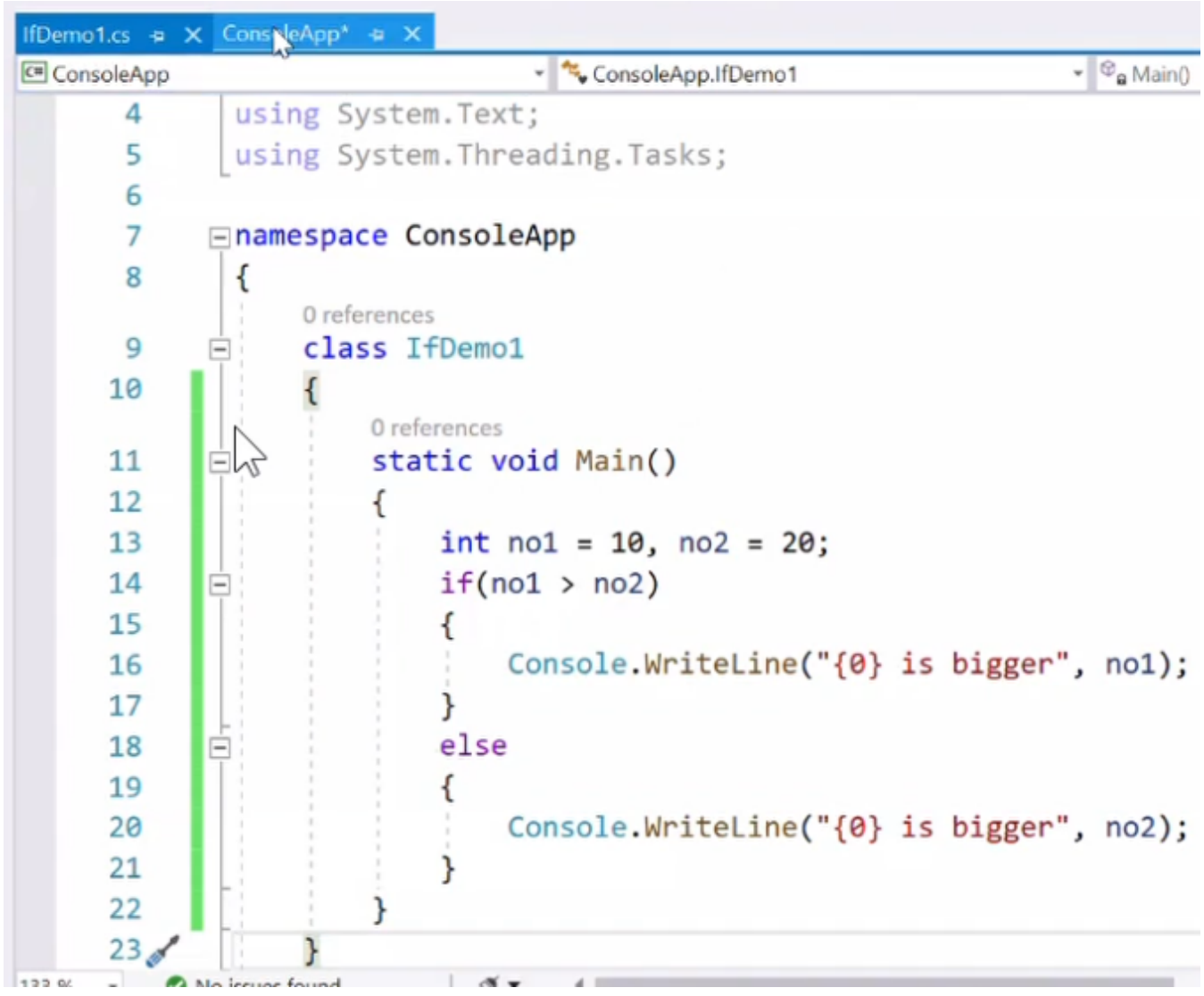
1. the person should have acc open in our bank if not we'll say first open the acc then come back then we'll process it

but if the person already has the acc then we will check for other document
So, How do you controll this

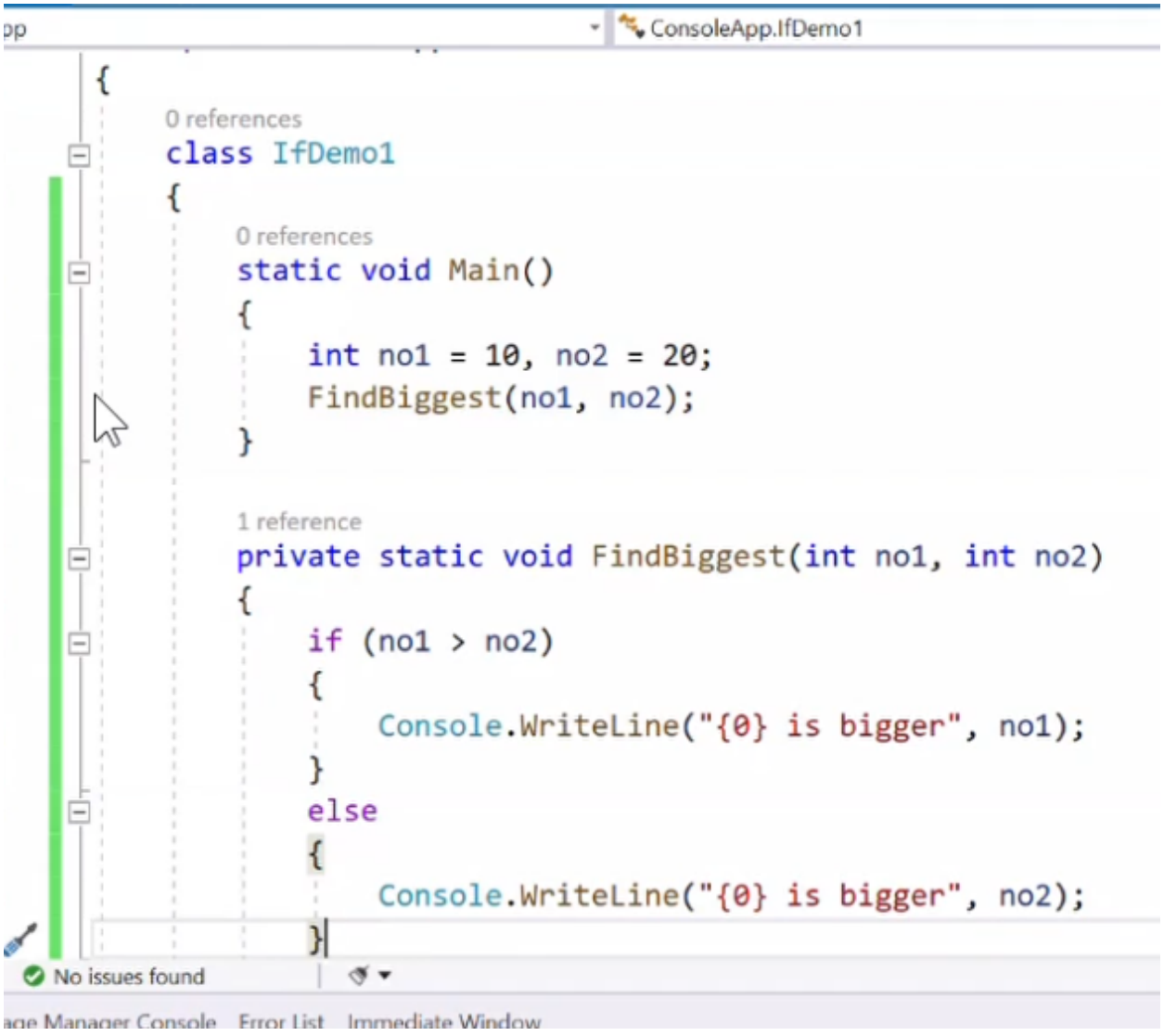
Here we use nested if because the 2nd condition is dependent on 1st condition



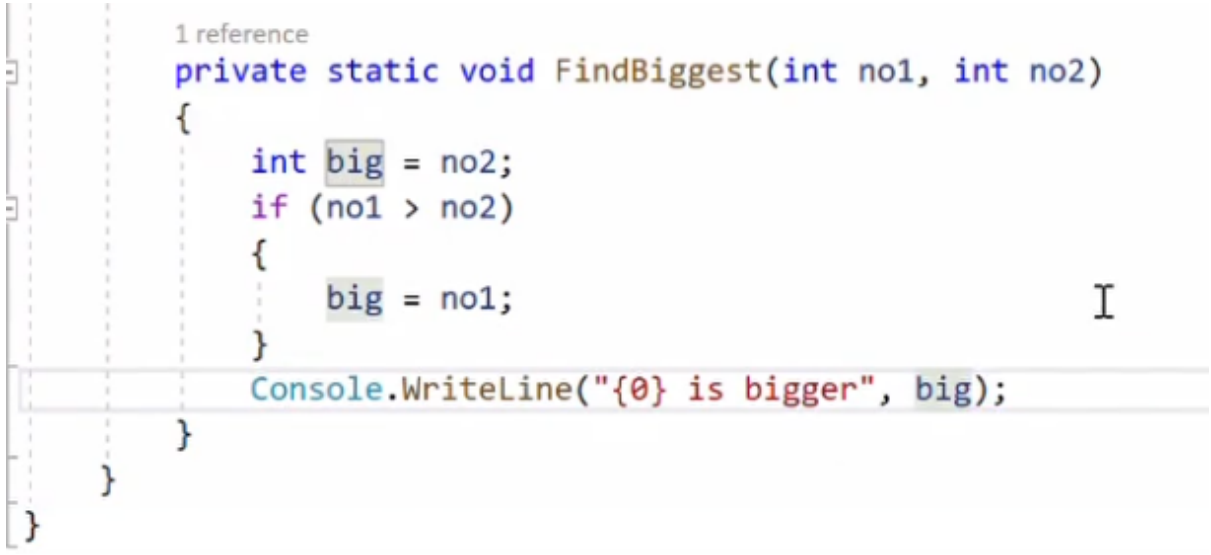
biggest of 2 number using if else



we always have to use different methods for input process and output



now can we optimize this code can we do this without else

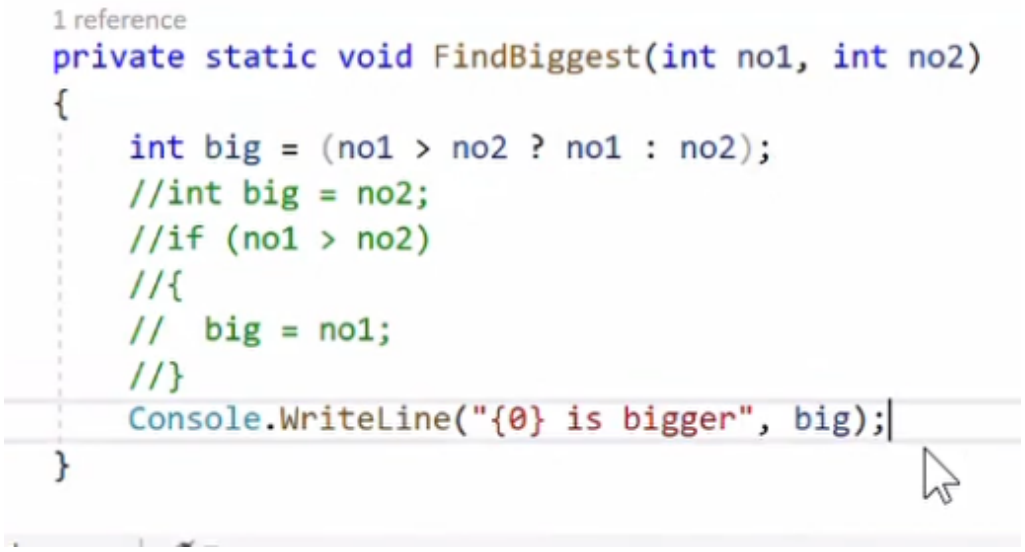


```
        1 reference
        private static void FindBiggest(int no1, int no2)
        {
            int big = no2;
            if (no1 > no2)
            {
                big = no1;
            }
            Console.WriteLine("{0} is bigger", big);
        }
    }
}
```

performance wise it is better but you can see we are taking a additional variable here

Remember if is always a costly process so try don't to use if else as much as possible it is always a dangerous process

Now if it is just a matter of finding the biggest so you can always use a tarnary operator

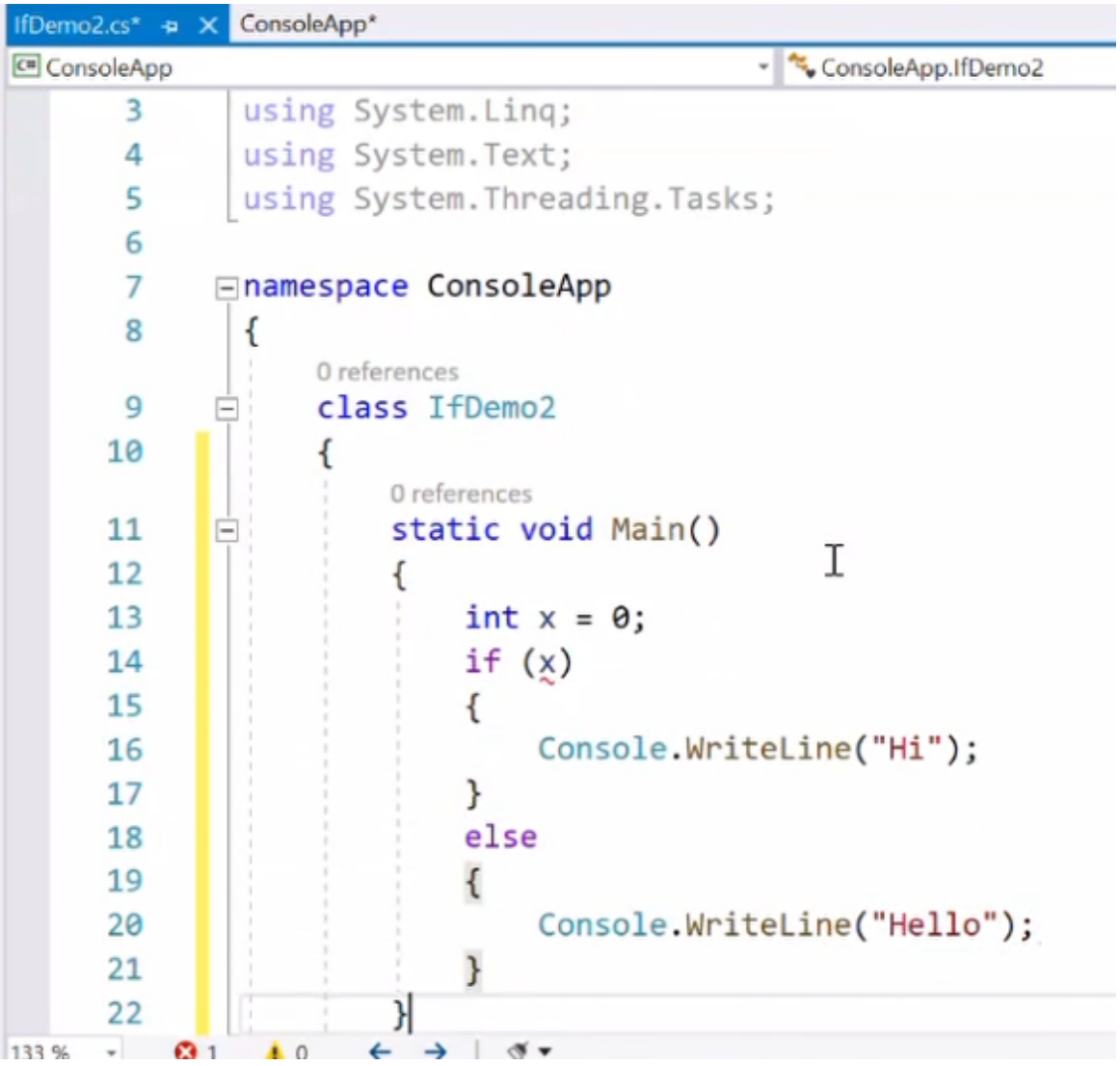


Assignment :-Find Biggest of 3 Number using turnary operator

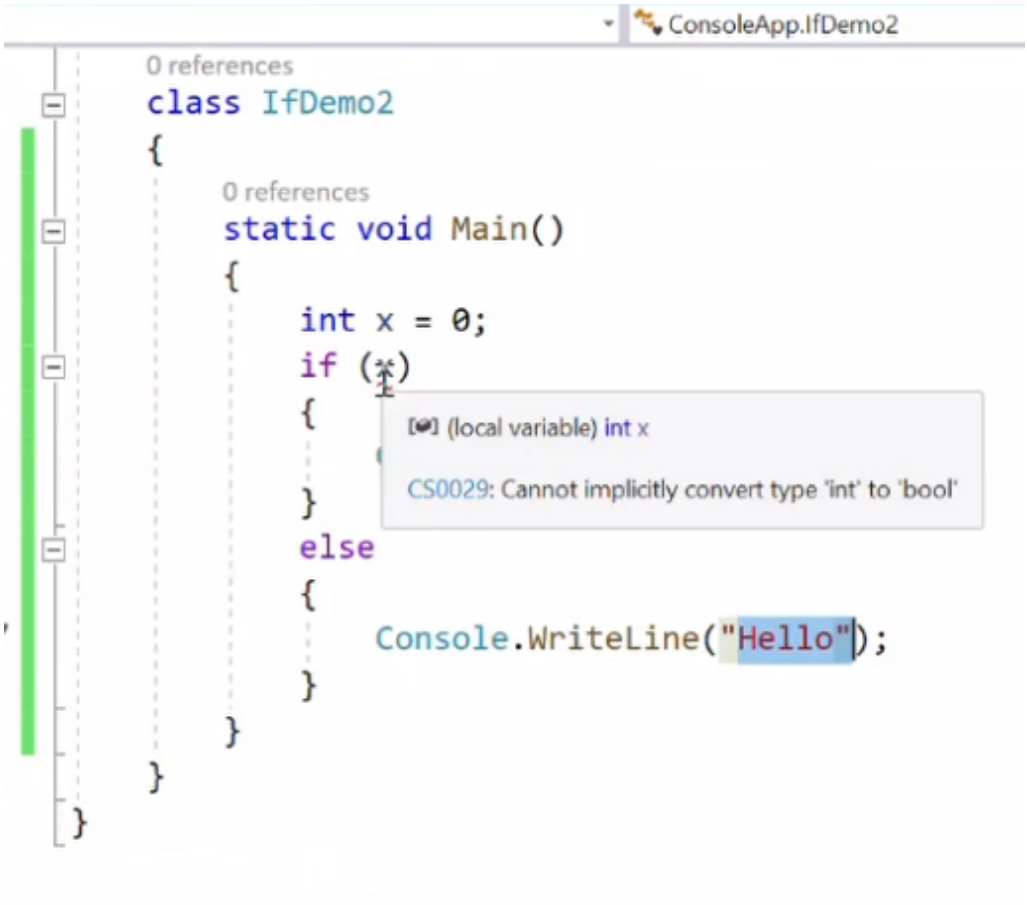
Cascading if Statements

```
Console.WriteLine("Enter Color Choice: ");
string choice = Console.ReadLine();
if (choice == "Black")
    Console.WriteLine("Choice is Black..");
else if (choice == "Blue")
    Console.WriteLine("Choice is Blue..");
else if (choice == "Red")
    Console.WriteLine("Choice is Red..");
else
    Console.WriteLine("No Color choice..");
```

