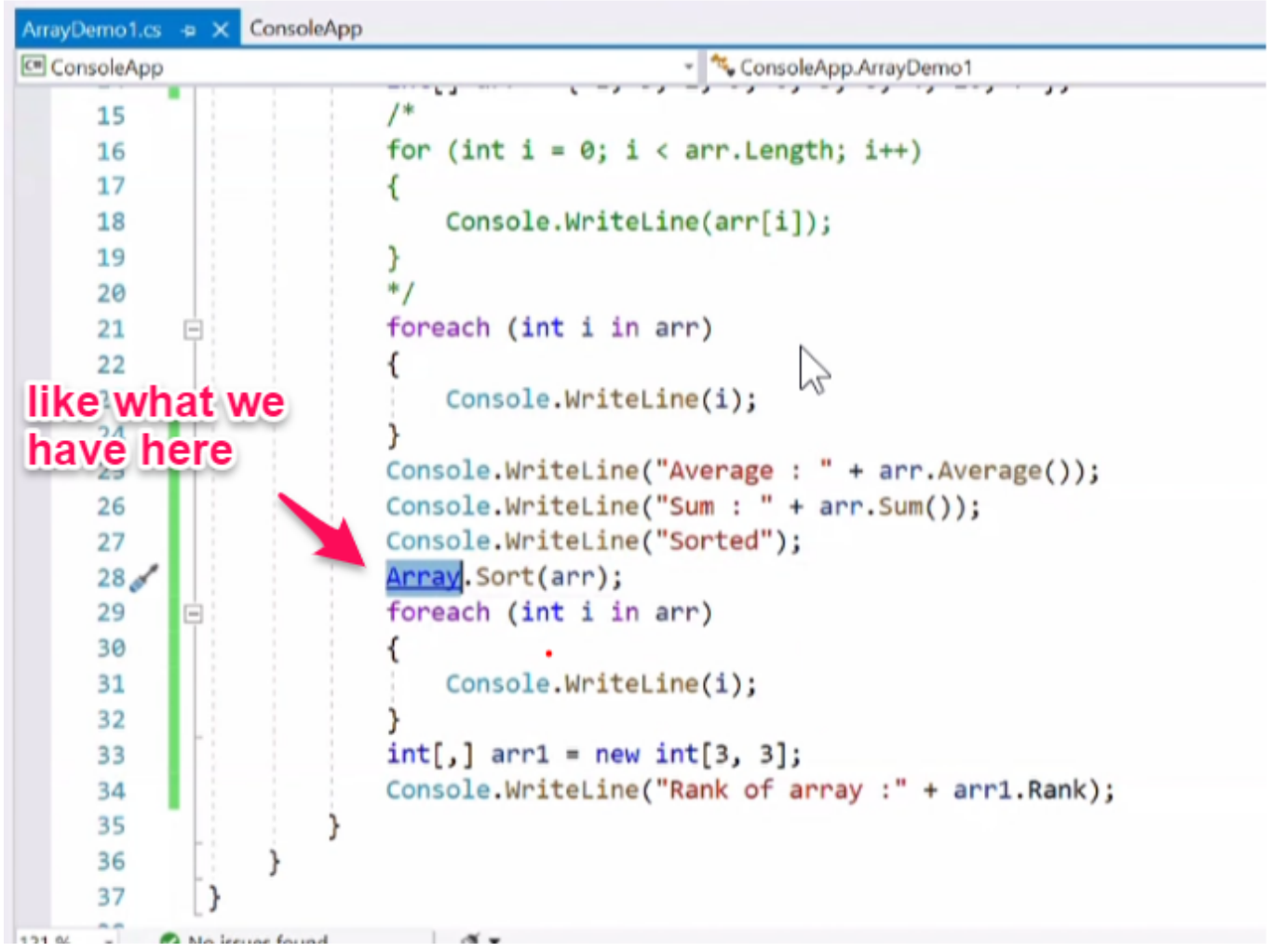
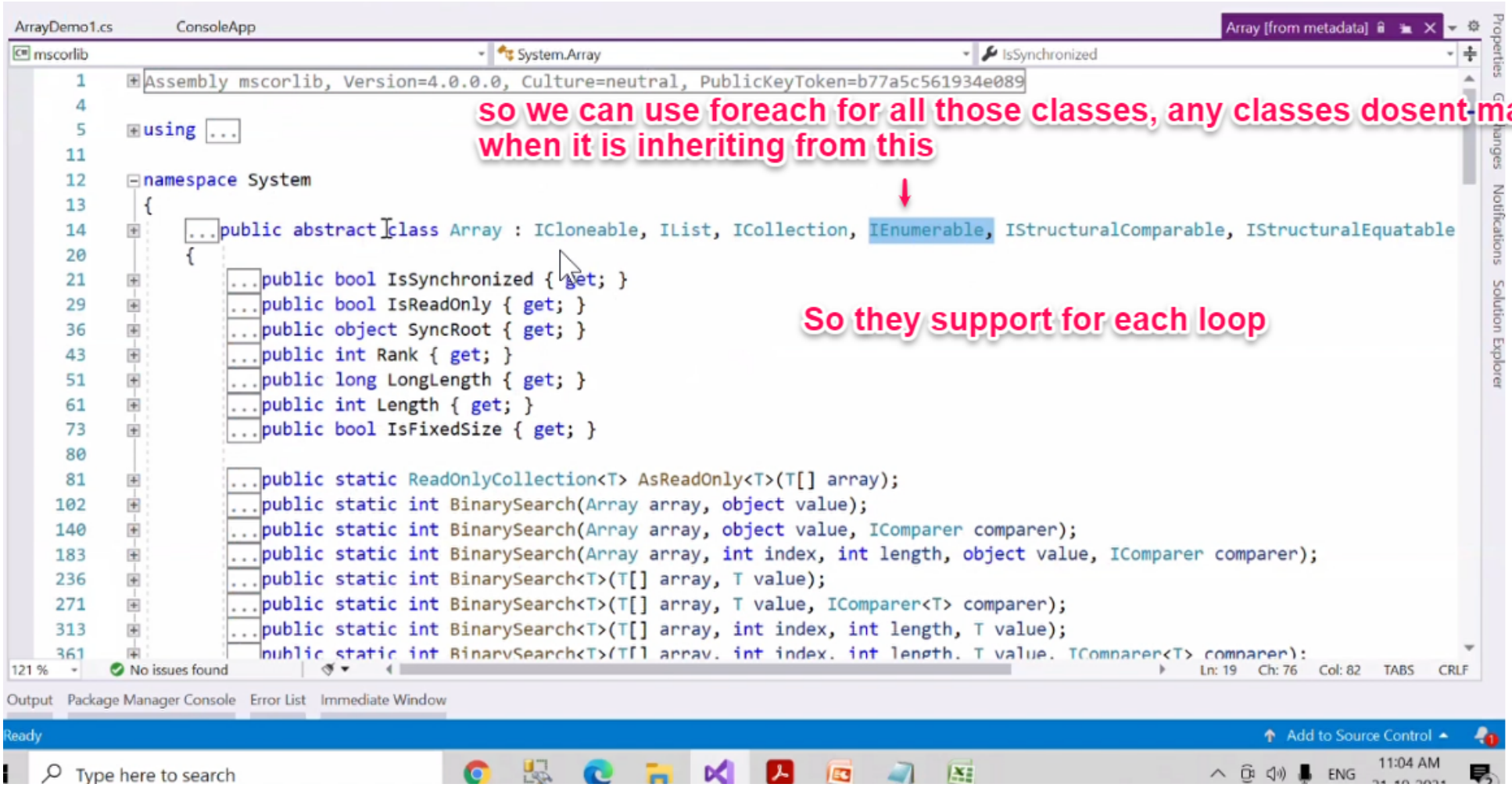
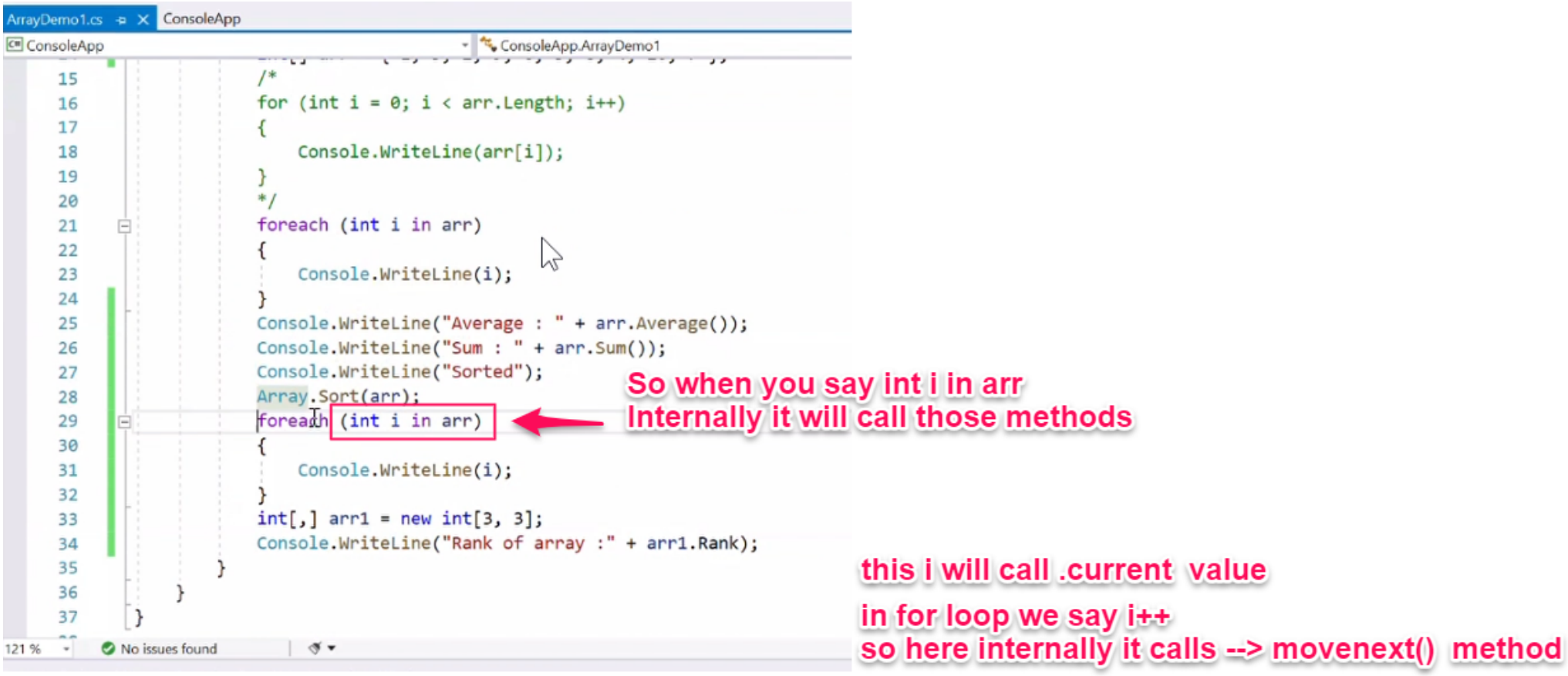
C#-5

That's how like any class which inherits from ienumerable




ArrayDemo1.cs
ConsoleApp
Array [from metadata]
mscorlib
System.Array
IsSynchronized
Assembly mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
using ...
namespace System
{
public abstract class Array : ICloneable, IList, ICollection, IEnumerable, IStructuralComparable, IStructuralEquatable
{
public bool IsSynchronized { get; }
public bool IsReadOnly { get; }
public object SyncRoot { get; }
public int Rank { get; }
public long LongLength { get; }
public int Length { get; }
public bool IsFixedSize { get; }
public static ReadOnlyCollection<T> AsReadOnly<T>(T[] array);
public static int BinarySearch(Array array, object value);
public static int BinarySearch(Array array, object value, IComparer comparer);
public static int BinarySearch(Array array, int index, int length, object value, IComparer comparer);
public static int BinarySearch<T>(T[] array, T value);
public static int BinarySearch<T>(T[] array, T value, IComparer<T> comparer);
public static int BinarySearch<T>(T[] array, int index, int length, T value);
so we can use foreach for all those classes, any classes dosent matter when it is inheriting from this
So they support for each loop
121 %
No issues found
Ln: 19
Ch: 76
Col: 82
TABS
CRLF
Output
Package Manager Console
Error List
Immediate Window
Ready
Add to Source Control
Type here to search
ENG
11:04 AM