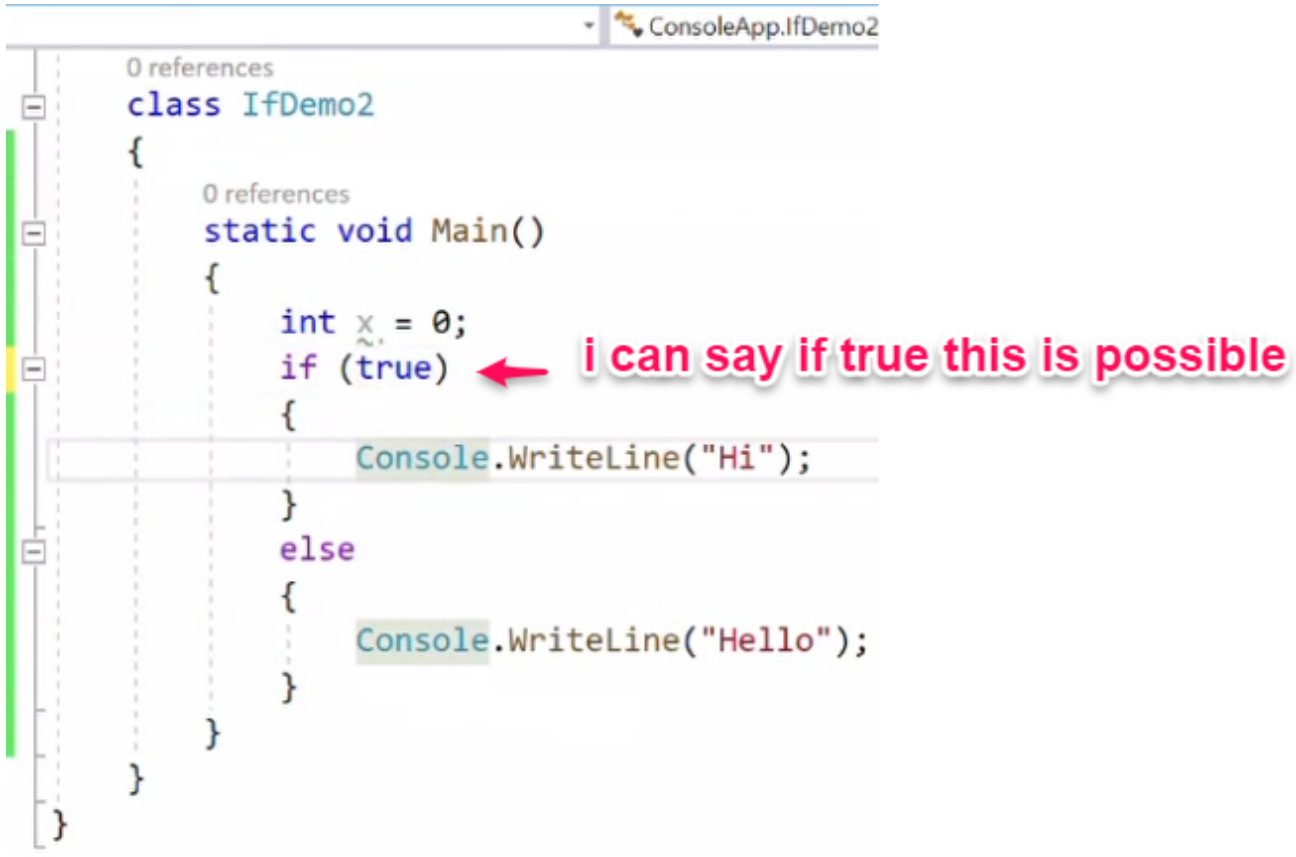
what will be the output ? will it print Hi or Hello

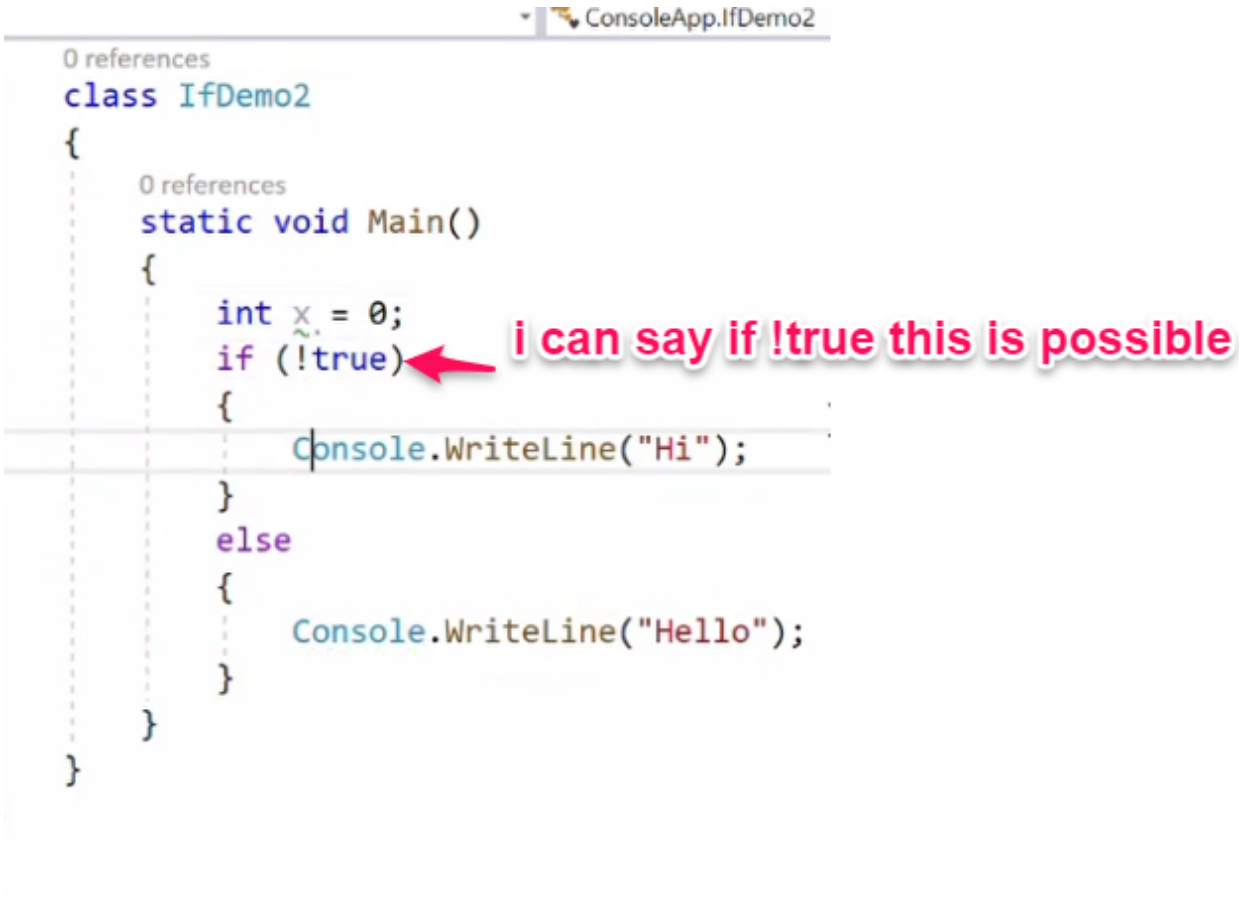


there is a compilation error

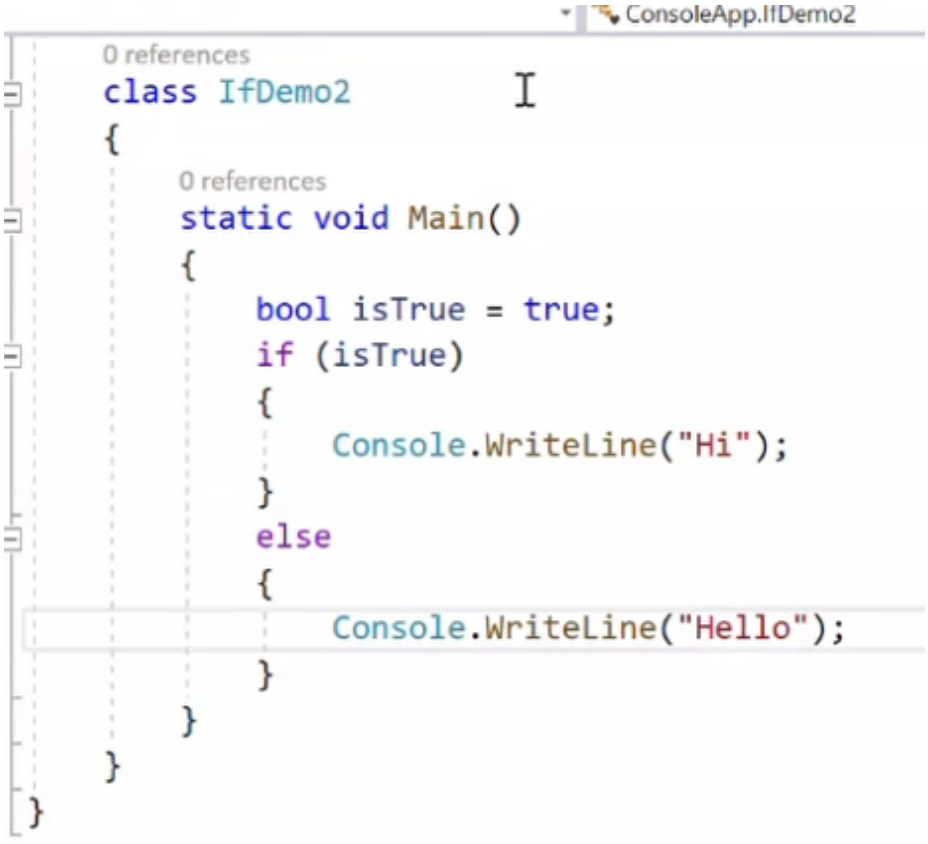


I C# or in java you cannot do this because if will always check for boolean value **in C or C++ this will work**

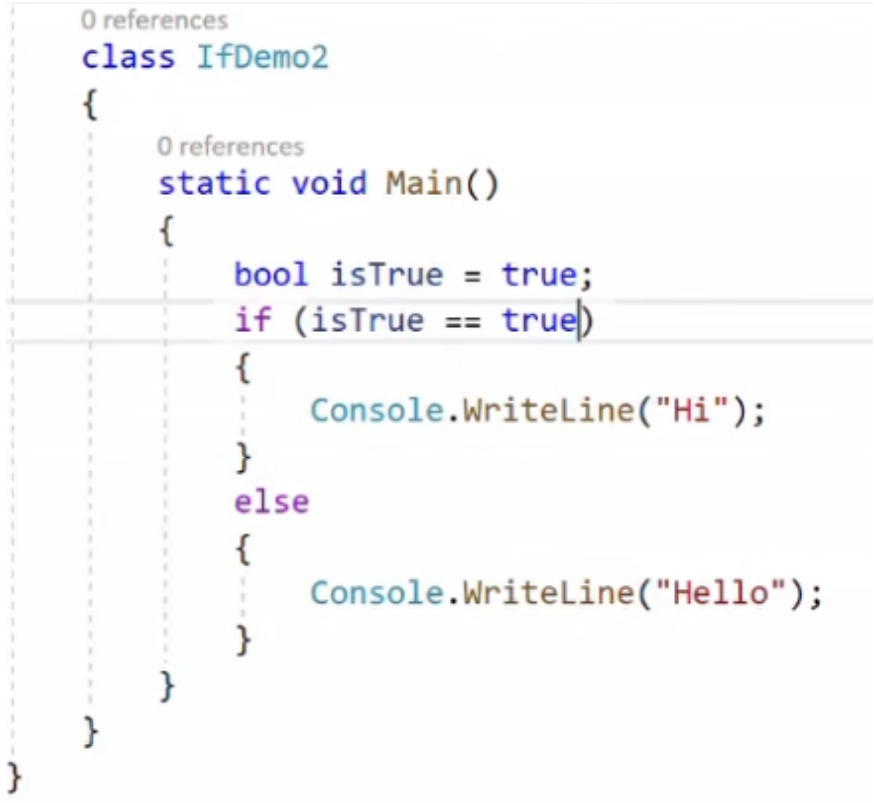

ConsoleApp.IfDemo2
0 references
class IfDemo2
{
0 references
static void Main()
{
int x = 0;
if (true)
{
Console.WriteLine("Hi");
}
else
{
Console.WriteLine("Hello");
}
}
}
i can say if true this is possible



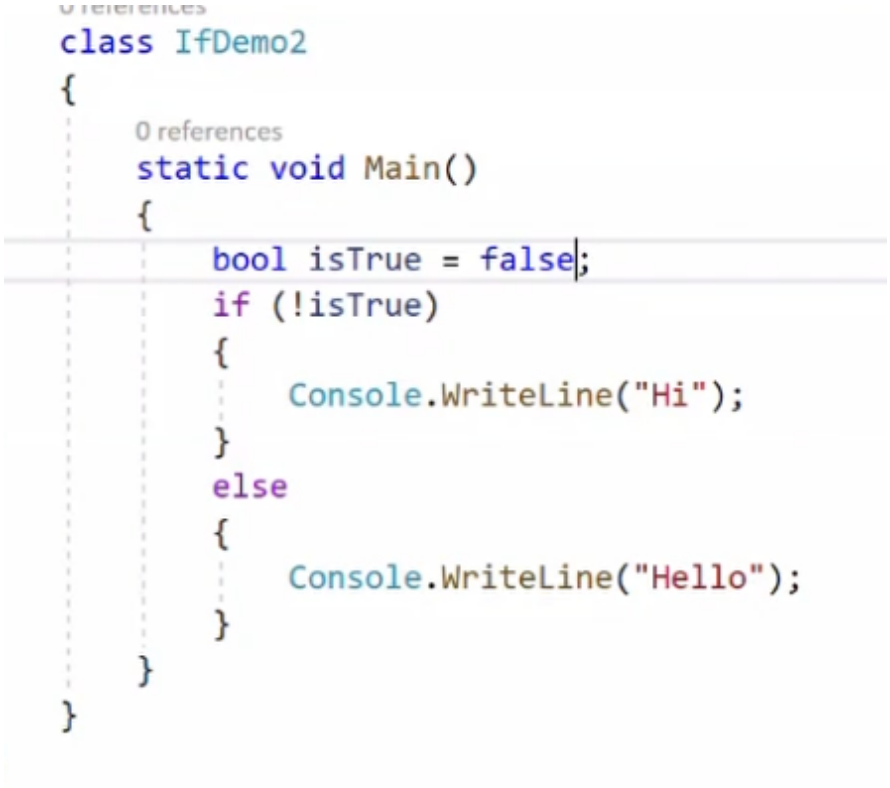
i can say if false that is also possible but i cannot say if 0 or 1 that is not possible



we can write like this



we can also write like this
but internally my compiler treat this as 1st one,
so we don't write like this
we write like the 1st one



```
class IfDemo2
{
    0 references
    static void Main()
    {
        bool isTrue = false;
        if (!isTrue)
        {
            Console.WriteLine("Hi");
        }
        else
        {
            Console.WriteLine("Hello");
        }
    }
}
```

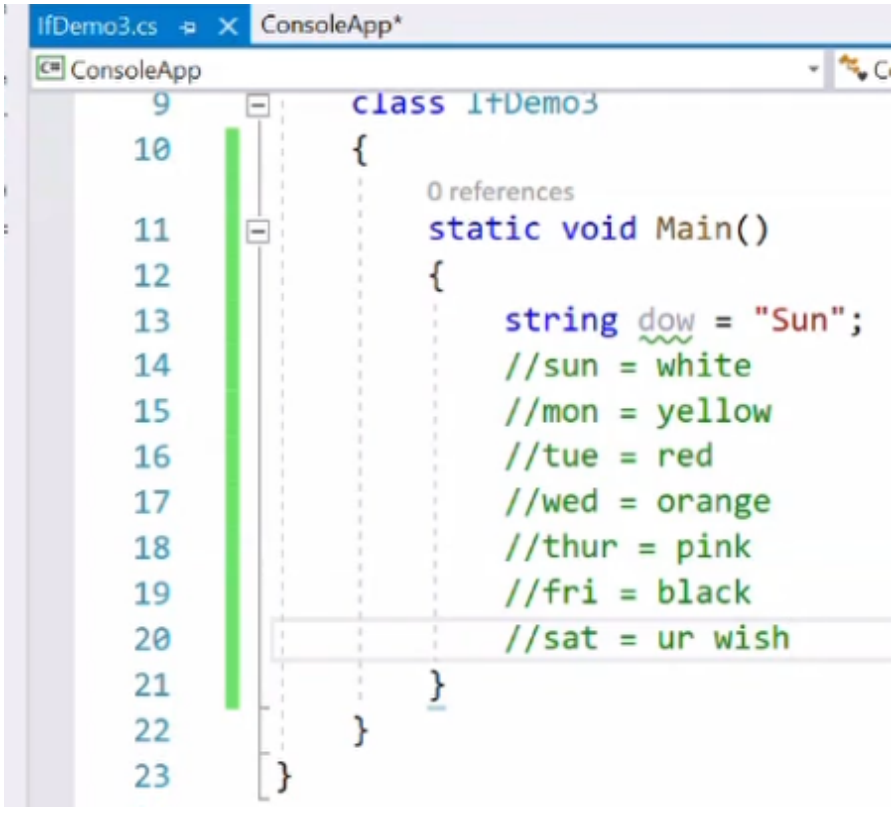
Now what will be the output ---------Hi

```
0 references
class IfDemo2
{
    0 references
    static void Main()
    {
        bool isTrue = false;
        if (!isTrue)
        {
            Console.WriteLine("Hi");
            isTrue = !true;
        }
        else
        {
            Console.WriteLine("Hello");
        }
        Console.WriteLine("Value of isTrue is {0}", isTrue);
    }
}
}
```

```
0 references
class IfDemo2
{
    0 references
    static void Main()
    {
        bool isTrue = true;
        if (isTrue)
        {
            Console.WriteLine("Hi");
        }
        else
        {
            Console.WriteLine("Hello");
        }
        isTrue = !isTrue;
        Console.WriteLine("Value of isTrue is {0}", isTrue);
    }
}
}
```

Write program for this if day == sunday then color = white , if day == mon then color =yellow

.....


IfDemo3.cs
ConsoleApp*
ConsoleApp
class IfDemo3
{
0 references
static void Main()
{
string dow = "Sun";
//sun = white
//mon = yellow
//tue = red
//wed = orange
//thur = pink
//fri = black
//sat = ur wish
}
}
}

```
{
    string dow = "Sun";
    //sun = white
    //mon = yellow
    //tue = red
    //wed = orange
    //thur = pink
    //fri = black
    //sat = ur wish
    if(dow == "Sun")
    {
        Console.WriteLine("White");
    }
    else if(dow == "Mon")
    {
        Console.WriteLine("Yellow");
    }
    else if(dow == "Tue")
    {
        Console.WriteLine("Red");
    }
}
```

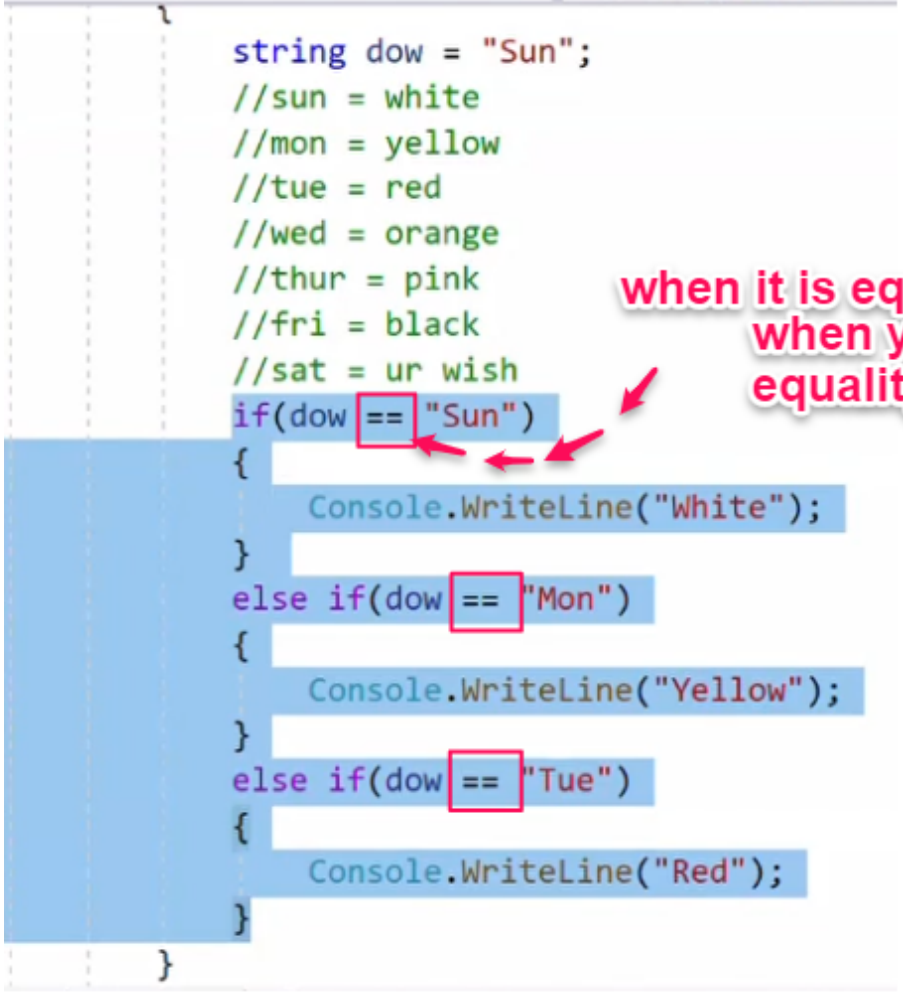
WE can keep writing like this elseif elseif .....

But when you have the same variable but with different value

it is the same you but you behave differently

when you sitting in the class u act like a student

when you being with your mom u act like a Son



when it is equal to
when you have a variable and you checking for equality

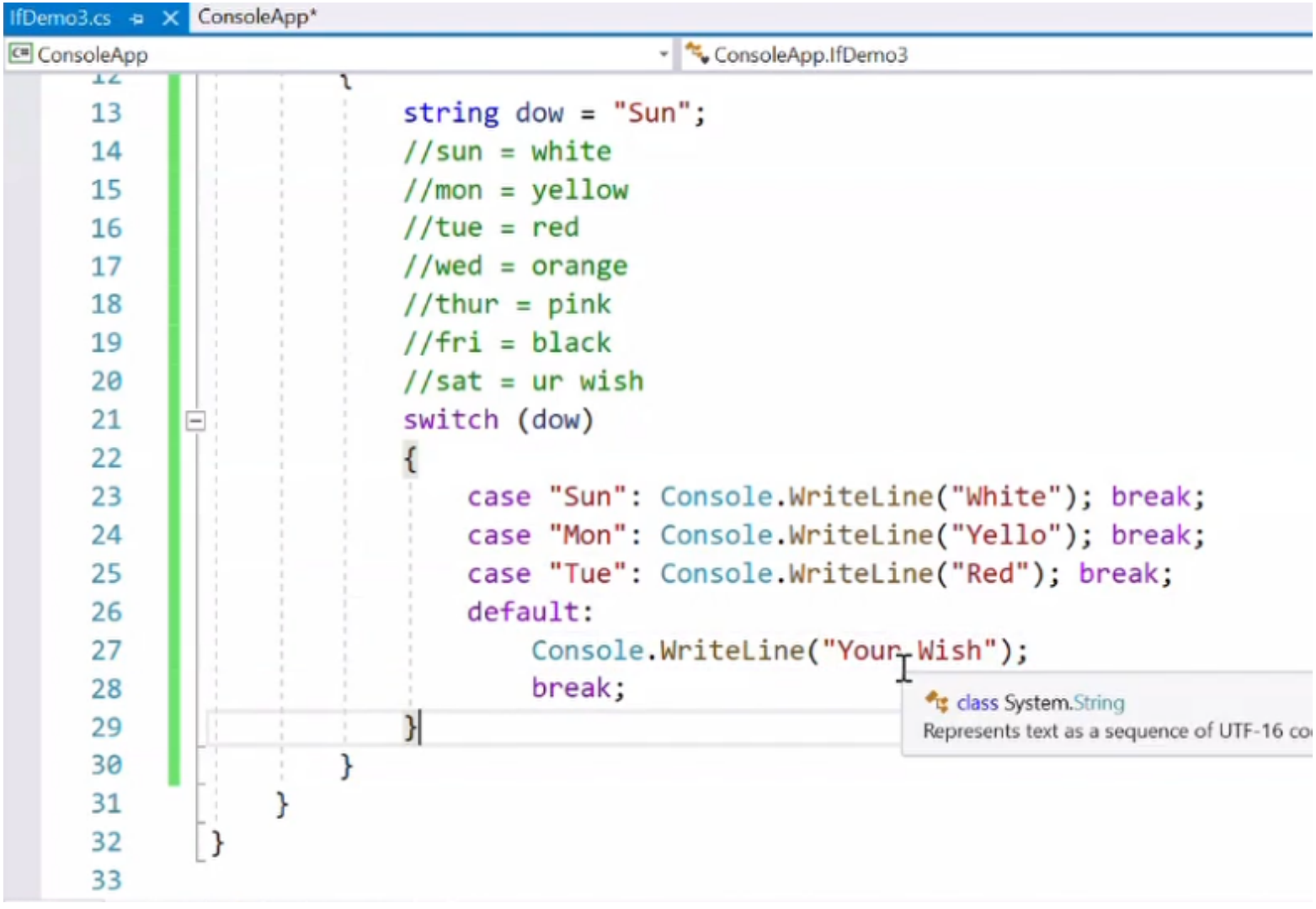
SO you should NOT use if

It is always a Performance head
it will take a lot of time
to render /Process

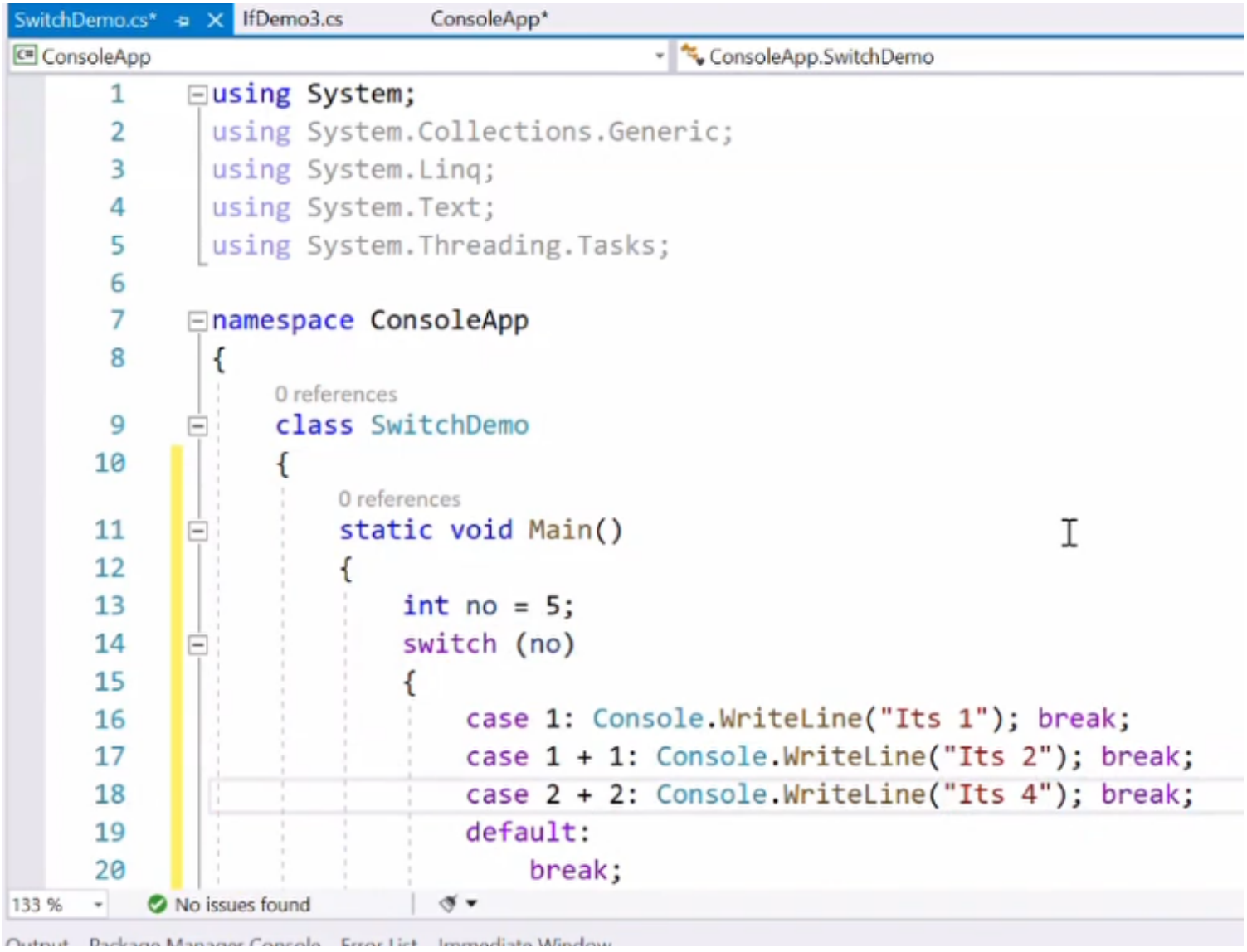
So in such senario You should always use Switch

So, Switch is used when ever you have the same variable but may have different value in this case you can always go