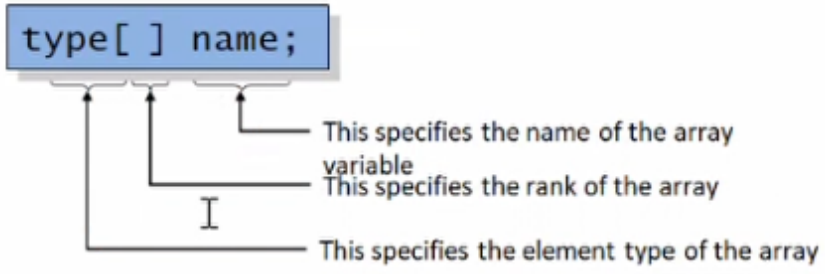


you can read more about that

like when ever you want to know more about internal methods, you just have to hold ctrl key and click on that

Array Notation in C#

- You declare an array variable by specifying:
  - The element type of the array
  - The rank of the array
  - The name of the variable



Array Rank

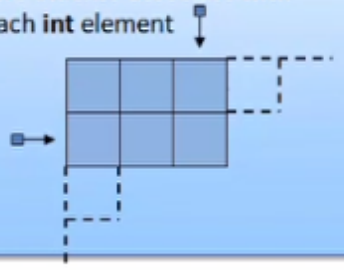
- Rank is also known as the array dimension
- The number of indexes associated with each element

```
long[ ] row;
```

Rank 1: One-dimensional
Single index associates with each **long** element

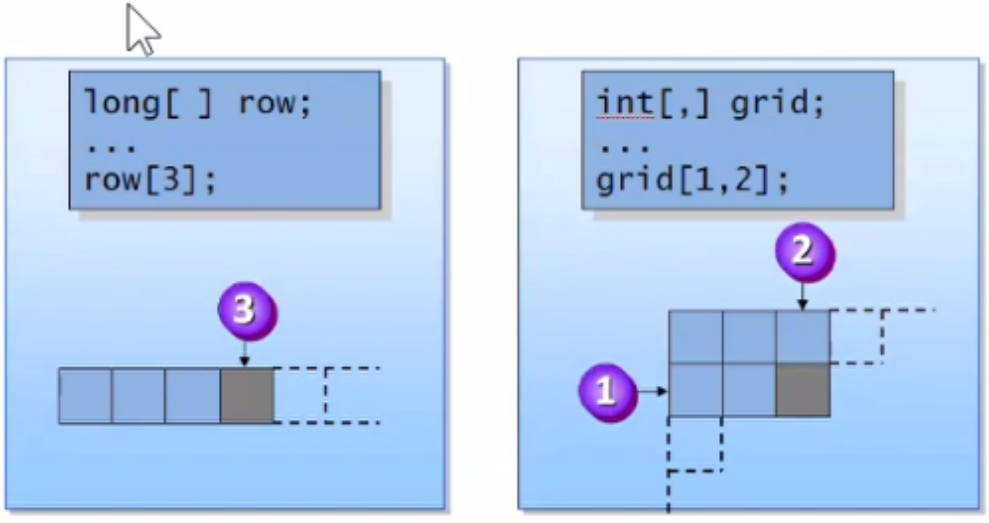
```
int[,] grid;
```

Rank 2: Two-dimensional
Two indexes associate with each **int** element



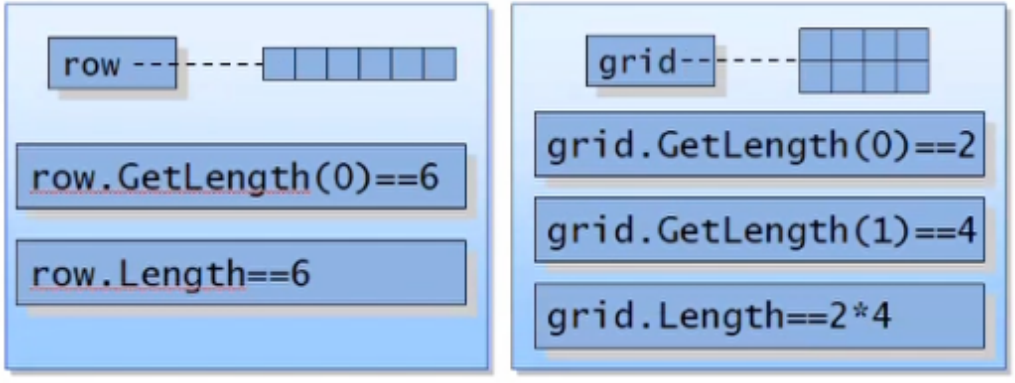
Accessing Array Elements

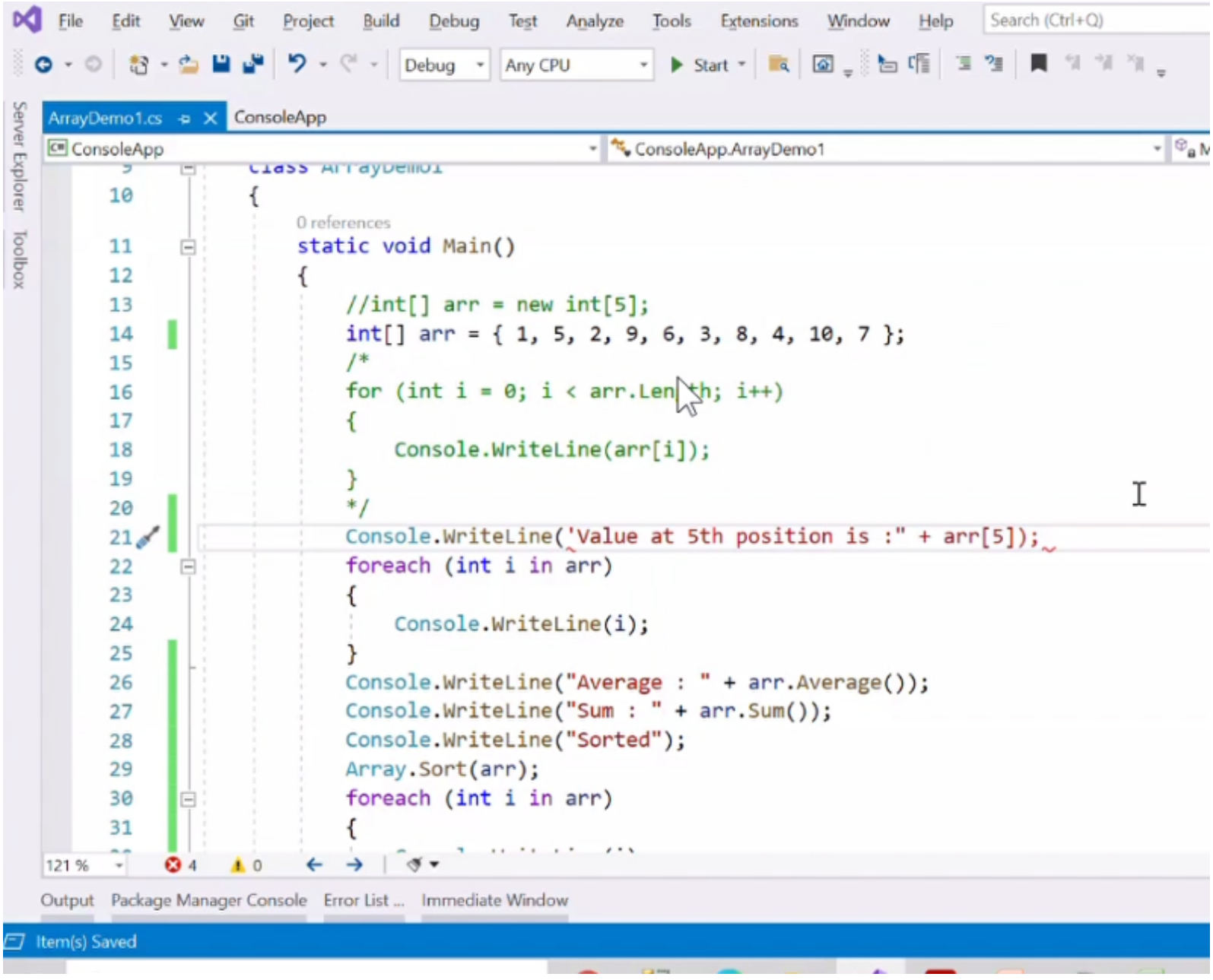
- Supply an integer index for each rank
  - Indexes are zero-based