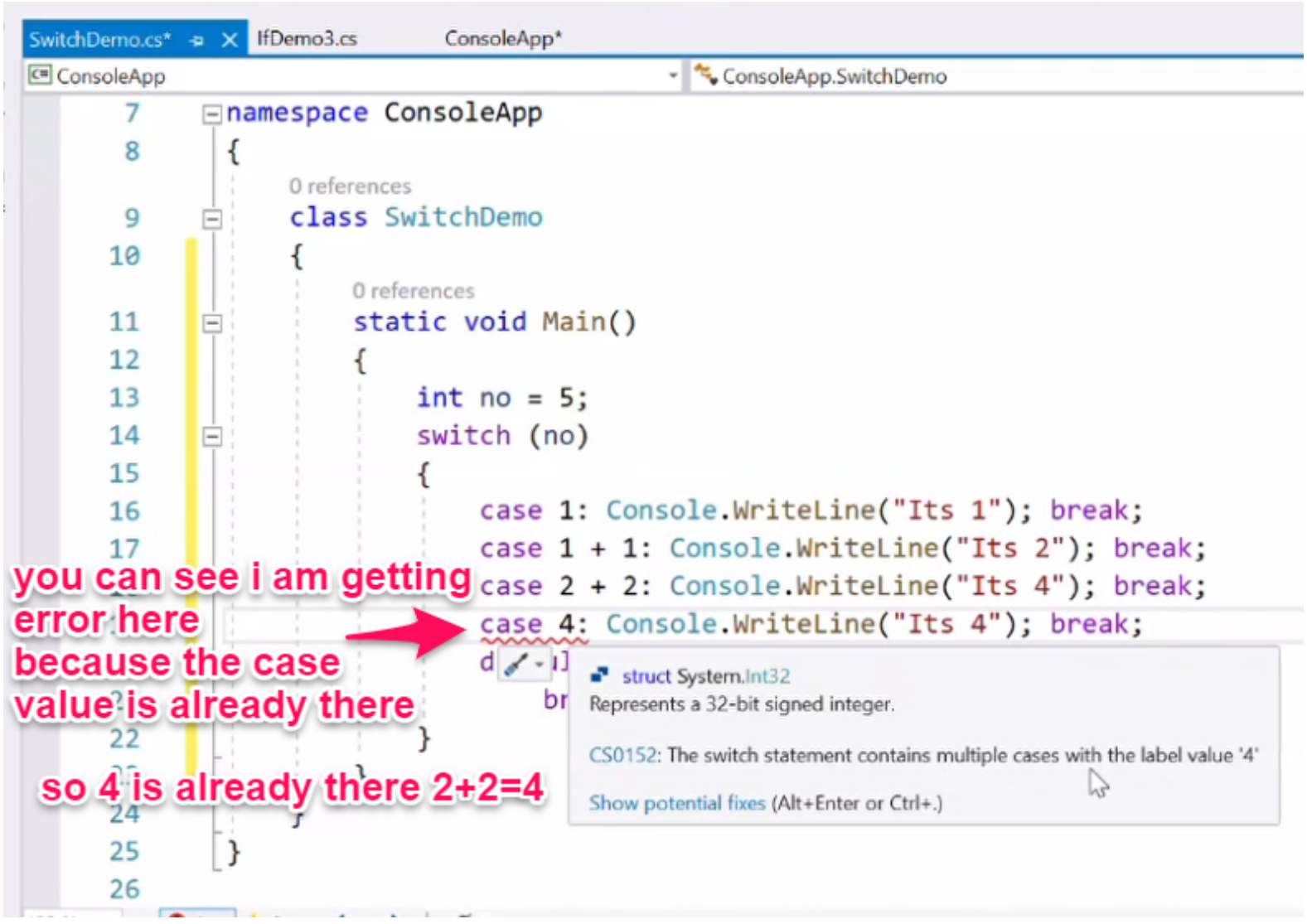
with Switch



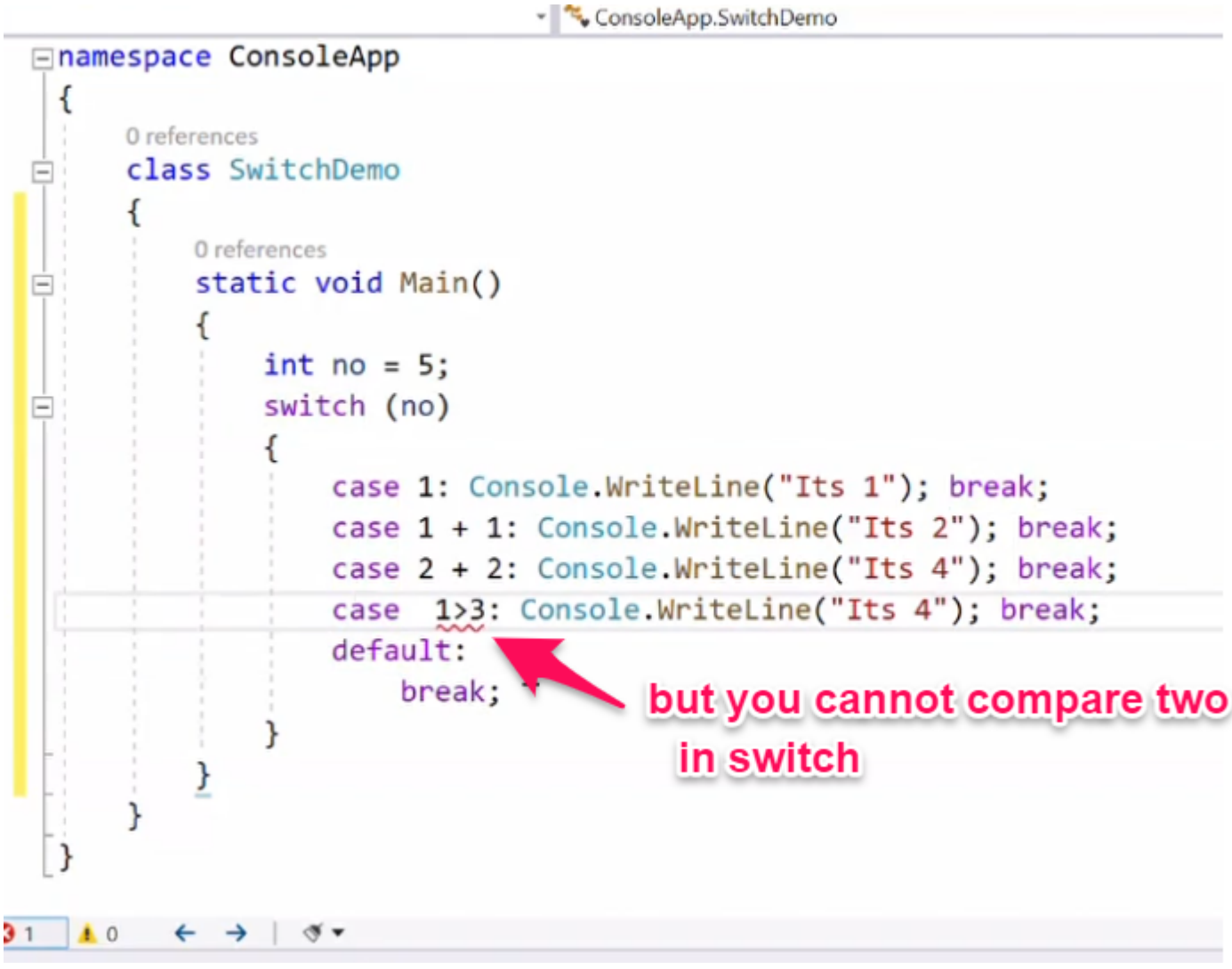
Can we do like this ---------yes -------switch will support this



what switch will not support



So, in the case you can always use + - * / it is possible, so you can use expressions here you can use arithmatic opertors here


ConsoleApp.SwitchDemo
namespace ConsoleApp
{
0 references
class SwitchDemo
{
0 references
static void Main()
{
int no = 5;
switch (no)
{
case 1: Console.WriteLine("Its 1"); break;
case 1 + 1: Console.WriteLine("Its 2"); break;
case 2 + 2: Console.WriteLine("Its 4"); break;
case 1>3: Console.WriteLine("Its 4"); break;
default:
break;
}
}
}
}
but you cannot compare two values here
in switch
1
0

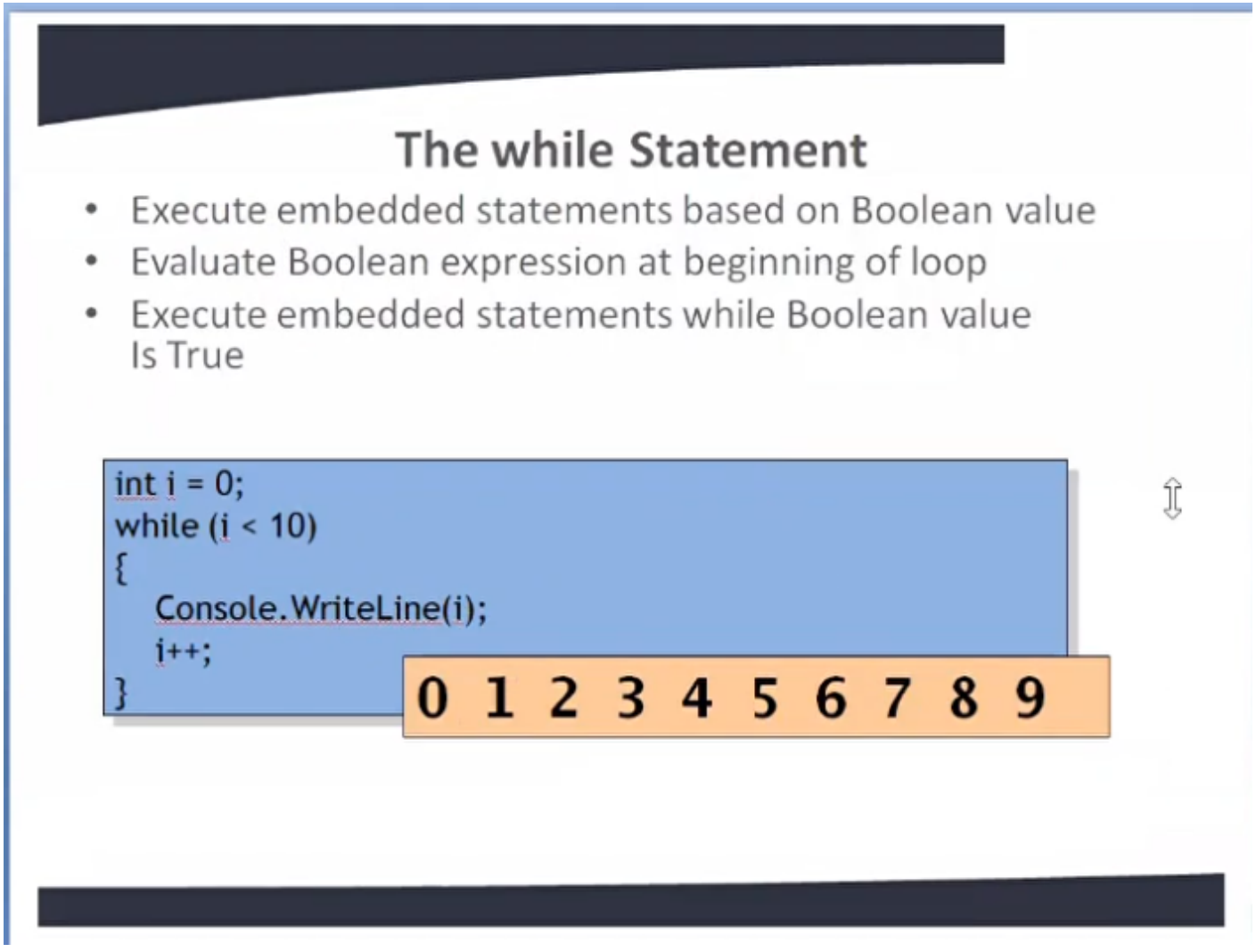
The switch Statement

- Use switch statements for multiple case blocks
- Use break statements to ensure that no fall through occurs

```
switch (choice)
{
case “Black”:
        Console.WriteLine(“Choice is Black..”);
        break;
case “Blue”:
        Console.WriteLine(“Choice is Blue..”);
        break;
case “Red”:
        Console.WriteLine(“Choice is Red..”);
default:
        Console.WriteLine(“No Color choice..”);
        break;
}
```

LOOPs

While

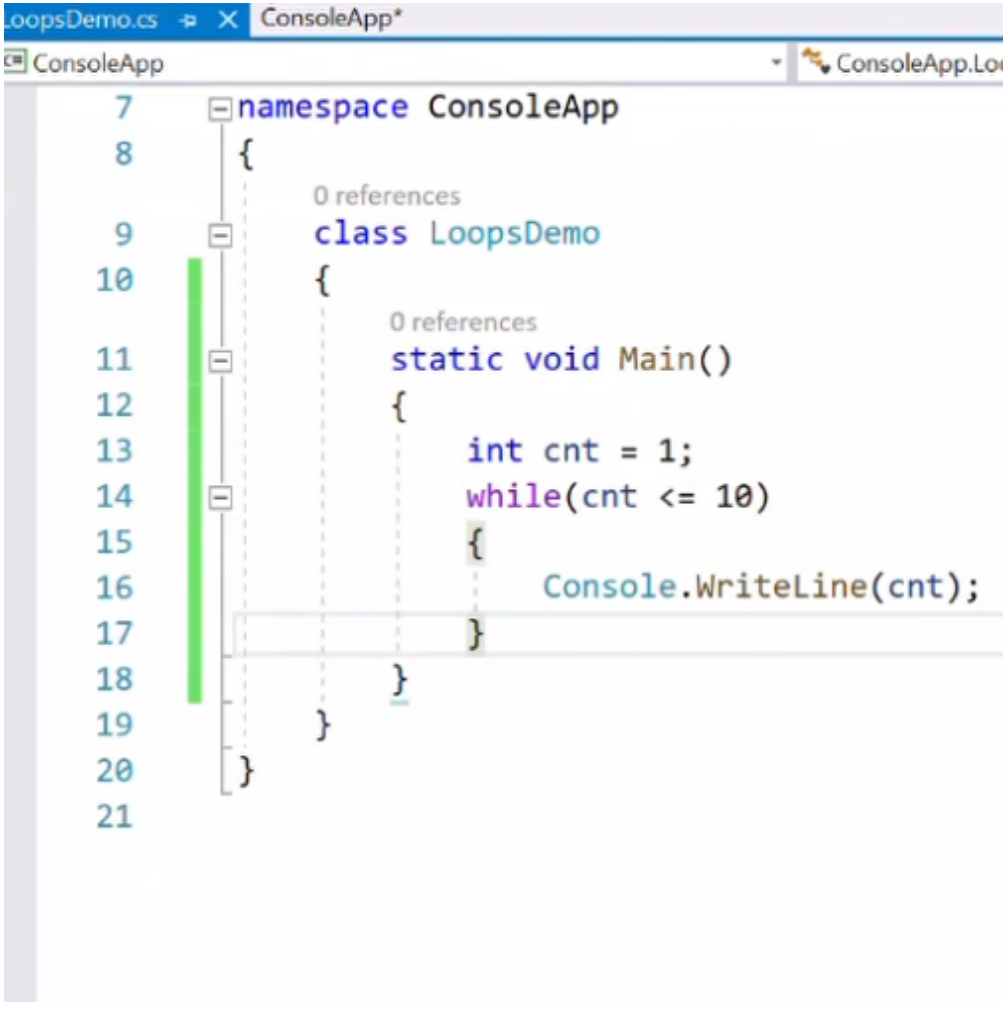


Loops are use to perform similar task again and again

How many types of loop we have ?

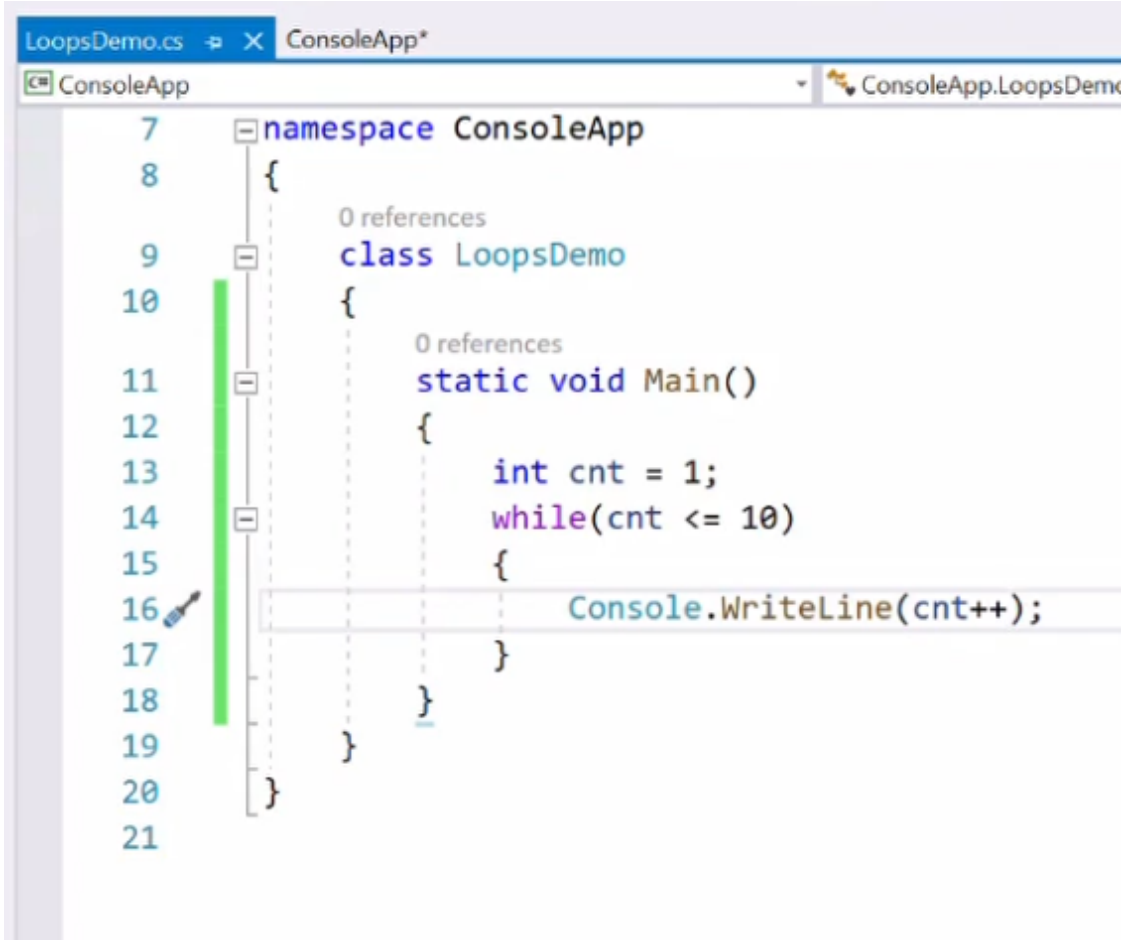
We have only 2 types of loops

1. **Entry Restricted**
2. **Exit Restricted------even if the condition is false exist restricted will run atleast once**

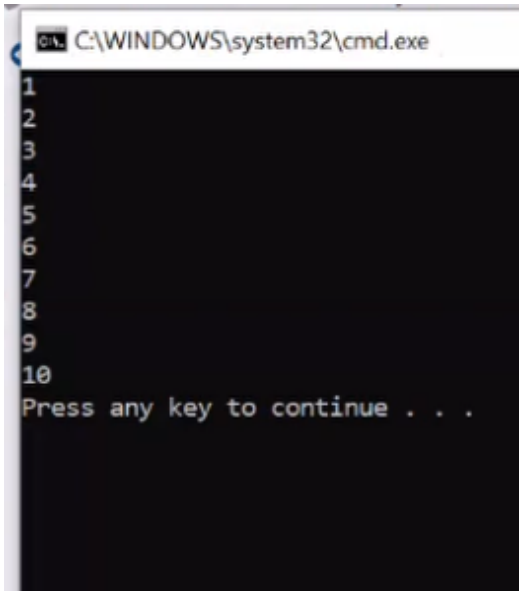


What is the output of this loop ? --1111111111111111111111.....infinite

because we are not incrementing the value of cnt here

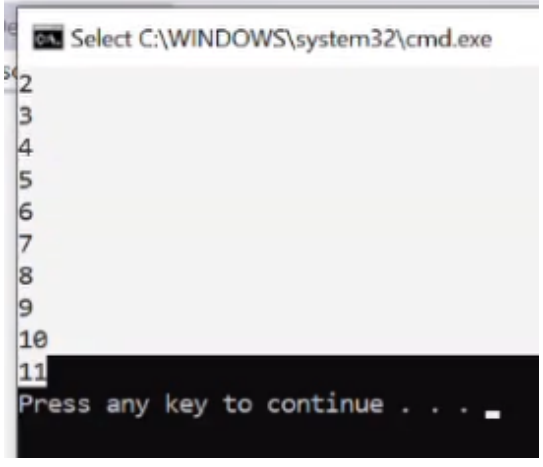


Now what is the output


C:\WINDOWS\system32\cmd.exe
1
2
3
4
5
6
7
8
9
10
Press any key to continue . . .

```
namespace ConsoleApp
{
    0 references
    class LoopsDemo
    {
        0 references
        static void Main()
        {
            int cnt = 1;
            while(cnt <= 10)
            {
                Console.WriteLine(++cnt);
            }
        }
    }
}
```

Now what will be the output?

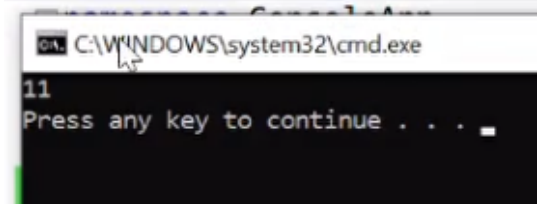


Now we don't get any output because the condition is false 10 = 10 not 10<10

do While

```
ConsoleApp.LoopsDem
namespace ConsoleApp
{
    0 references
    class LoopsDemo
    {
        0 references
        static void Main()
        {
            int cnt = 10;
            do
            {
                Console.WriteLine(++cnt);
            } while (cnt < 10);
        }
    }
}
```

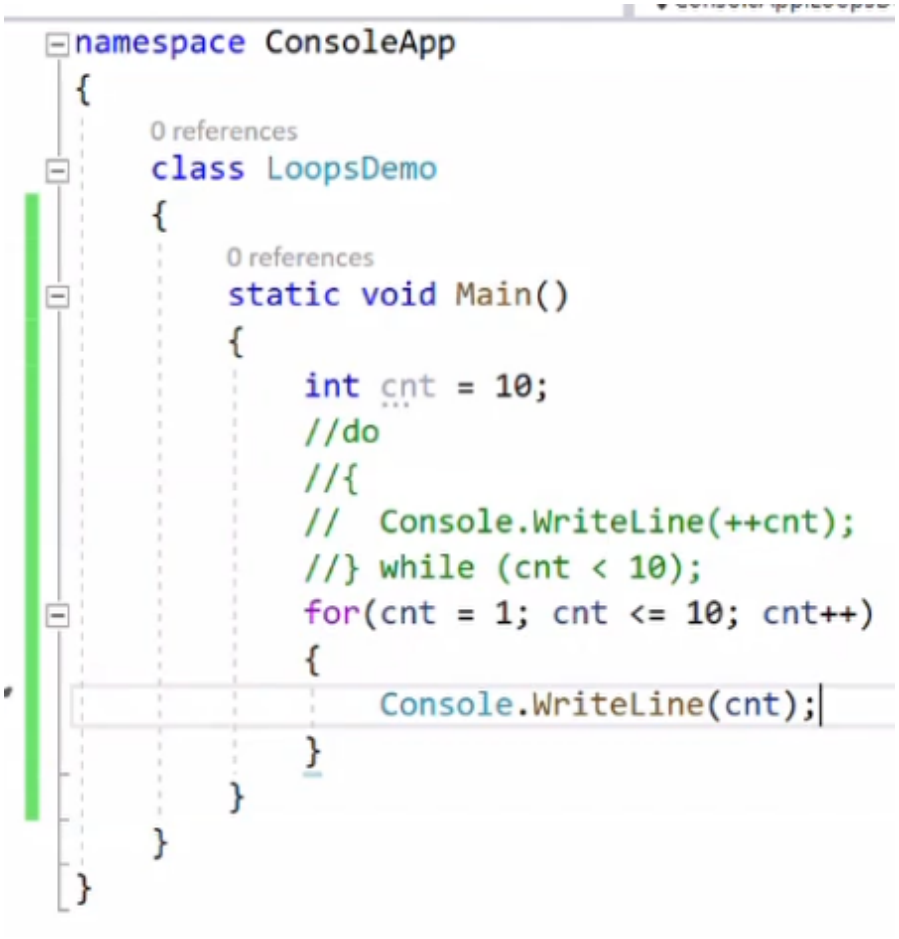
Now what will be the output in case of do while

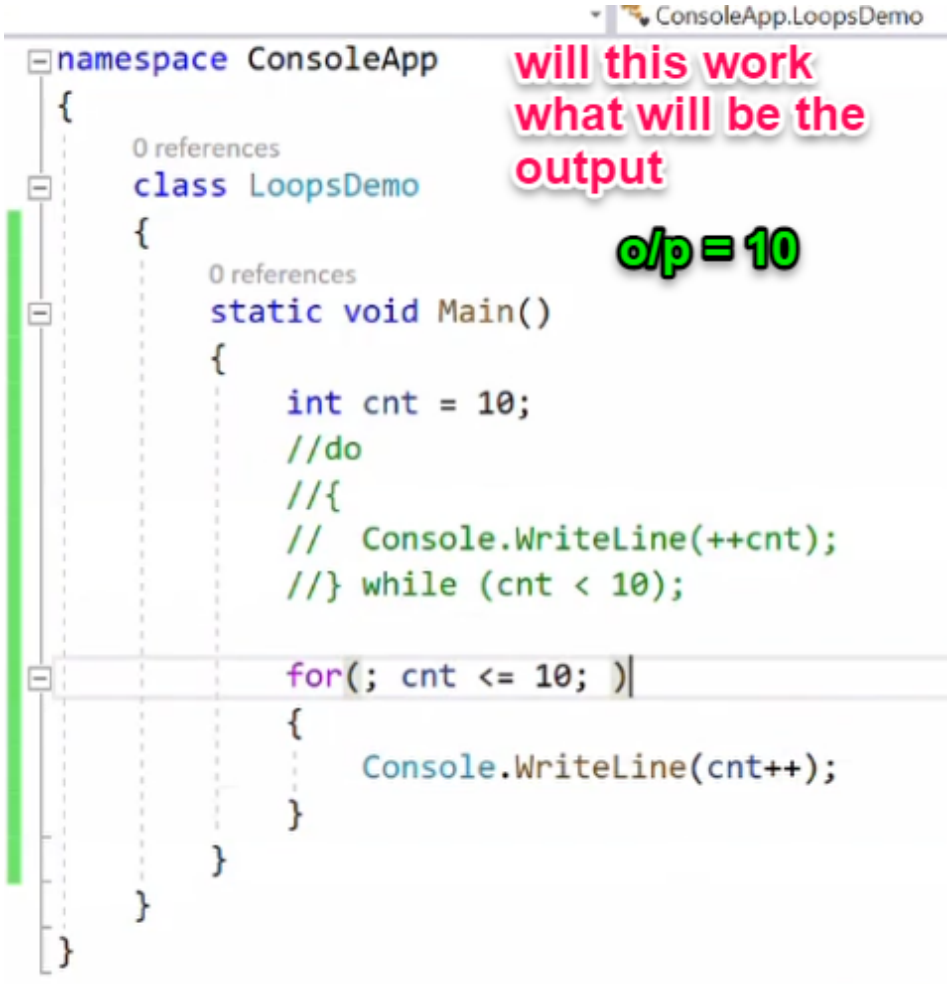


WhenEver you are working with loop there are 3 most important parts of the loop

1. **Initialization**
2. **Condition Check**
3. **Increment or Decrement**

For Loop


namespace ConsoleApp
{
0 references
class LoopsDemo
{
0 references
static void Main()
{
int cnt = 10;
//do
//{
// Console.WriteLine(++cnt);
//} while (cnt < 10);
for(cnt = 1; cnt <= 10; cnt++)
{
Console.WriteLine(cnt);
}
}
}
}



Where Can we Use While Loop and where we use For Loop or do-while Loop

Can't you do this using for----yes you can do this but you feel convinient /flexible with while loop

For-----when we go with numeric calculation , like to print all natural number from 1 to 100

Numeric Manupulation we go with for

I want to read the character from the user and if the character is equal to vowel (a,e,i,o,u) then i'll print the message saying this particular character is a vowel
and if the user enter 0 then i have to exit the program

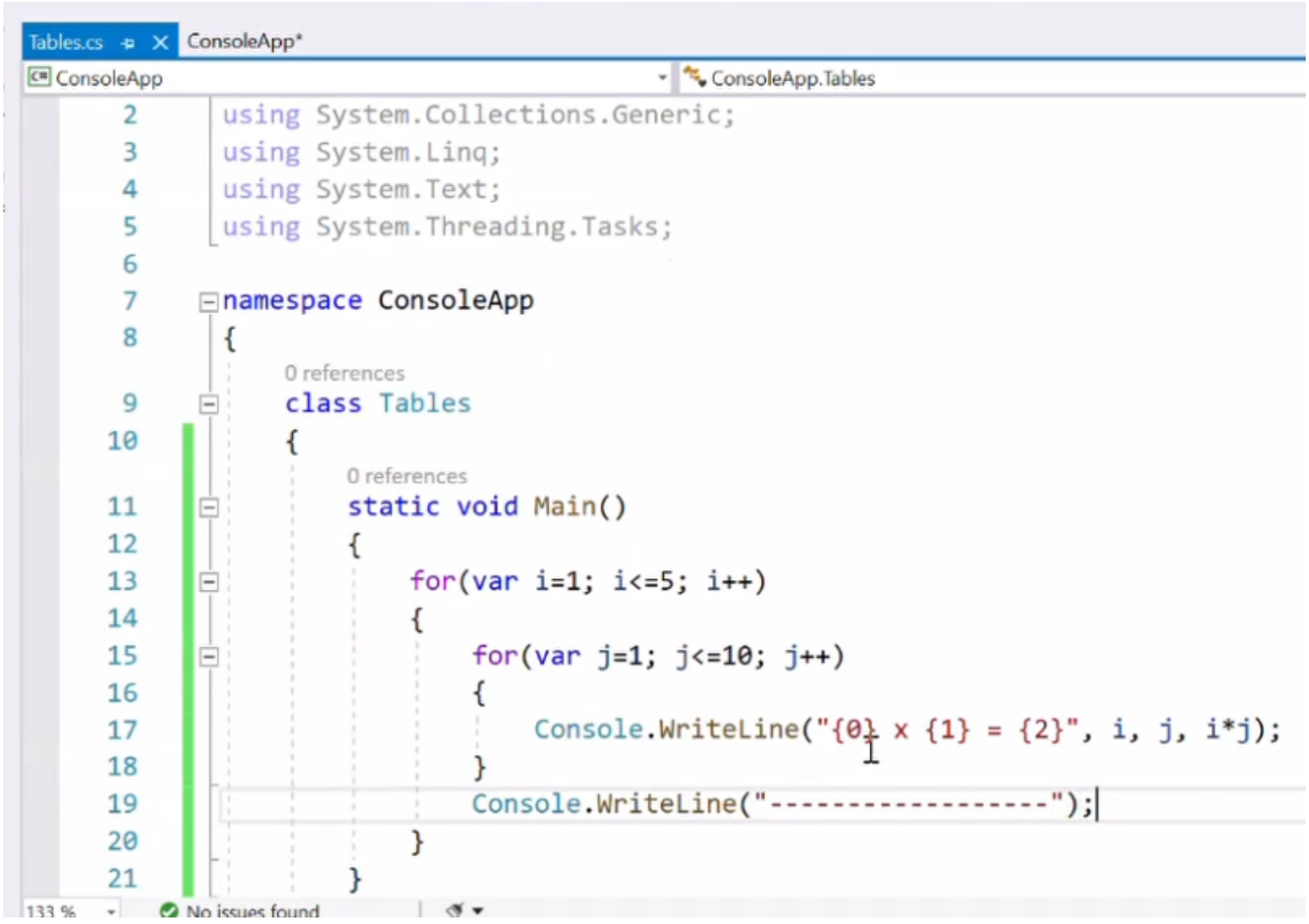
```
static void Main()
{
    int cnt = 1;
    Console.WriteLine("Enter characters:");
    do
    {
        char c = (char)Console.Read();
        if(c == '0')
        {
            break;
        }
        switch (c)
        {
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u': Console.WriteLine("Its ovel"); break;
            default:
                Console.WriteLine("Invalid character");
                break;
```

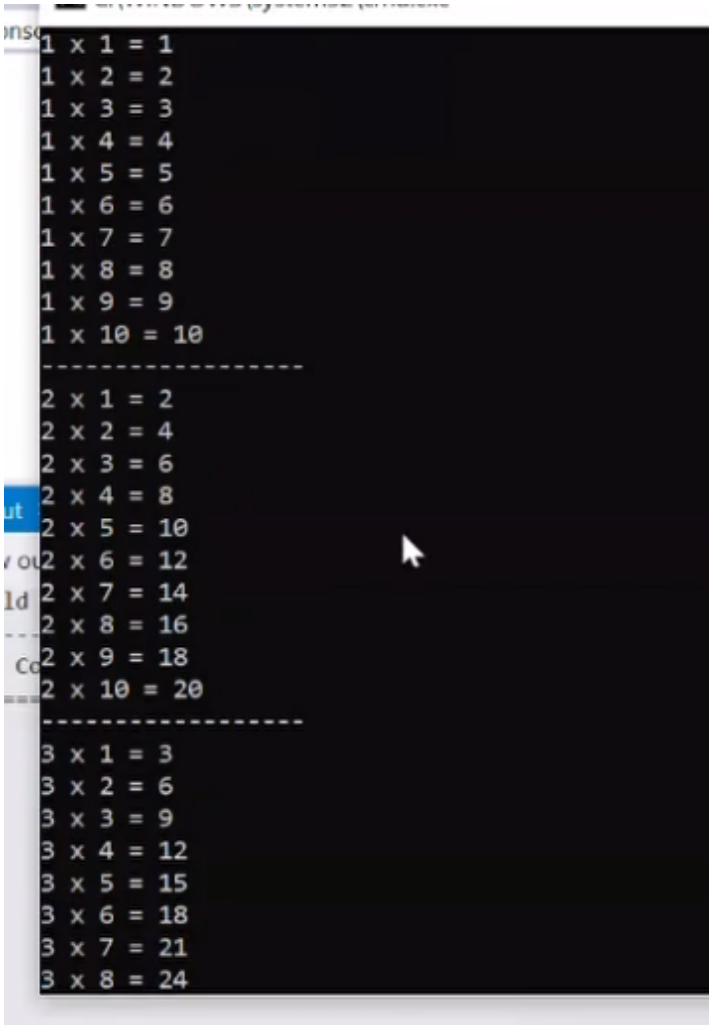
```
            break;
    }

    Console.WriteLine(++cnt);
} while (1==2);
```

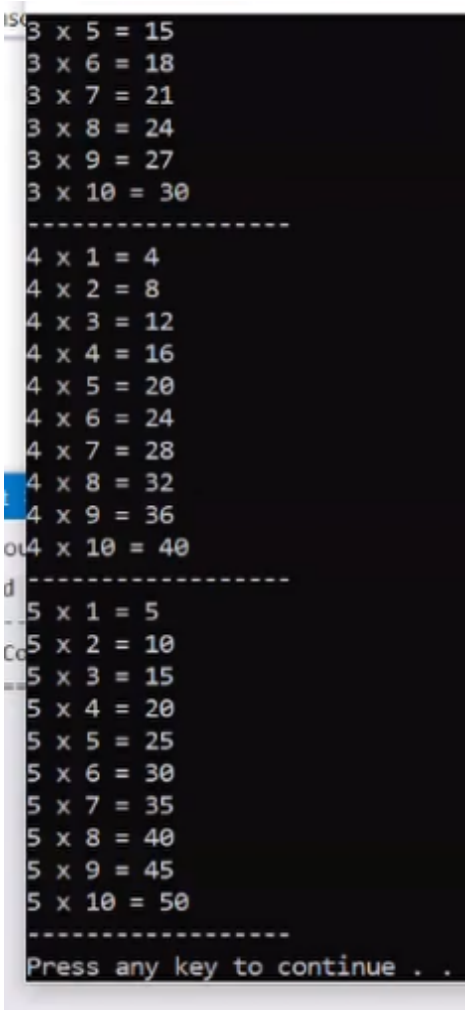
Now if i want to do something like this in for loop it will be really challenging for me that's a reason for string manupulation we'll go with while or do while

Assignment :-take the number from user and convert that to words , for example if the number is 1264 then the output

should be one two six four


Tables.cs
ConsoleApp*
ConsoleApp
ConsoleApp.Tables
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class Tables
{
0 references
static void Main()
{
for(var i=1; i<=5; i++)
{
for(var j=1; j<=10; j++)
{
Console.WriteLine("{0} x {1} = {2}", i, j, i*j);
}
Console.WriteLine("------------------");
}
}
133 %
No issues found


1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24


3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
---
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
---
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

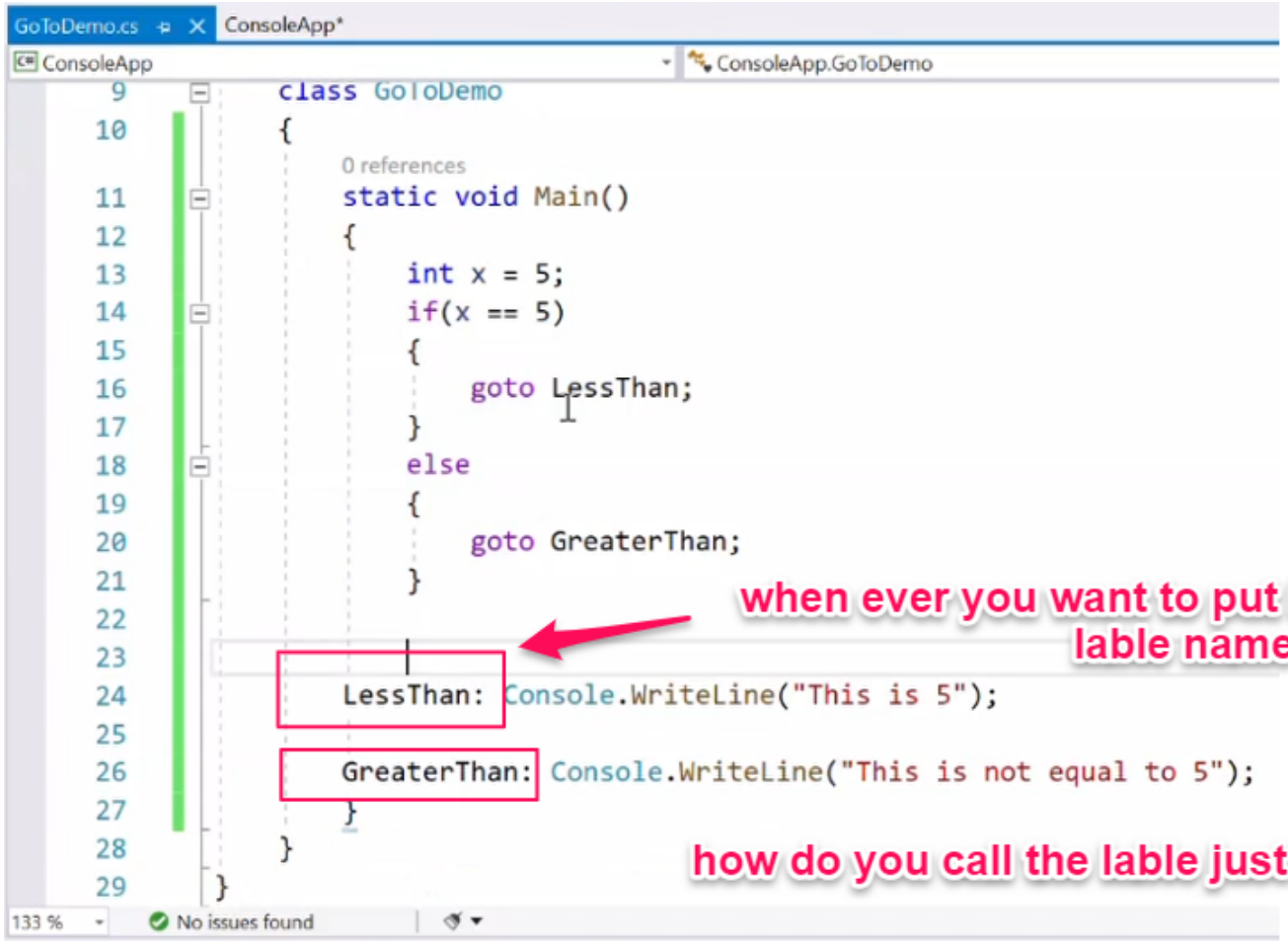
Press any key to continue . .