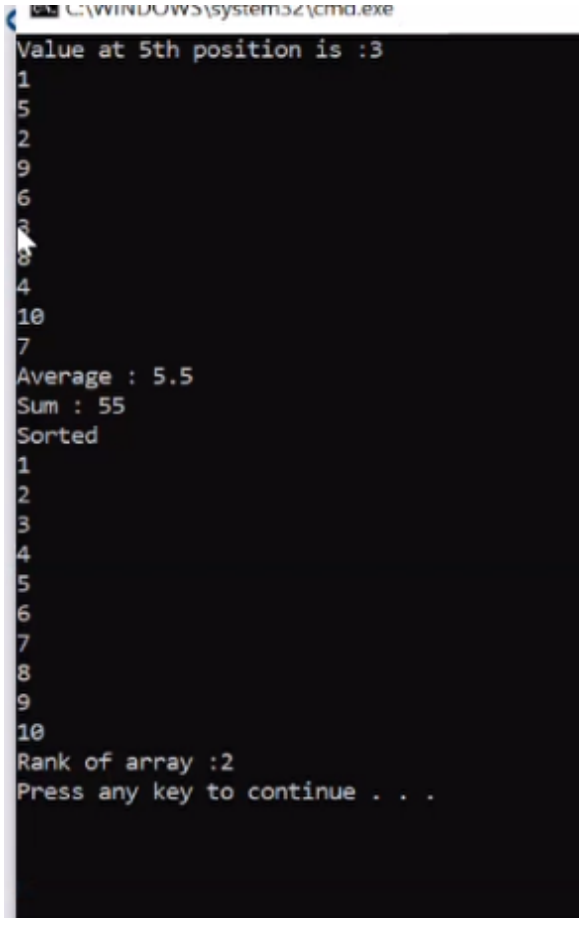


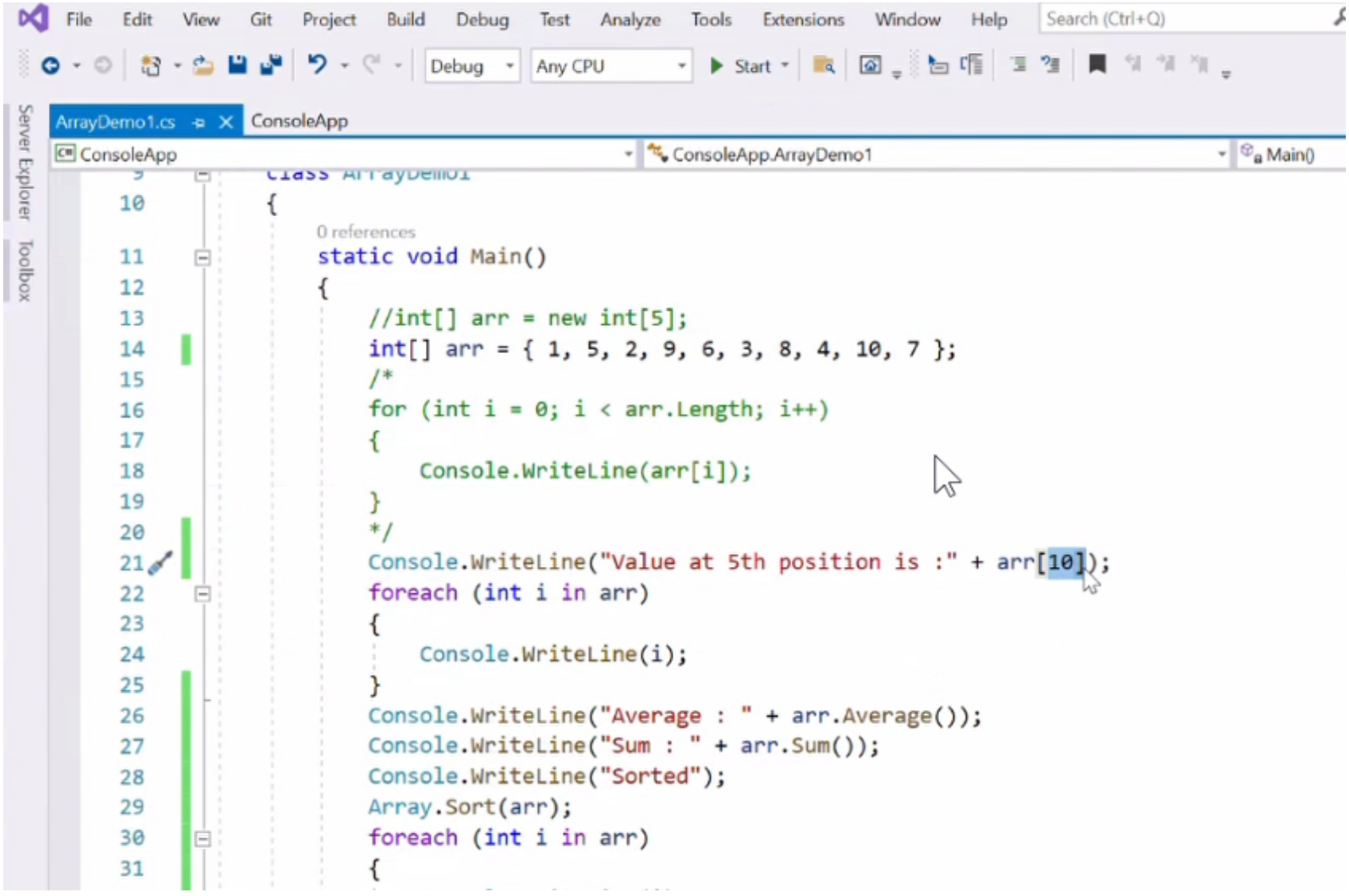
Checking Array Bounds

- All array access attempts are bounds checked
  - A bad index throws an IndexOutOfRangeException
  - Use the **Length** property and the **GetLength** method

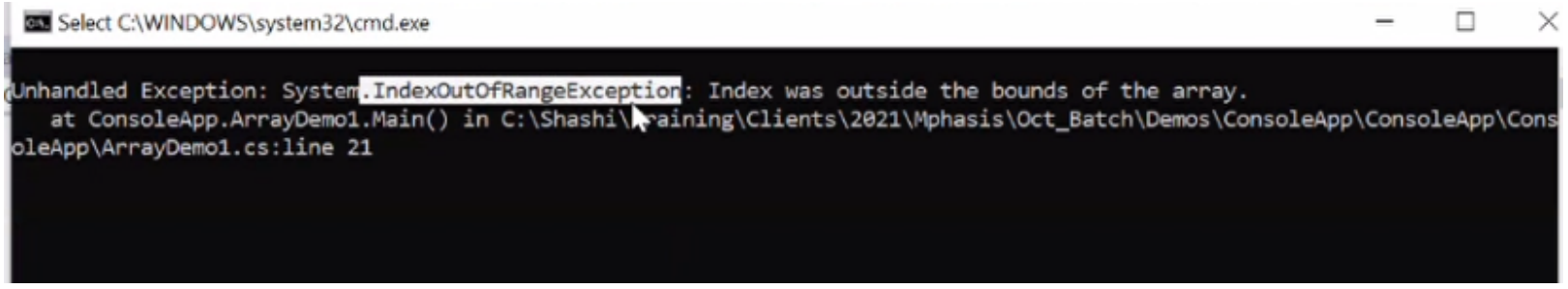


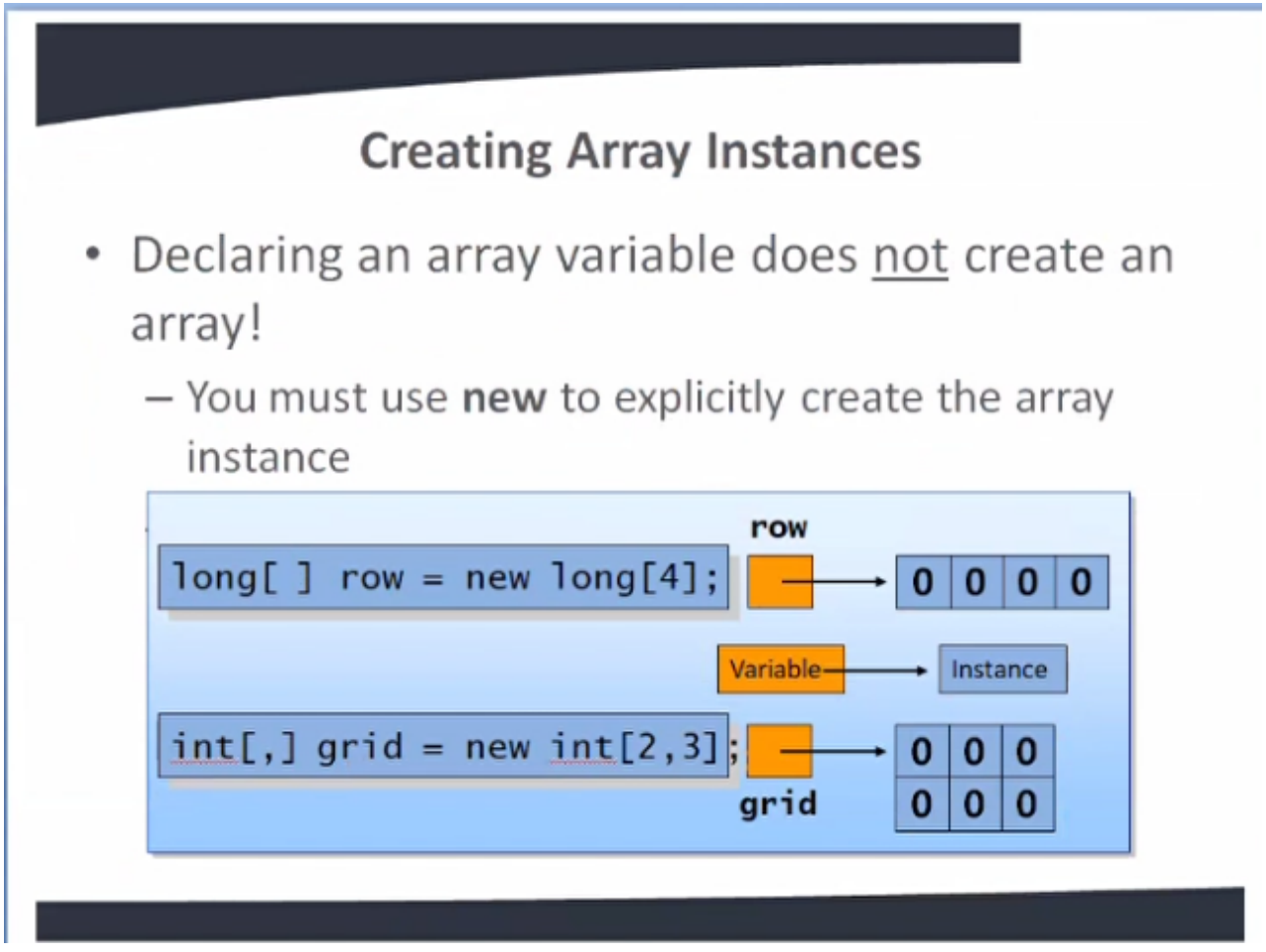

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Debug
Any CPU
Start
Server Explorer
Toolbox
ArrayDemo1.cs
ConsoleApp
ConsoleApp
ConsoleApp.ArrayDemo1
class ArrayDemo1
{
0 references
static void Main()
{
//int[] arr = new int[5];
int[] arr = { 1, 5, 2, 9, 6, 3, 8, 4, 10, 7 };
/*
for (int i = 0; i < arr.Length; i++)
{
Console.WriteLine(arr[i]);
}
*/
Console.WriteLine('Value at 5th position is :" + arr[5]);
foreach (int i in arr)
{
Console.WriteLine(i);
}
Console.WriteLine("Average : " + arr.Average());
Console.WriteLine("Sum : " + arr.Sum());
Console.WriteLine("Sorted");
Array.Sort(arr);
foreach (int i in arr)
{
121 %
4
0
Output
Package Manager Console
Error List ...
Immediate Window
Item(s) Saved


C:\WINDOWS\system32\cmd.exe
Value at 5th position is :3
1
5
2
9
6
3
8
4
10
7
Average : 5.5
Sum : 55
Sorted
1
2
3
4
5
6
7
8
9
10
Rank of array :2
Press any key to continue . . .

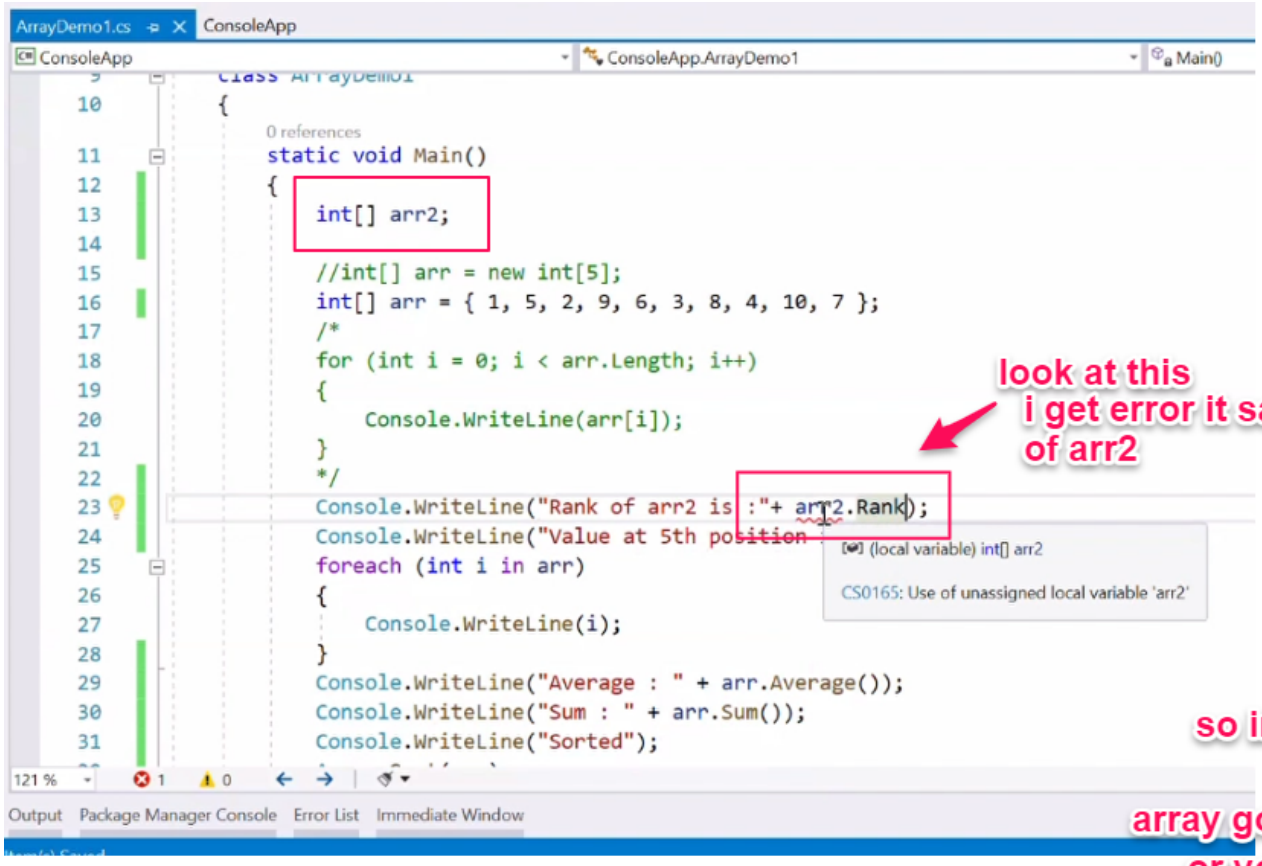


Now when i want to access array element at index 10 i'll get indexoutofrange exception