GOTO demo

The goto Statement

- Flow of control transferred to a labeled statement
- Can easily result in obscure “spaghetti” code

```
if (number % 2 == 0) goto Even;
Console.WriteLine("odd");
goto End;
Even:
Console.WriteLine("even");
End:;
```



when ever you want to put lable you have to put lable name followed by : colon

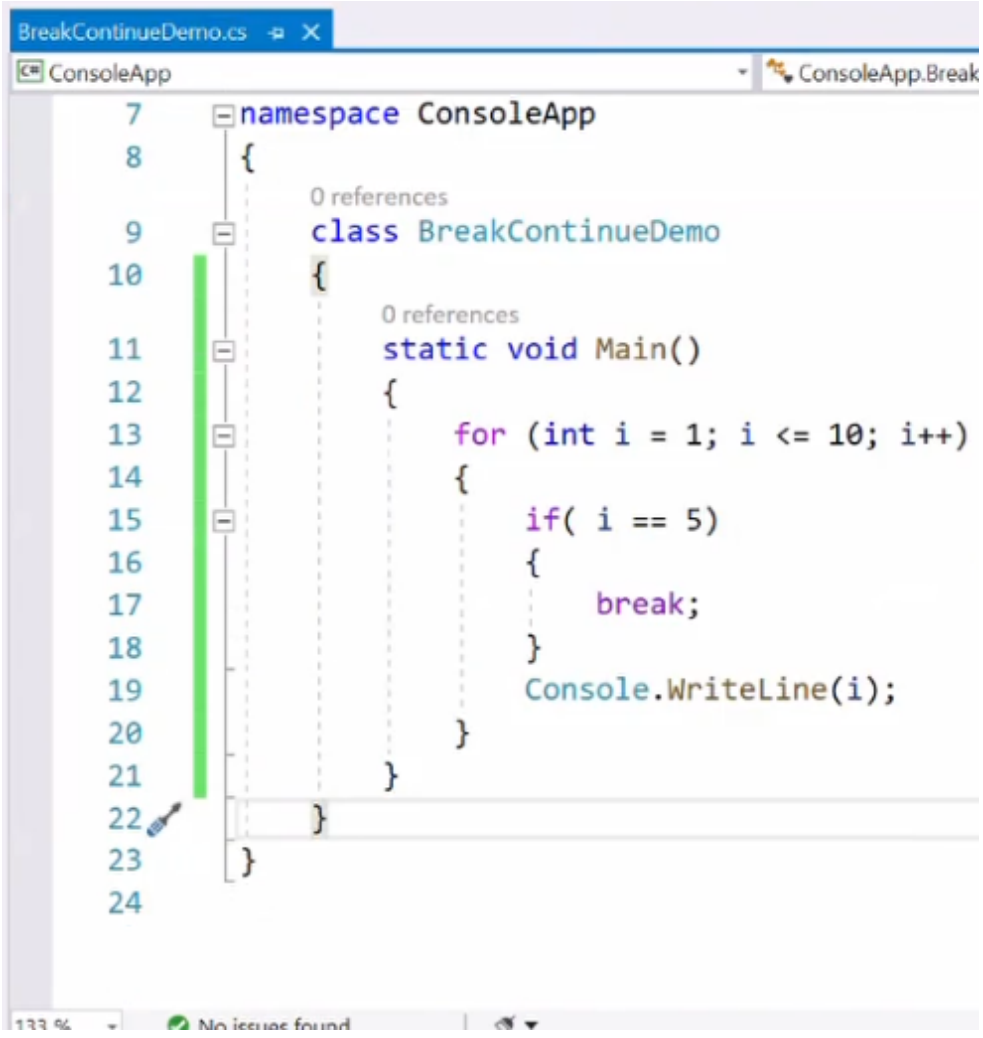
how do you call the lable just write goto followed by lable name

The break and continue Statements

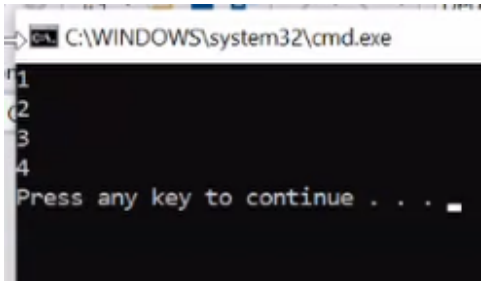
- The break statement jumps out of an iteration
- The continue statement jumps to the next iteration

```
int i = 0;
while (true)
{
    Console.WriteLine(i);
    i++;
    if (i < 10)
        continue;
    else
        break;
}
```

What will be the output ?



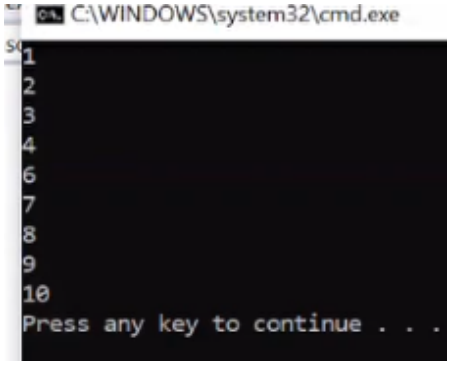
i got 1 2 3 4 because of the fact that i'll say break once the i value becomes 5 it will break this loop



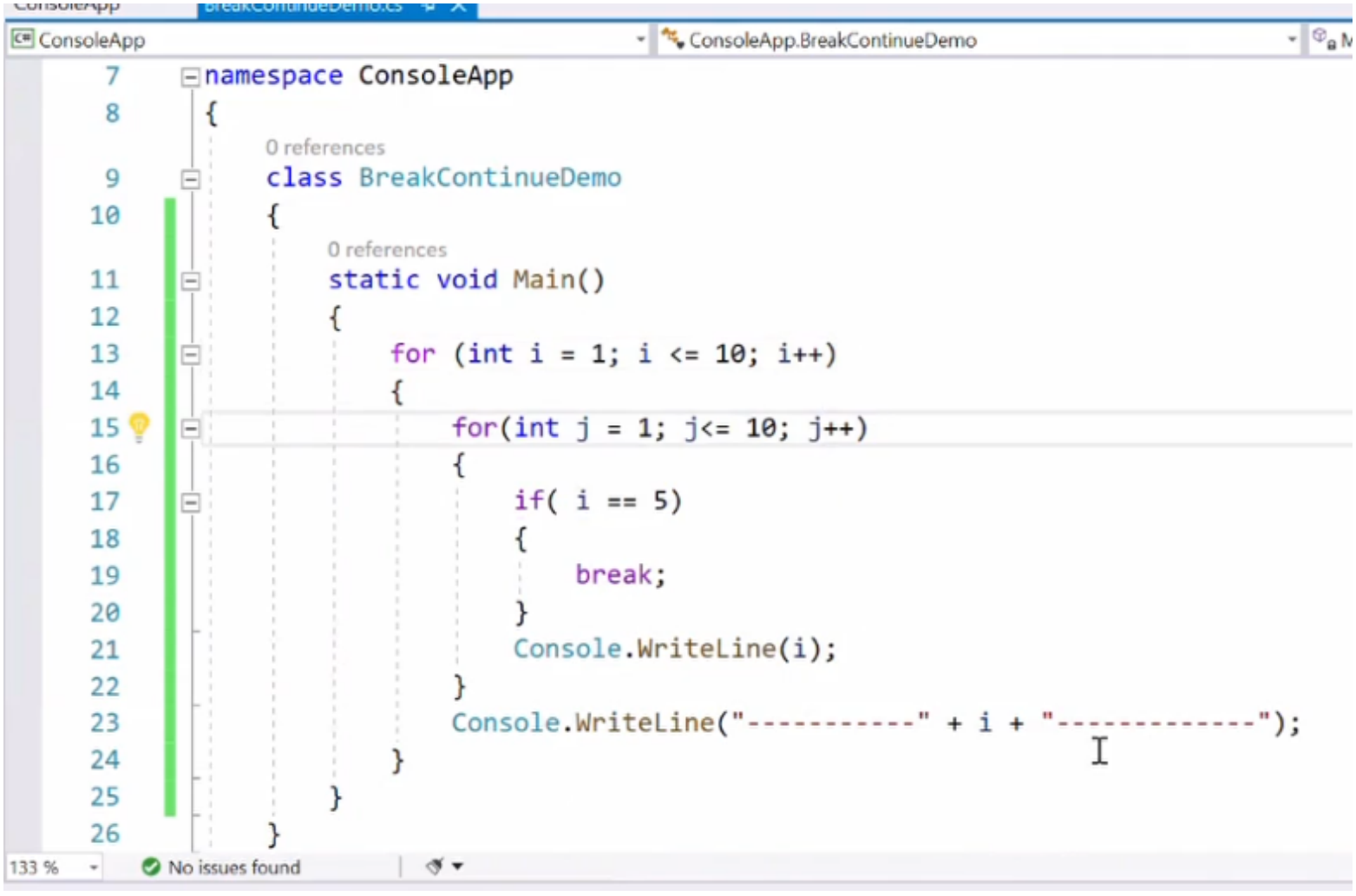
Now what will be the output ?

```
namespace ConsoleApp
{
    0 references
    class BreakContinueDemo
    {
        0 references
        static void Main()
        {
            for (int i = 1; i <= 10; i++)
            {
                if( i == 5)
                {
                    continue;
                }
                Console.WriteLine(i);
            }
        }
    }
}
```

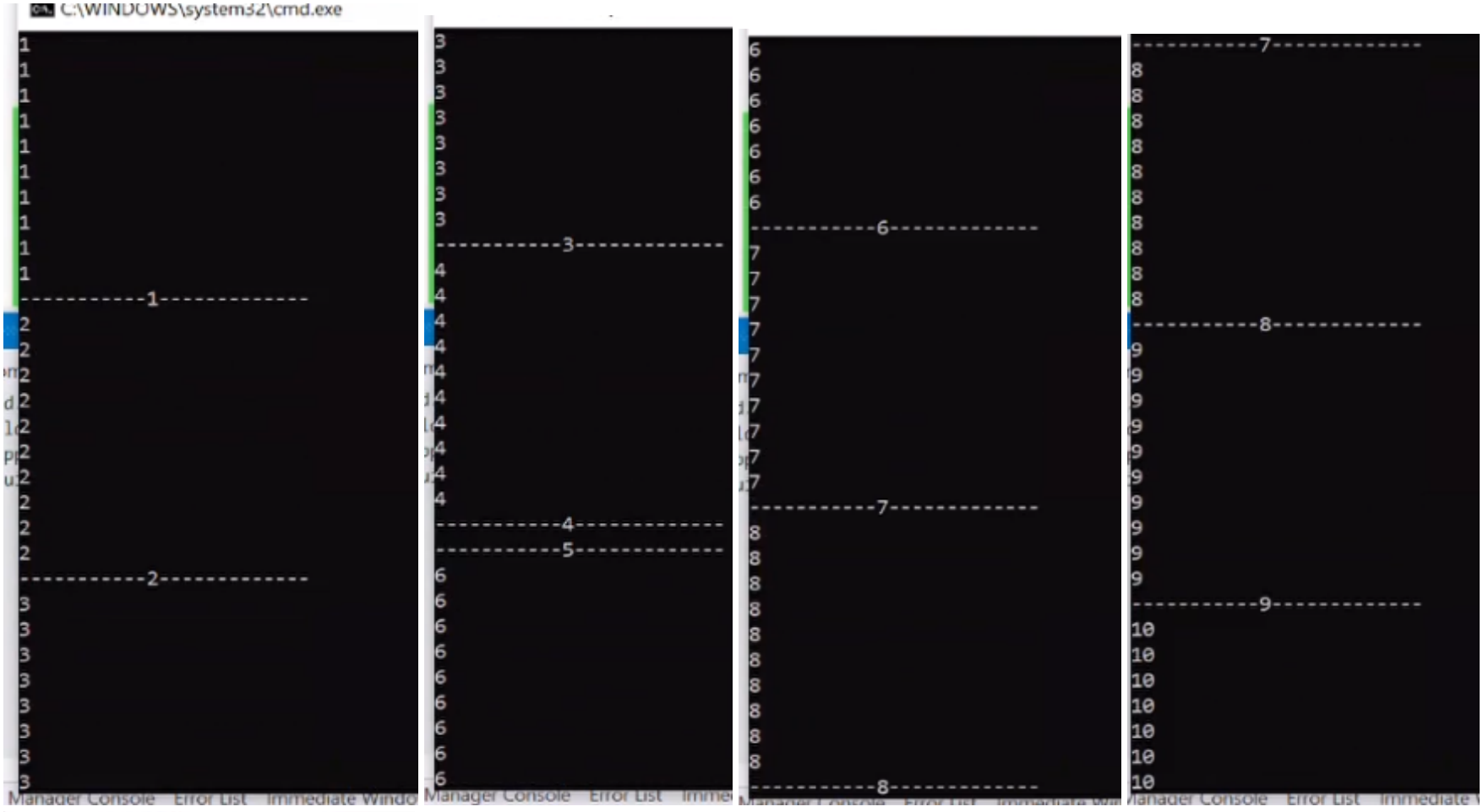
I will get 1 2 3 4 5will be skipped 6 7 8 9 10



Now What will be the output ?