What do you mean declaring an array variable does not create an array



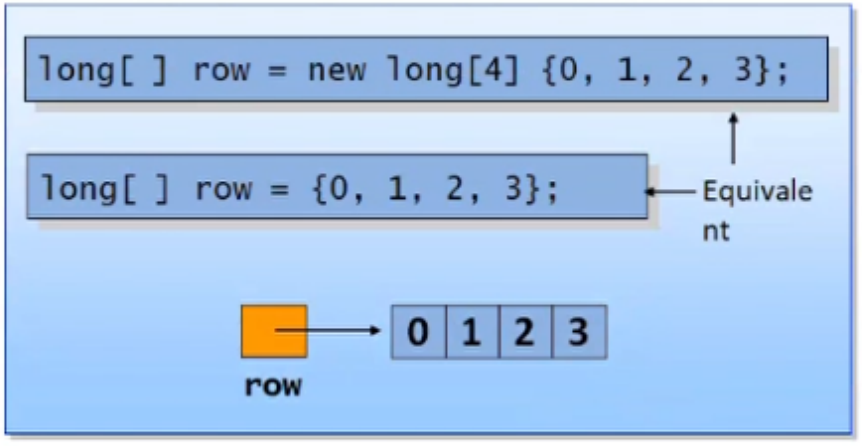
look at this
i get error it says unassigned local variable
of arr2

so int[] arr2 is just a declaration

you have not created a integer array
array got created only when you initialize them
or you say = new int[5(5 is the range of array)]

Initializing Array Elements

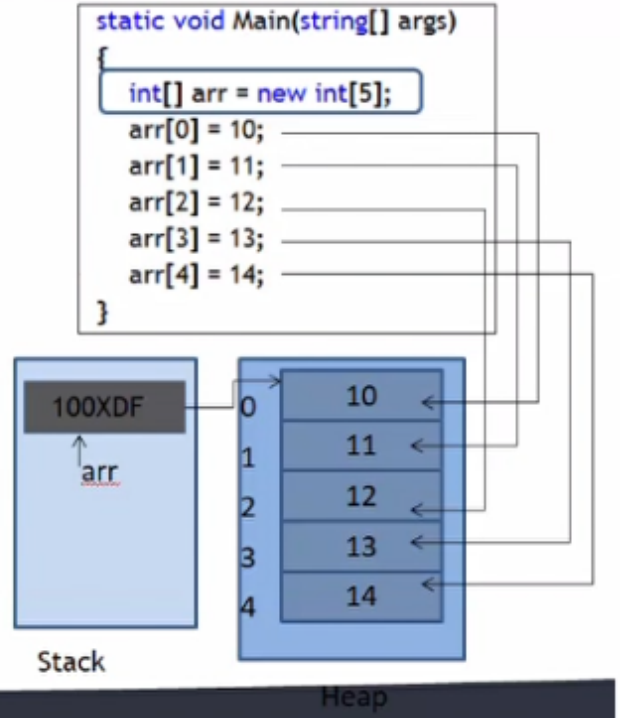
- The elements of an array can be explicitly initialized
  - You can use a convenient shorthand



Single dimensional Array

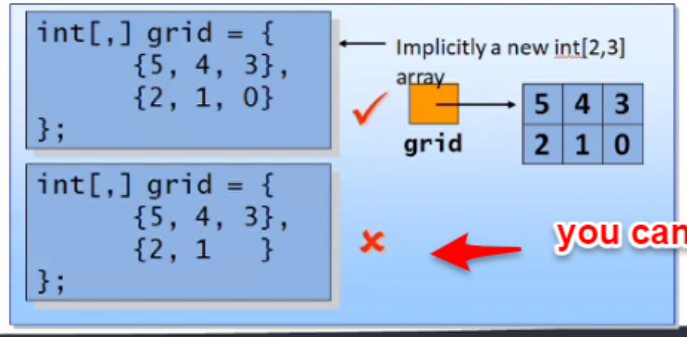
- Array containing a single row of values

- Example:
  - int[] numbers = new int[5] {1, 2, 3, 4, 5};
  - string[] names = new string[3] {"Matt", "Joanne", "Robert"};



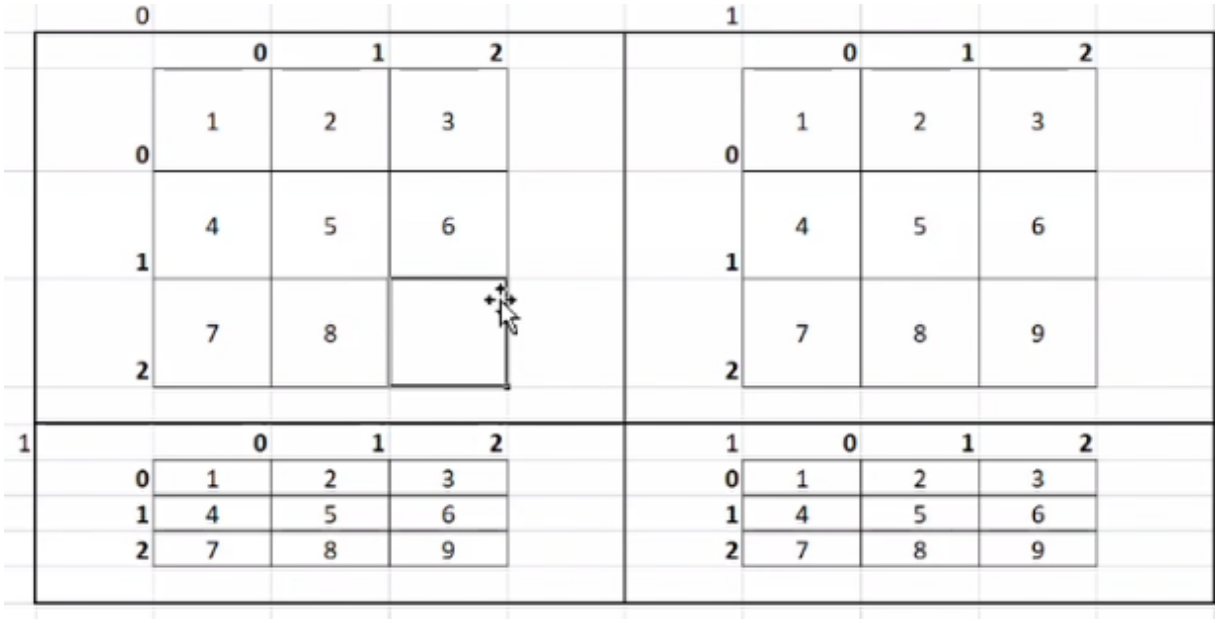
Initializing Multidimensional Array Elements

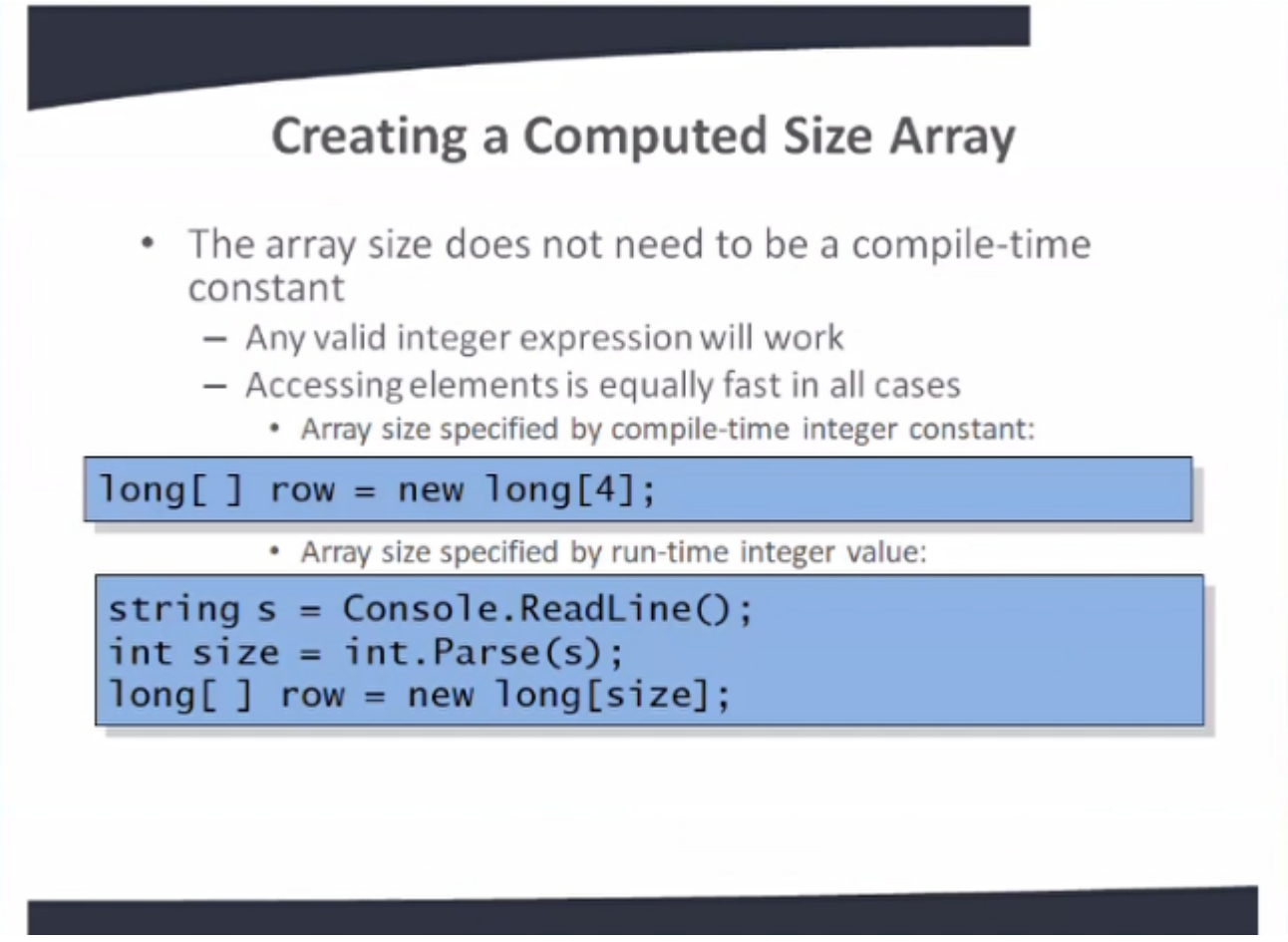
- Array containing multiple rows with same number of items for each row.
- You can also initialize multidimensional array elements
  - All elements must be specified



you cannot do like this, this will give you error

Why this is a error ?
because of mwmory allocation, you are not allocating the memory when you say 3 x3 so you should have 9 elements


0
0 1 2
0 1 2 3
1 4 5 6
2 7 8
1
0 1 2
0 1 2 3
1 4 5 6
2 7 8 9
1
0 1 2
0 1 2 3
1 4 5 6
2 7 8 9
1
0 1 2
0 1 2 3
1 4 5 6
2 7 8 9