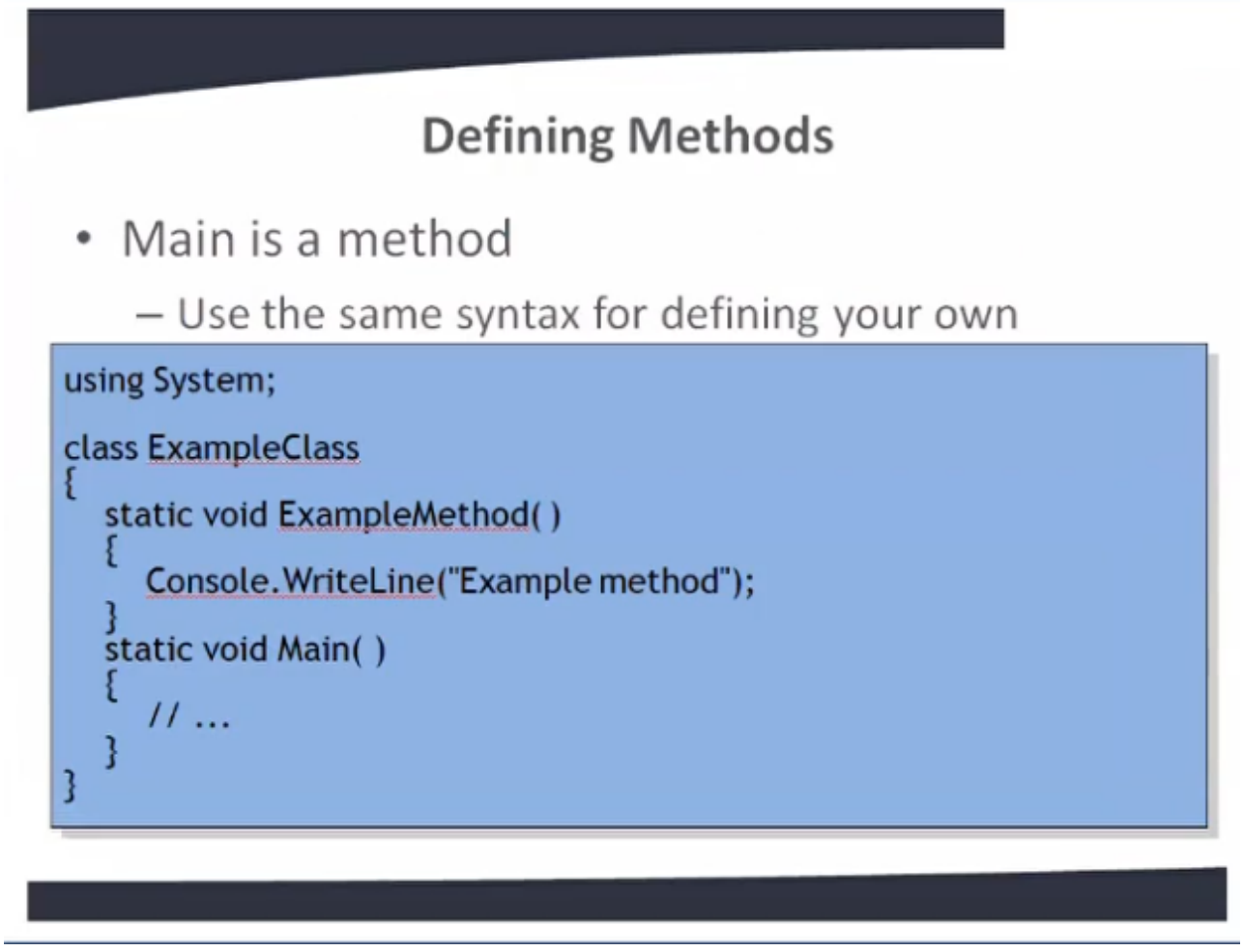
1,2,3,4, ,6,7,8,9,10 print 10 10 times but 5 will not print



So, when you say break what will happen ----------**When ever you put break so break will always break the loop where it is currently present , if it is present in the inner loop it will**

break the inner loop only same with continue

as we are saying break when i==5 so it will keep breaking when i==5 so 5 will not get printed



Whenever you have any sorts of statements to execute push that to a particular method

We always go with

SRP -Single Responsiblity Principle

Assigment :-Read on SOLID principle

As per the solid principle your one method should not perform more than one task, because if you don't do this so maintenance becomes very difficult

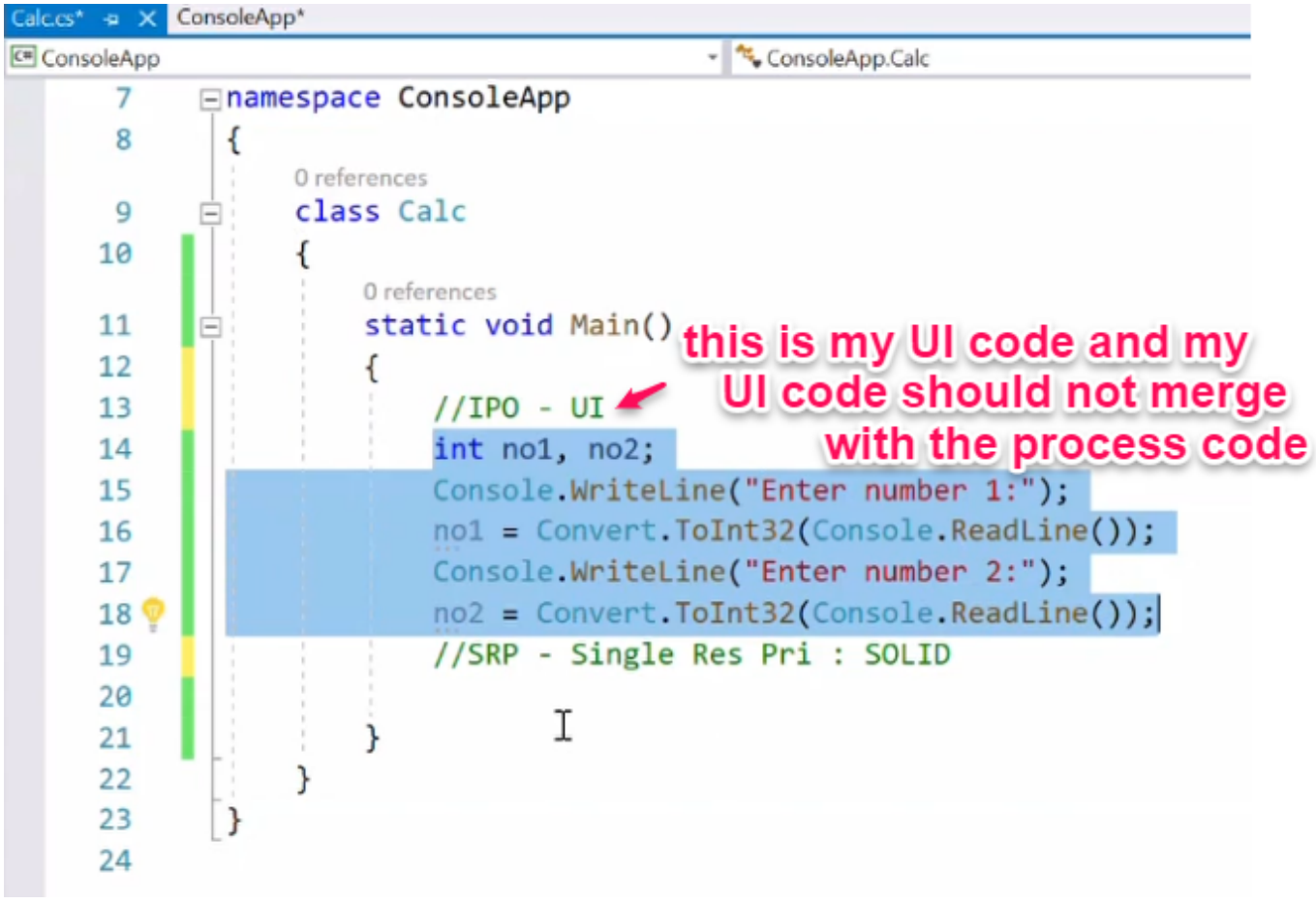
1. maintainance becomes difficult
2. Debugging becomes difficult
3. Reading becomes difficult

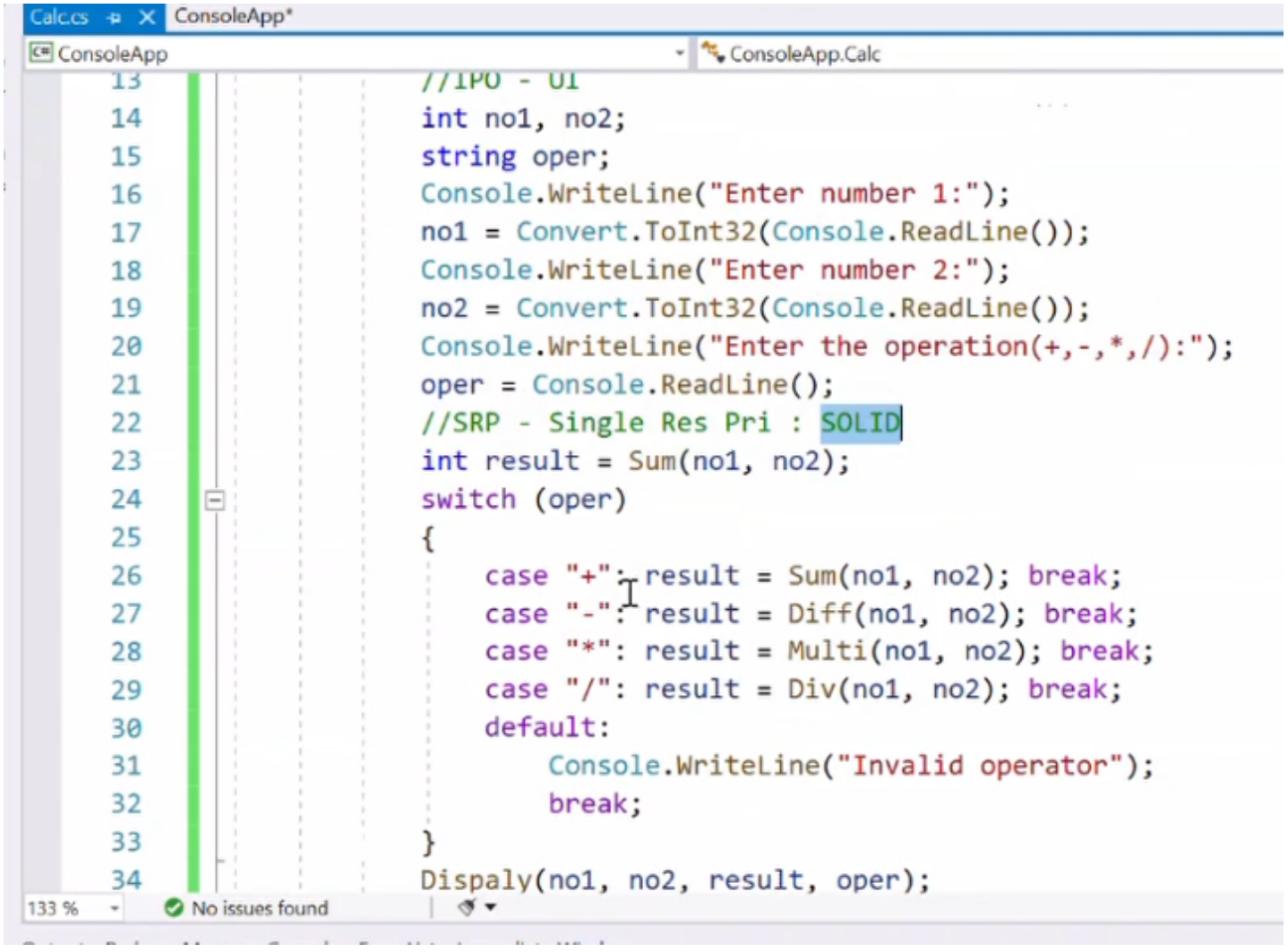
Suppose 9 to 6 you are sitting for the classes ,imagine after 6 o'clock you have to goto shop and manage your shop

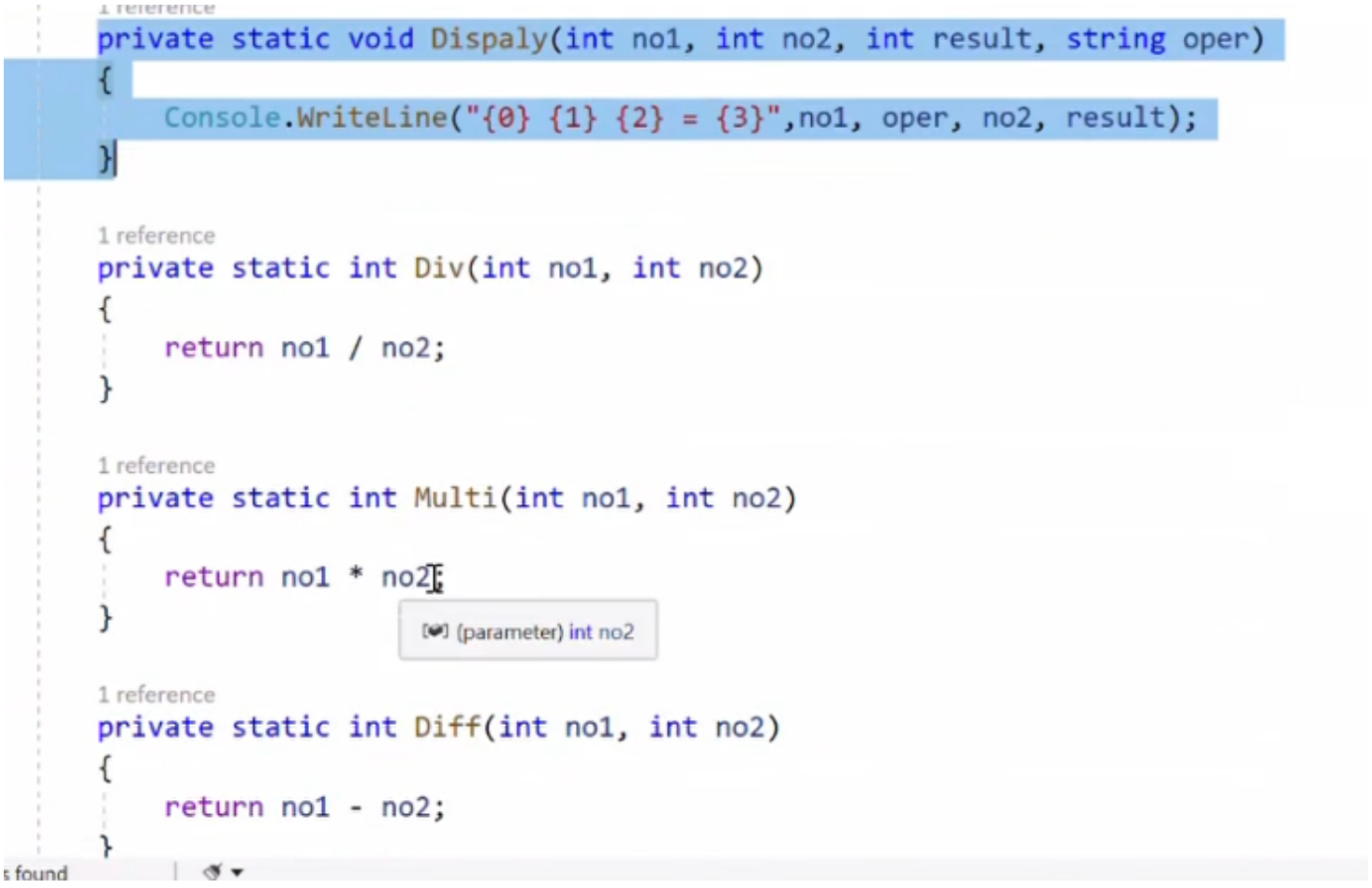
6:30 to 8 you will manage your shop and from 10 you'll go somewhere else and work will you be efficient, no doubt you will gone to work but will you be effecient ---------No

at any given point of time you should perform only one task , you should not perform more than one task

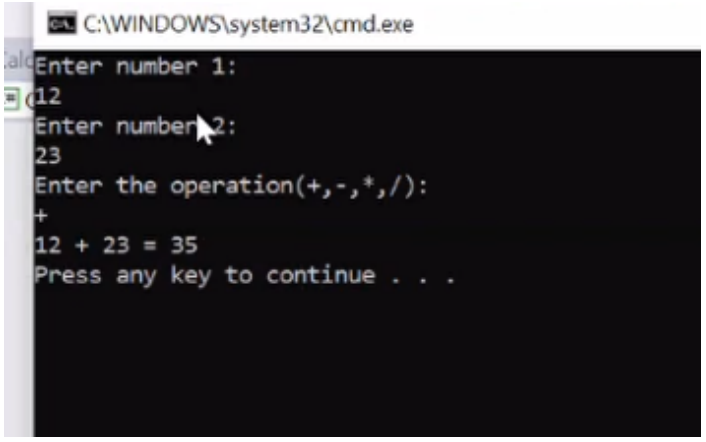
so any program should have a IPO---Input process output


Calc.cs*
ConsoleApp*
ConsoleApp
ConsoleApp.Calc
namespace ConsoleApp
{
0 references
class Calc
{
0 references
static void Main()
{
//IPO - UI
int no1, no2;
Console.WriteLine("Enter number 1:");
no1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter number 2:");
no2 = Convert.ToInt32(Console.ReadLine());
//SRP - Single Res Pri : SOLID
}
}
}
this is my UI code and my
UI code should not merge
with the process code


Calc.cs
ConsoleApp*
ConsoleApp
ConsoleApp.Calc
//IPO - UI
int no1, no2;
string oper;
Console.WriteLine("Enter number 1:");
no1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter number 2:");
no2 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter the operation(+,-,*,/):");
oper = Console.ReadLine();
//SRP - Single Res Pri : SOLID
int result = Sum(no1, no2);
switch (oper)
{
case "+": result = Sum(no1, no2); break;
case "-": result = Diff(no1, no2); break;
case "*": result = Multi(no1, no2); break;
case "/": result = Div(no1, no2); break;
default:
Console.WriteLine("Invalid operator");
break;
}
Dispaly(no1, no2, result, oper);
133 %
No issues found



Look at the above code it is more maintainable more clean so u should always follow SRP


C:\WINDOWS\system32\cmd.exe
Enter number 1:
12
Enter number 2:
23
Enter the operation(+,-,*,/):
+
12 + 23 = 35
Press any key to continue . . .