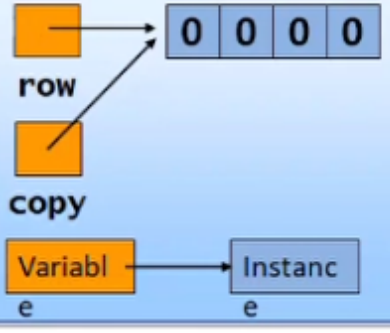


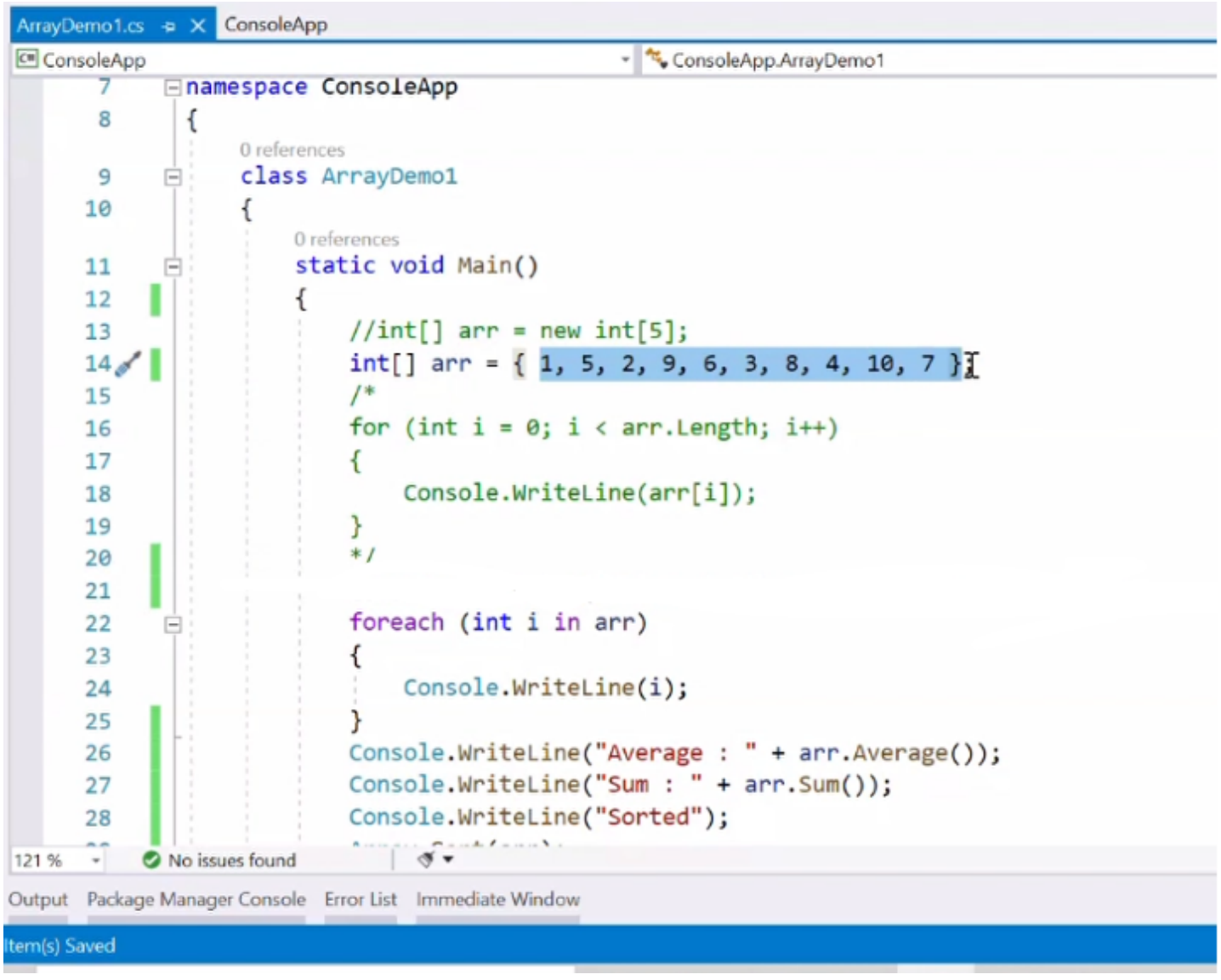
you can compute the array size based on particular array , may be you can ask user to give a particular array length and based on that you can pass the particular size as a parameter to your array that is possible

Copying Array Variables

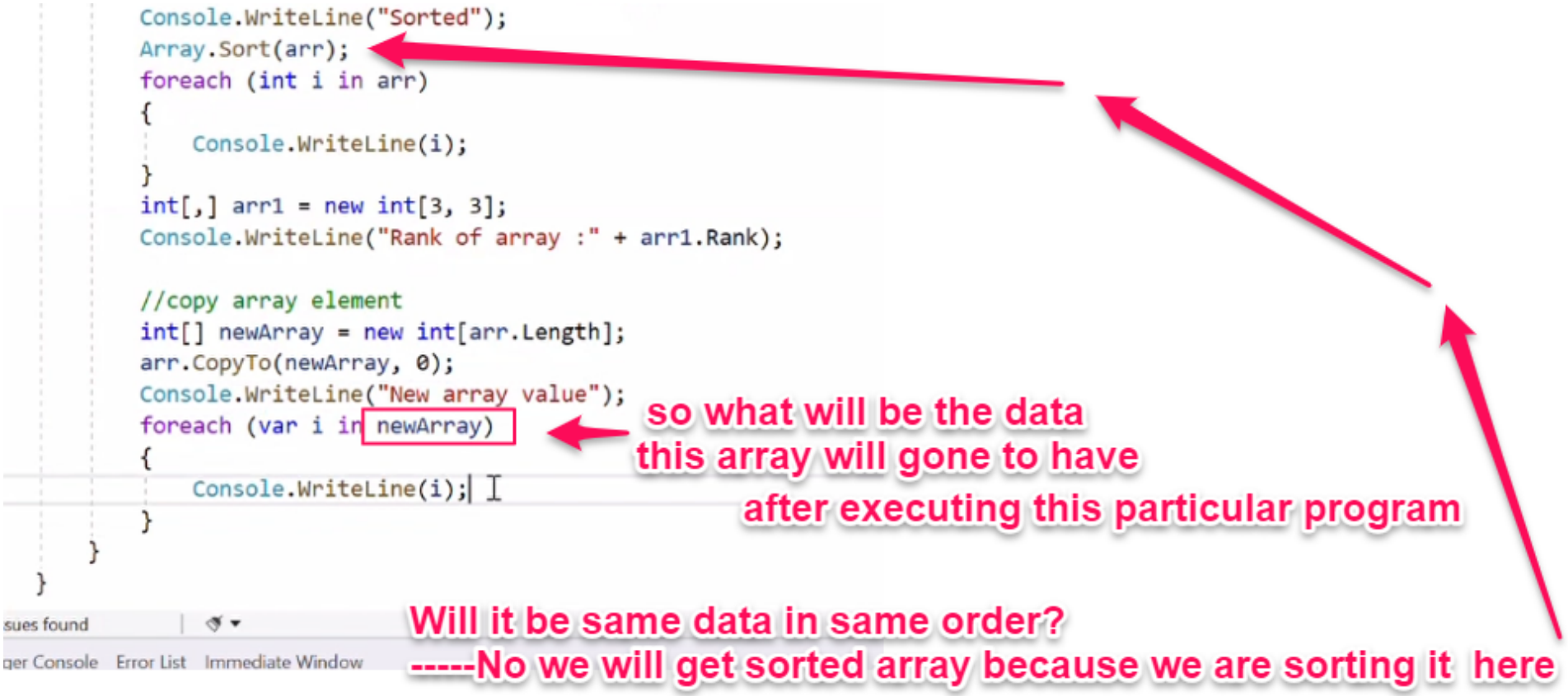
- Copying an array variable copies the array variable only
  - It does not copy the array instance
  - Two array variables can refer to the same array instance

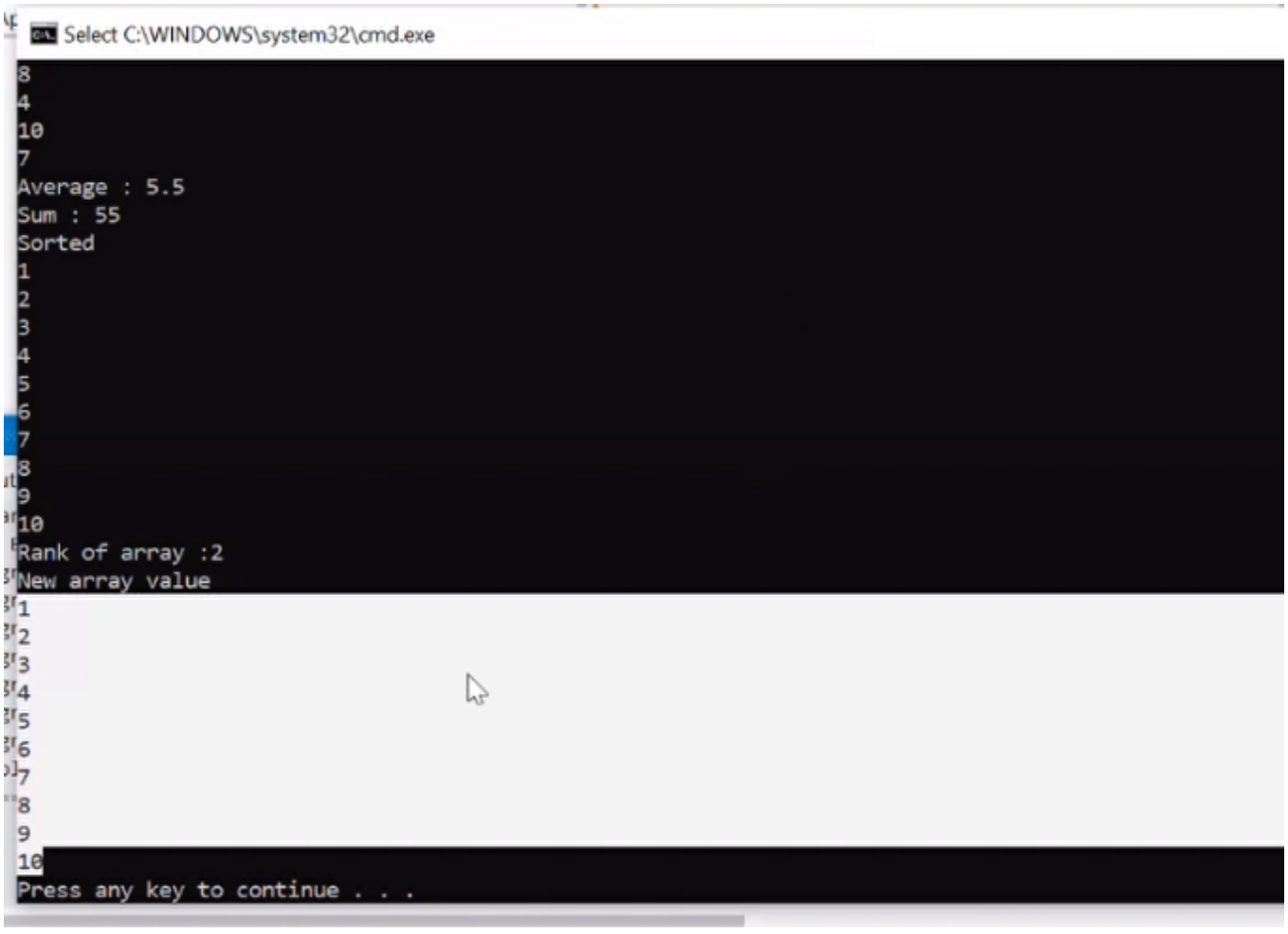
```
long[ ] row = new long[4];
long[ ] copy = row;
...
row[0]++;
long value = copy[0];
Console.WriteLine(value);
```