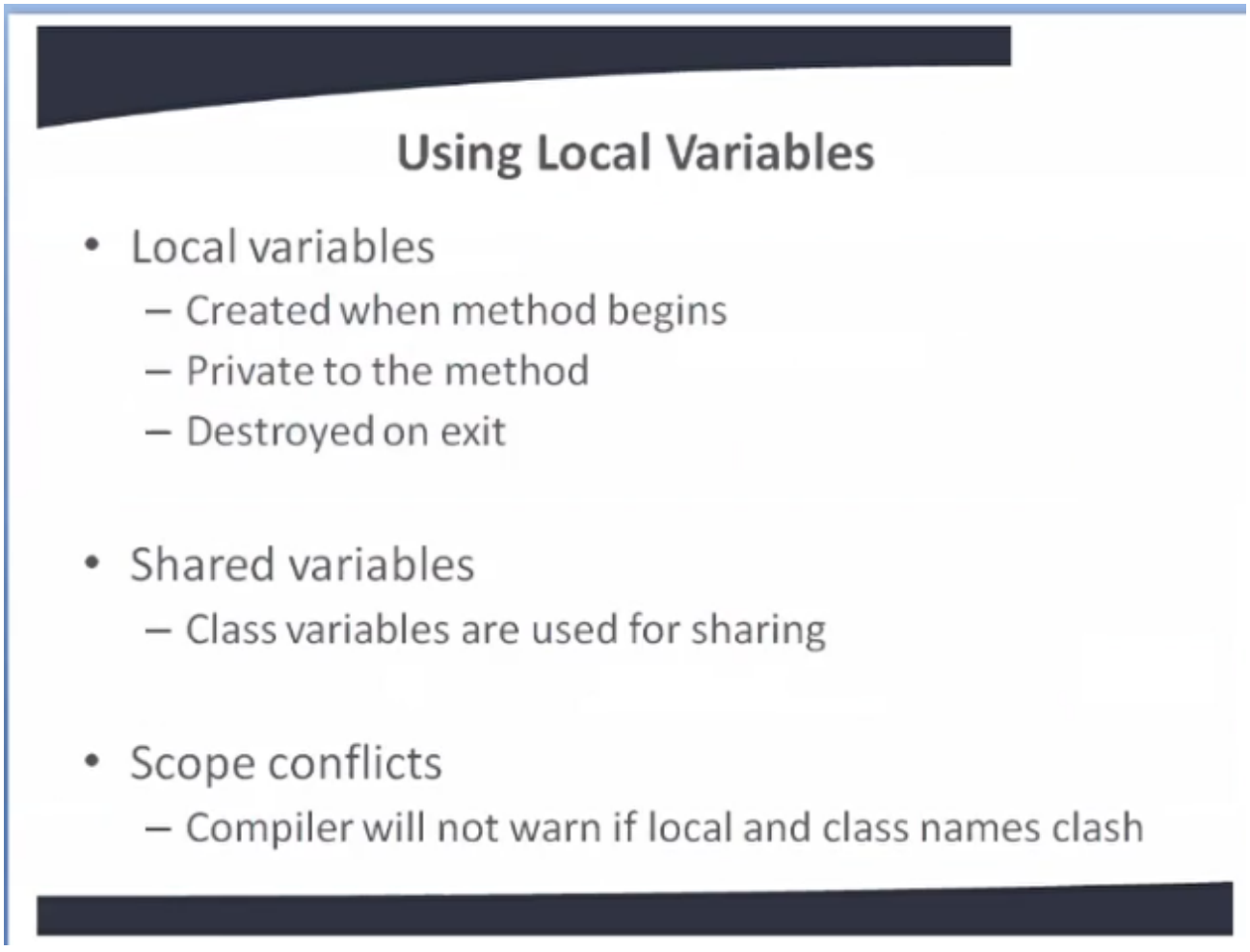
Calling Methods

- After you define a method, you can:
  - Call a method from within the same class
    - Use method's name followed by a parameter list in parentheses
  - Call a method that is in a different class
    - You must indicate to the compiler which class contains the method to call
    - The called method must be declared with the **public** keyword
  - Use nested calls
    - Methods can call methods, which can call other methods, and so on

Using the return Statement

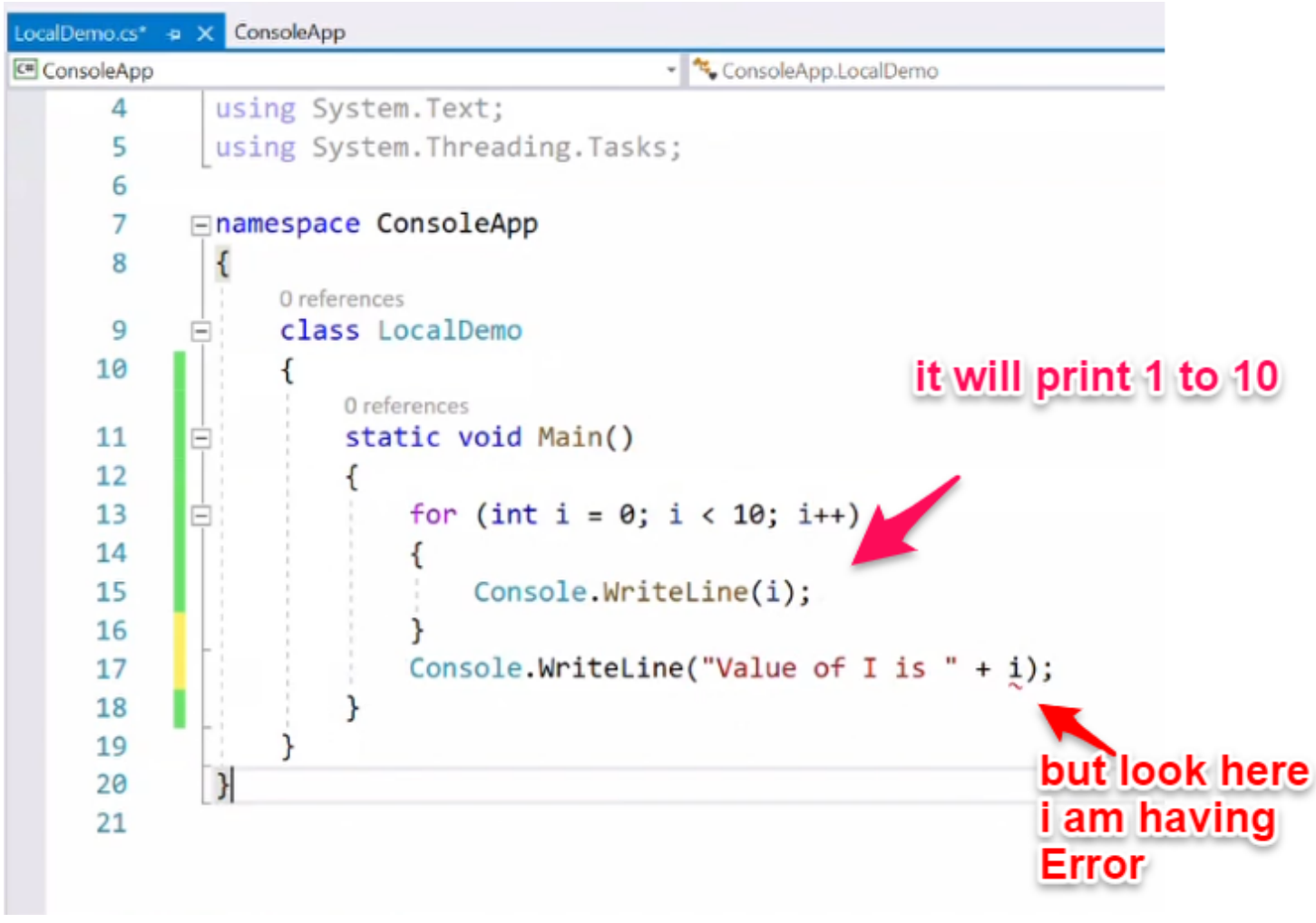
- Immediate return
- Return with a conditional statement

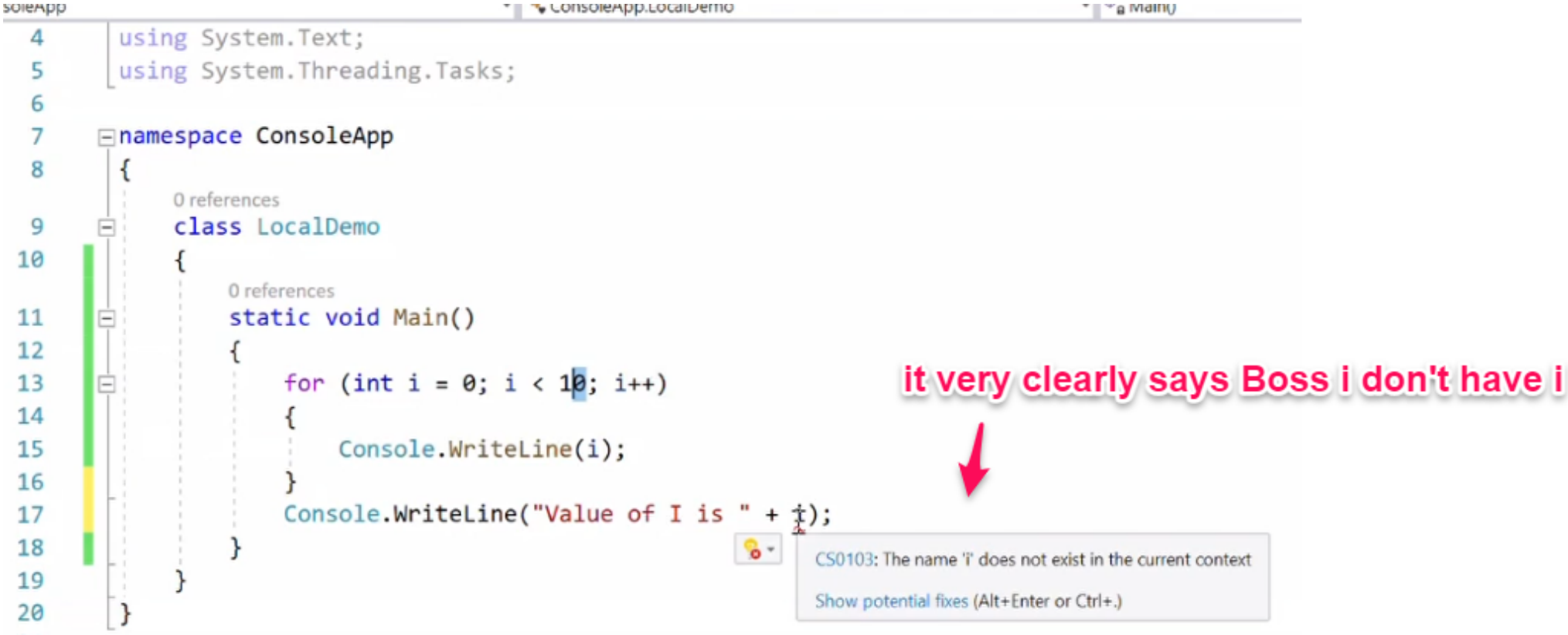
```
static void ExampleMethod( )
{
   int numBeans;
   //...

   Console.WriteLine("Hello");
   if (numBeans < 10)
      return;
   Console.WriteLine("World");
}
```

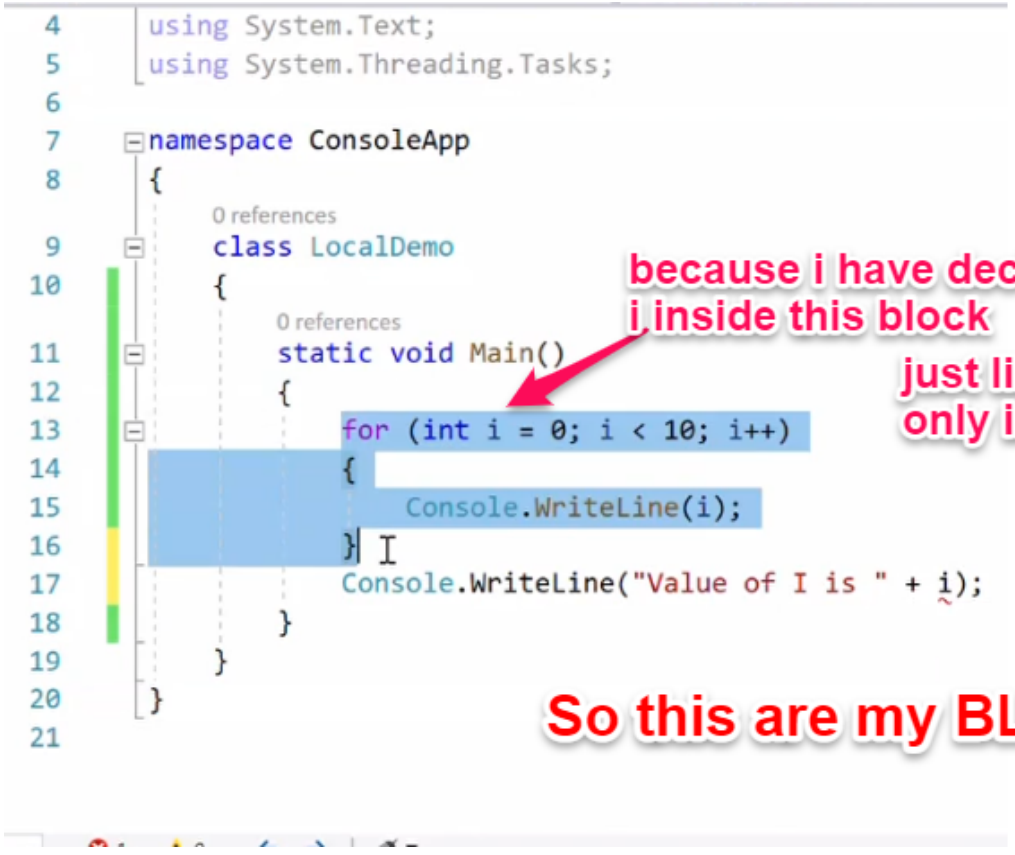


Local variables are those variables which are created inside the method ,private to that particular method and it gets destroyed with that particular method

So whenever you create a particular method you can always even specify the scope just like your let and var


LocalDemo.cs*
ConsoleApp
ConsoleApp
ConsoleApp.LocalDemo
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class LocalDemo
{
0 references
static void Main()
{
for (int i = 0; i < 10; i++)
{
Console.WriteLine(i);
}
Console.WriteLine("Value of I is " + i);
}
}
}
it will print 1 to 10
but look here
i am having
Error


using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class LocalDemo
{
0 references
static void Main()
{
for (int i = 0; i < 10; i++)
{
Console.WriteLine(i);
}
Console.WriteLine("Value of I is " + i);
}
}
}
it very clearly says Boss i don't have i
CS0103: The name 'i' does not exist in the current context
Show potential fixes (Alt+Enter or Ctrl+.)

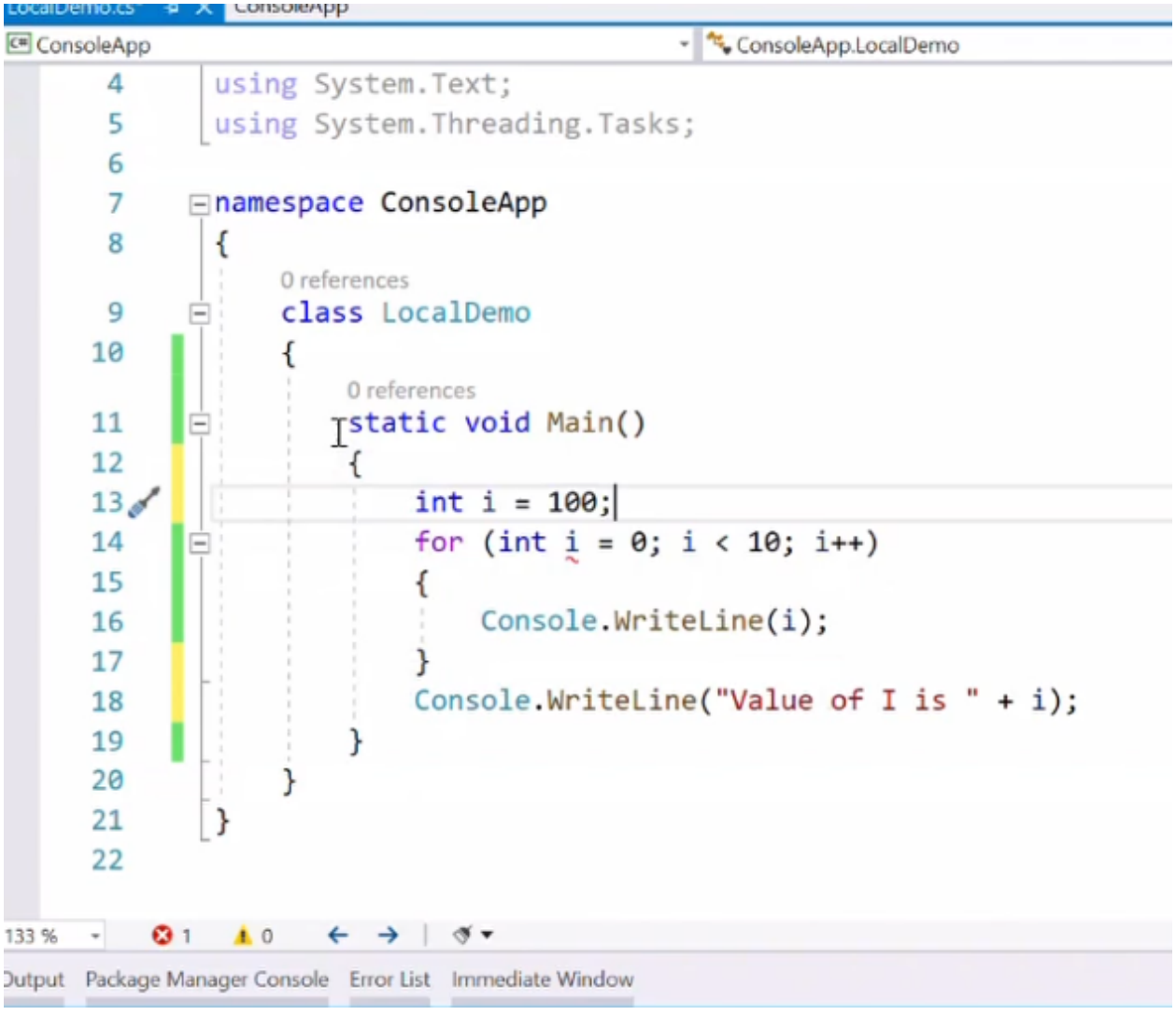


because i have declared
i inside this block

just like your let it will only be accessible
only in the block where it is declared

outside the block it cannot be accessible

So this are my BLOCK Variables


ConsoleApp
ConsoleApp.LocalDemo
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class LocalDemo
{
0 references
static void Main()
{
int i = 100;
for (int i = 0; i < 10; i++)
{
Console.WriteLine(i);
}
Console.WriteLine("Value of I is " + i);
}
}
}
133 %
1
0
Output
Package Manager Console
Error List
Immediate Window