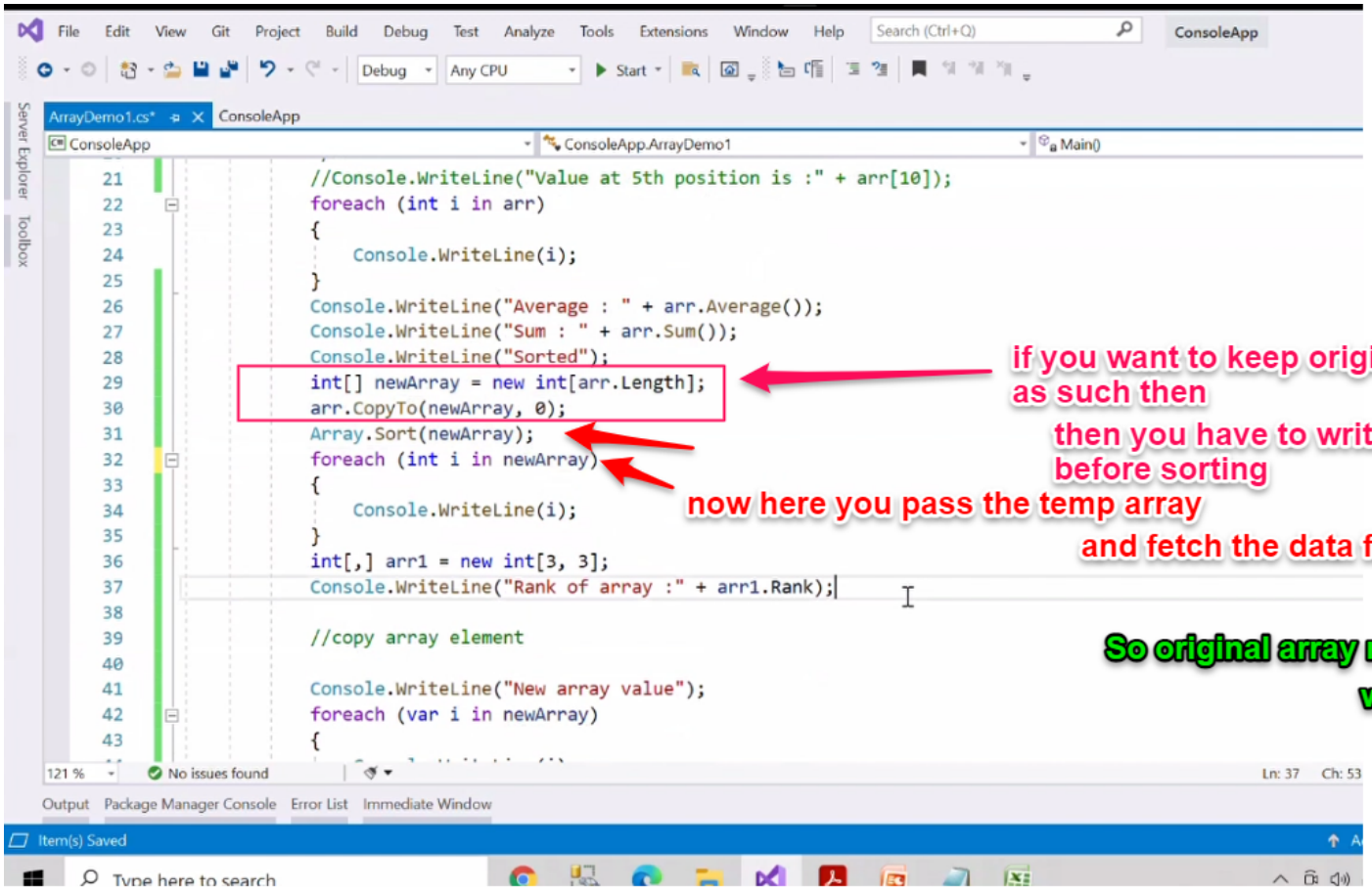




Now original array is having this data 1 5 2 9 6 3 8 4 10 7


Console.WriteLine("Sorted");
Array.Sort(arr);
foreach (int i in arr)
{
Console.WriteLine(i);
}
int[,] arr1 = new int[3, 3];
Console.WriteLine("Rank of array :" + arr1.Rank);
//copy array element
int[] newArray = new int[arr.Length];
arr.CopyTo(newArray, 0);
Console.WriteLine("New array value");
foreach (var i in newArray)
{
Console.WriteLine(i);
}
}
}
so what will be the data
this array will gone to have
after executing this particular program
sues found
ger Console
Error List
Immediate Window
Will it be same data in same order?
-----No we will get sorted array because we are sorting it here


Select C:\WINDOWS\system32\cmd.exe
8
4
10
7
Average : 5.5
Sum : 55
Sorted
1
2
3
4
5
6
7
8
9
10
Rank of array :2
New array value
1
2
3
4
5
6
7
8
9
10
Press any key to continue . . .



if you want to keep original array as such then

then you have to write this particular line here before sorting

now here you pass the temp array

and fetch the data from the temp array

So original array remains as such

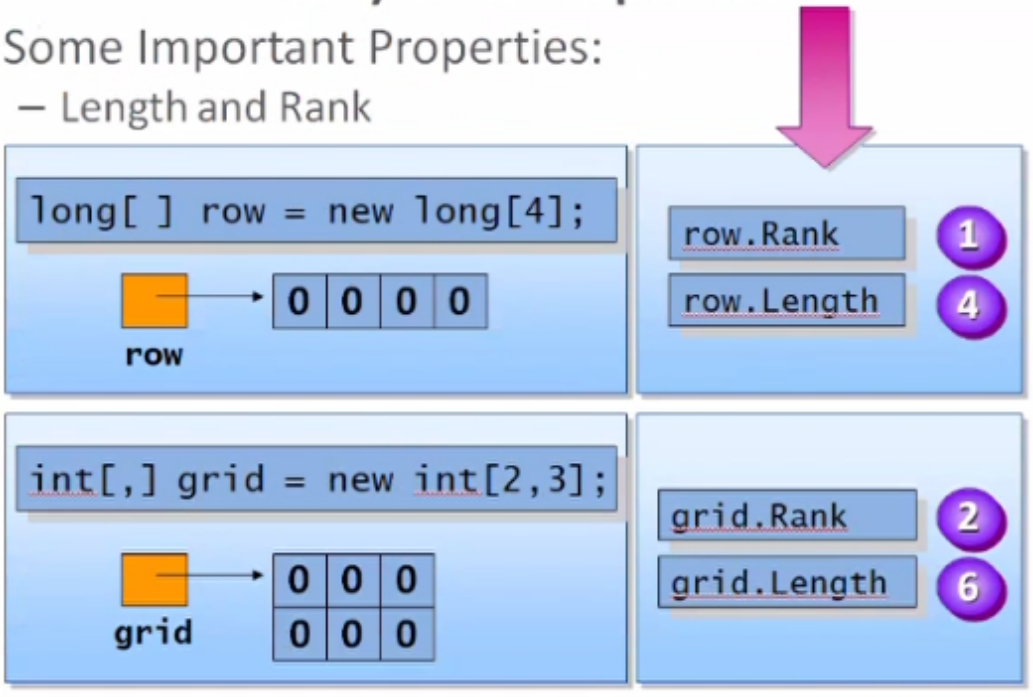
we are not touching the original array

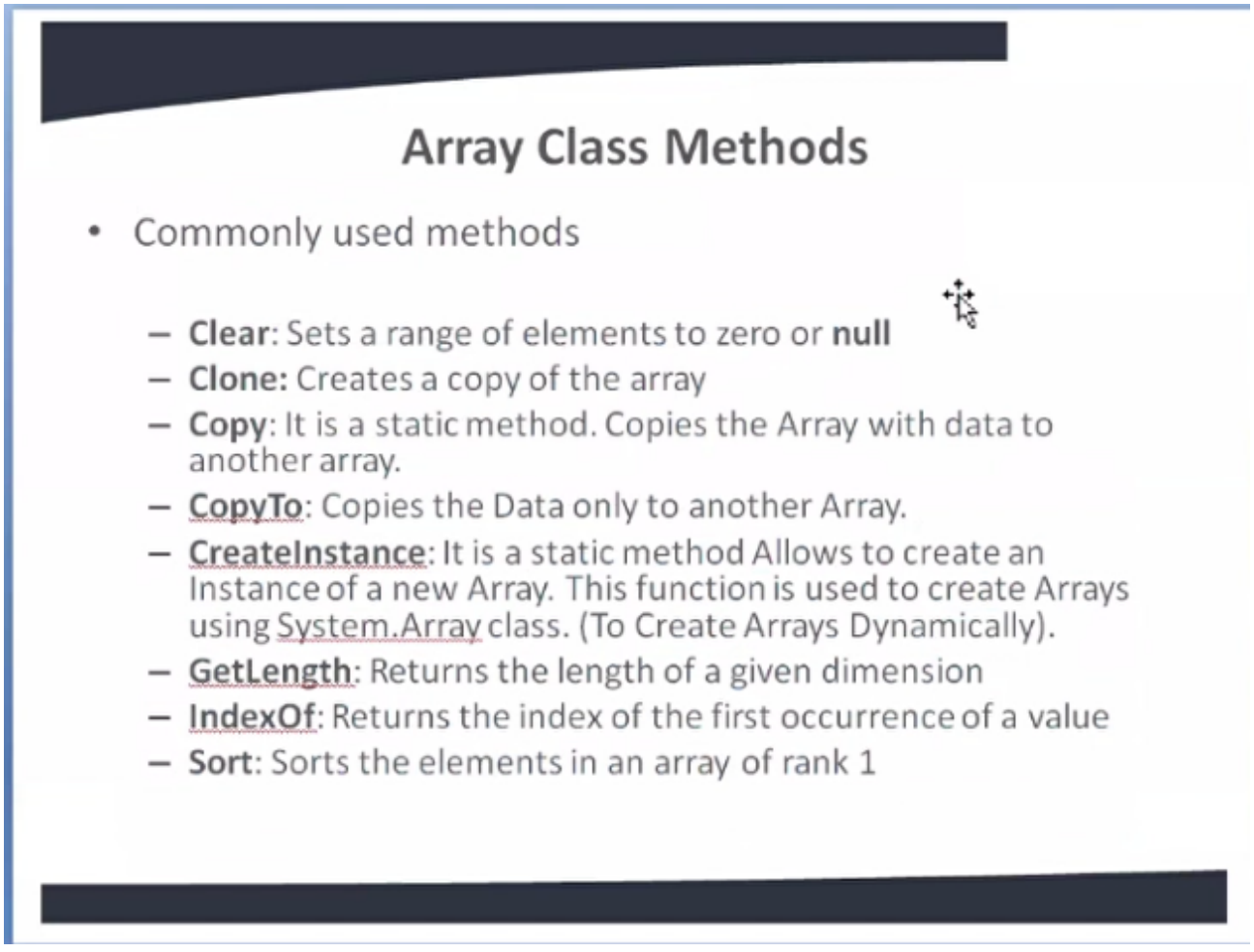
what ever the maupulatios you are doing is always on the temp array the newArray

thats how your original data will not get spoiled

Array Class Properties

- Some Important Properties:
  - Length and Rank





Clone - creates a new array copy will copy the statical data

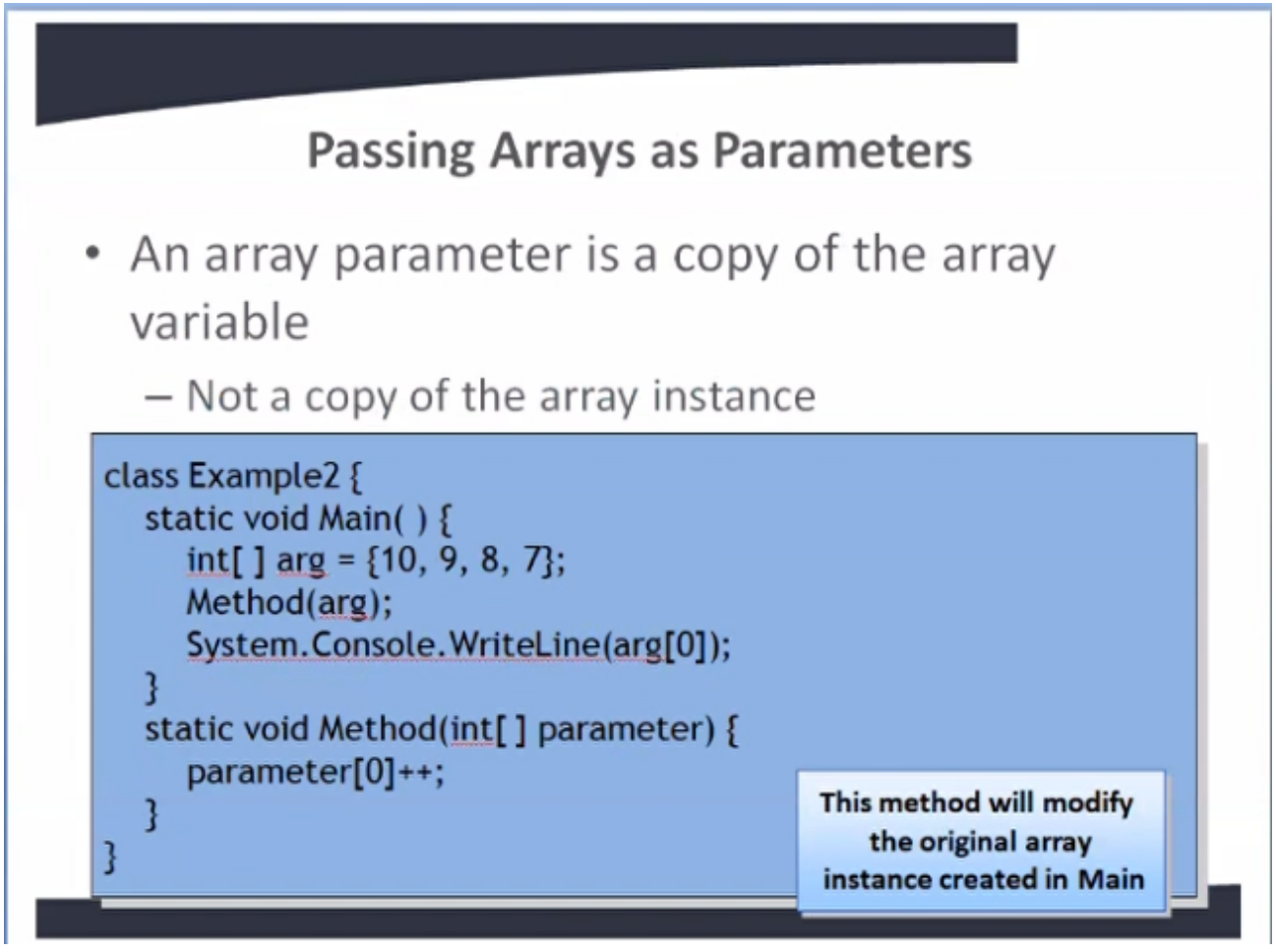
Copy - will copy the statical data

DO reading on shallow copy and deep copy

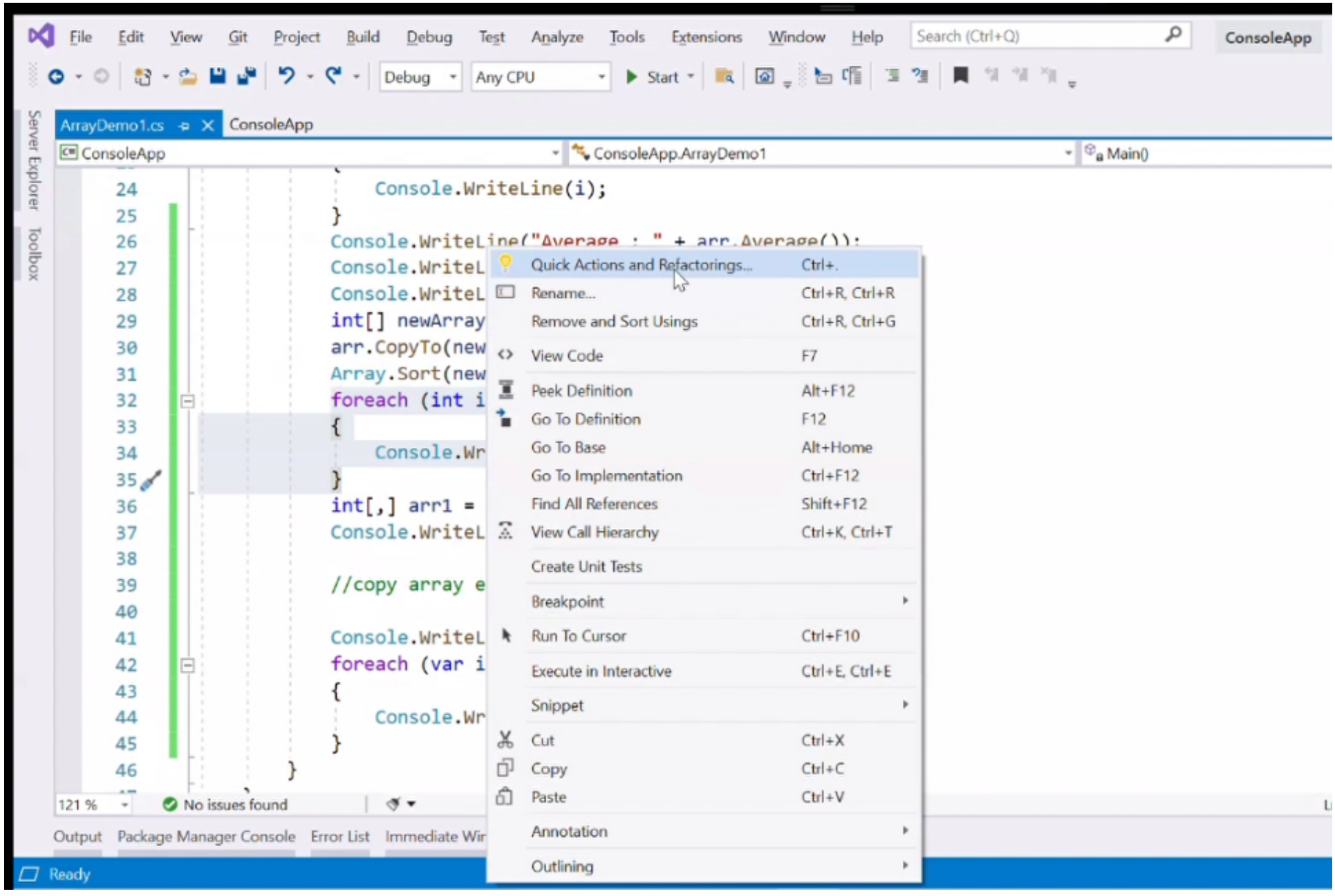
Returning Arrays from Methods

- You can declare methods to return arrays

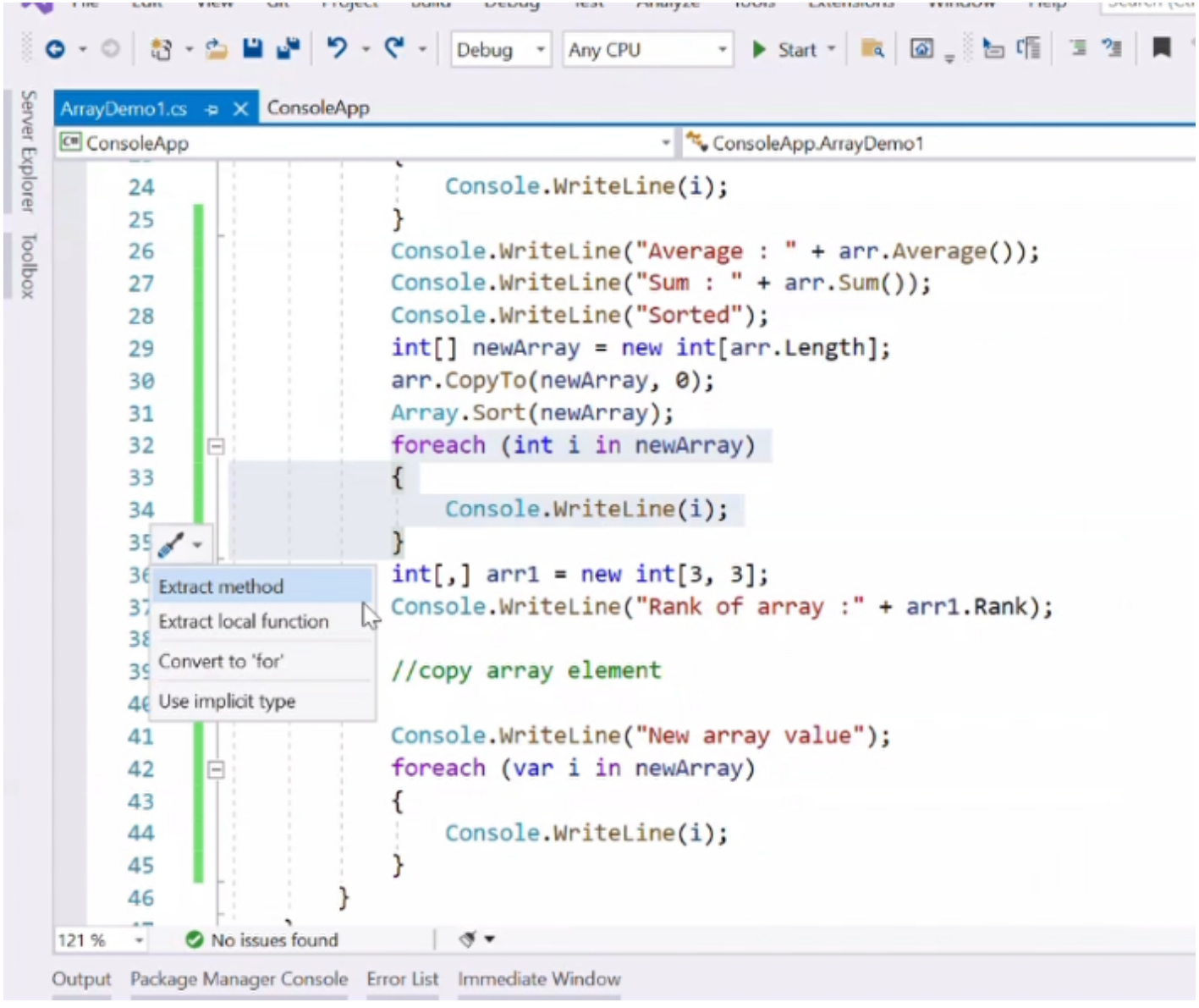
```
class Example {
    static void Main( ) {
        int[ ] array = CreateArray(42);
        ...
    }
    static int[ ] CreateArray(int size) {
        int[ ] created = new int[size];
        return created;
    }
}
```



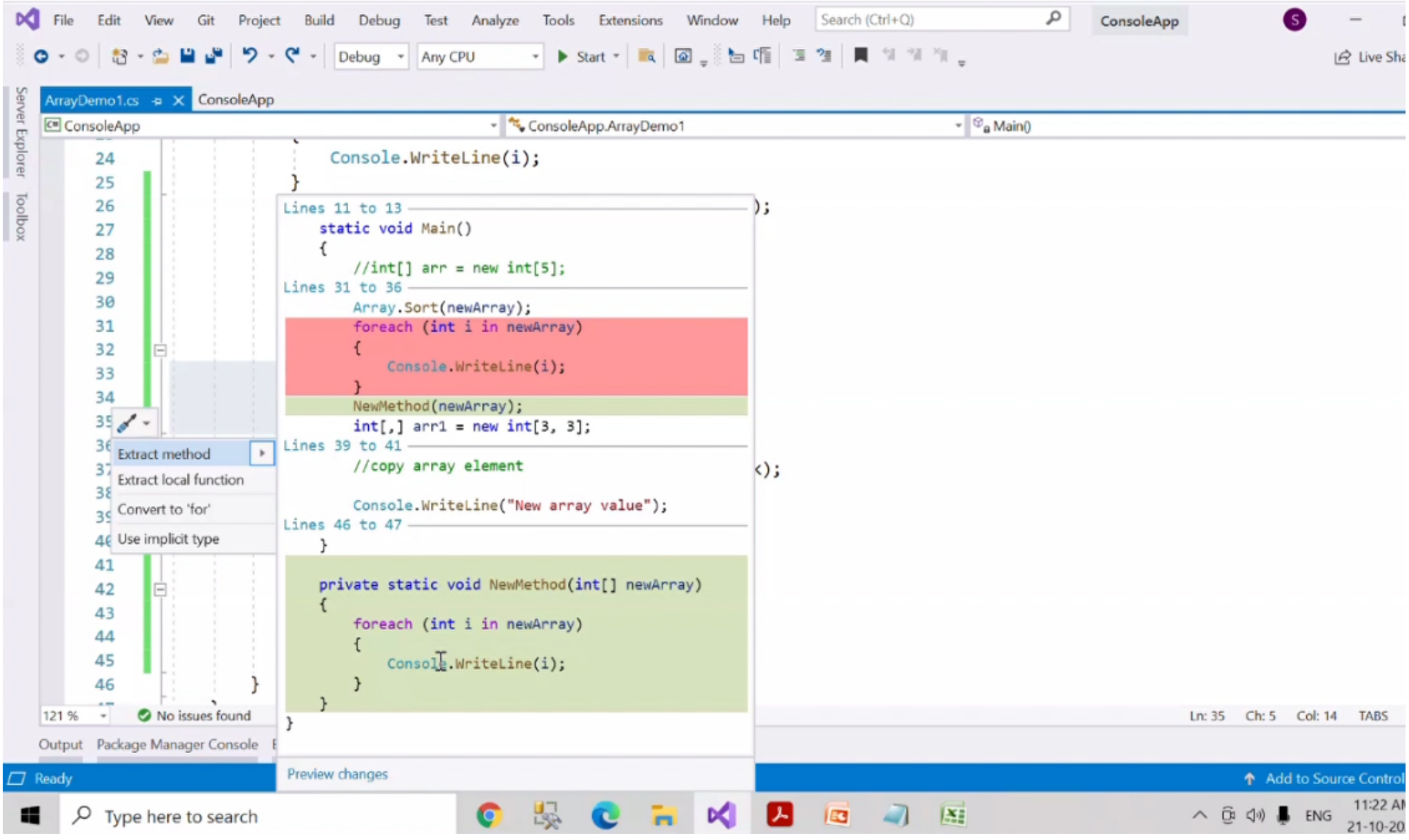
we'll gone to push this printing code outside this method ,
now when you have some piece of code which you want to push outside the method , so instead of cutting from here and pasting it,
you can always use refactoring


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
ConsoleApp
Debug
Any CPU
Start
ArrayDemo1.cs
ConsoleApp
ConsoleApp.ArrayDemo1
Main()
Server Explorer
Toolbox
Console.WriteLine(i);
}
Console.WriteLine("Average : " + arr.Average());
Console.WriteL
Console.WriteL
int[] newArray
arr.CopyTo(new
Array.Sort(new
foreach (int i
{
Console.Wr
}
int[,] arr1 =
Console.WriteL
//copy array e
Console.WriteL
foreach (var i
{
Console.Wr
}
}
Quick Actions and Refactorings... Ctrl+.
Rename... Ctrl+R, Ctrl+R
Remove and Sort Usings Ctrl+R, Ctrl+G
View Code F7
Peek Definition Alt+F12
Go To Definition F12
Go To Base Alt+Home
Go To Implementation Ctrl+F12
Find All References Shift+F12
View Call Hierarchy Ctrl+K, Ctrl+T
Create Unit Tests
Breakpoint
Run To Cursor Ctrl+F10
Execute in Interactive Ctrl+E, Ctrl+E
Snippet
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Annotation
Outlining
121 %
No issues found
Output
Package Manager Console
Error List
Immediate Wir
Ready

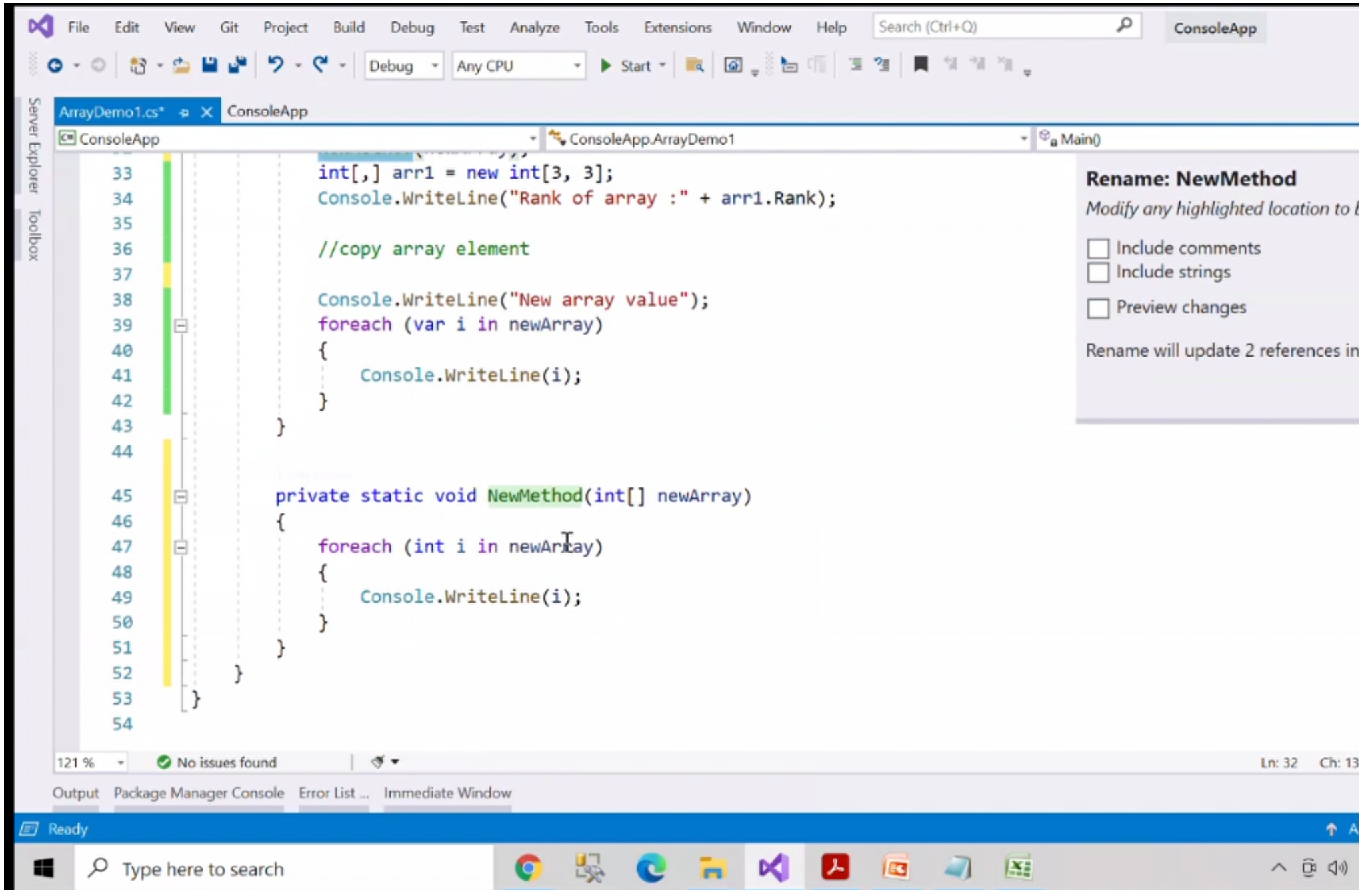
select code which you want to move from one method to another method ,right click on it , click on quick action and refactoring .., click on extract method

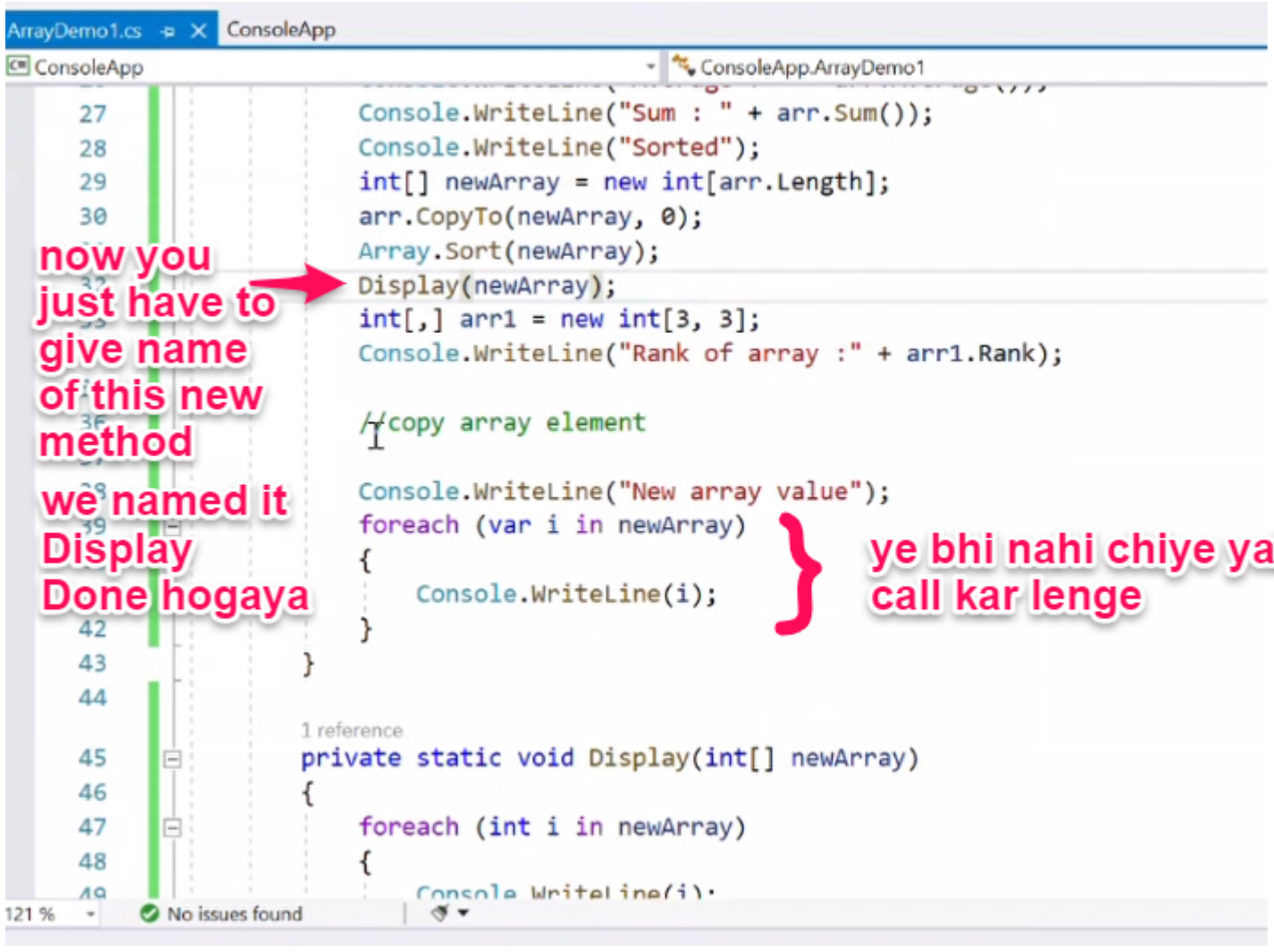


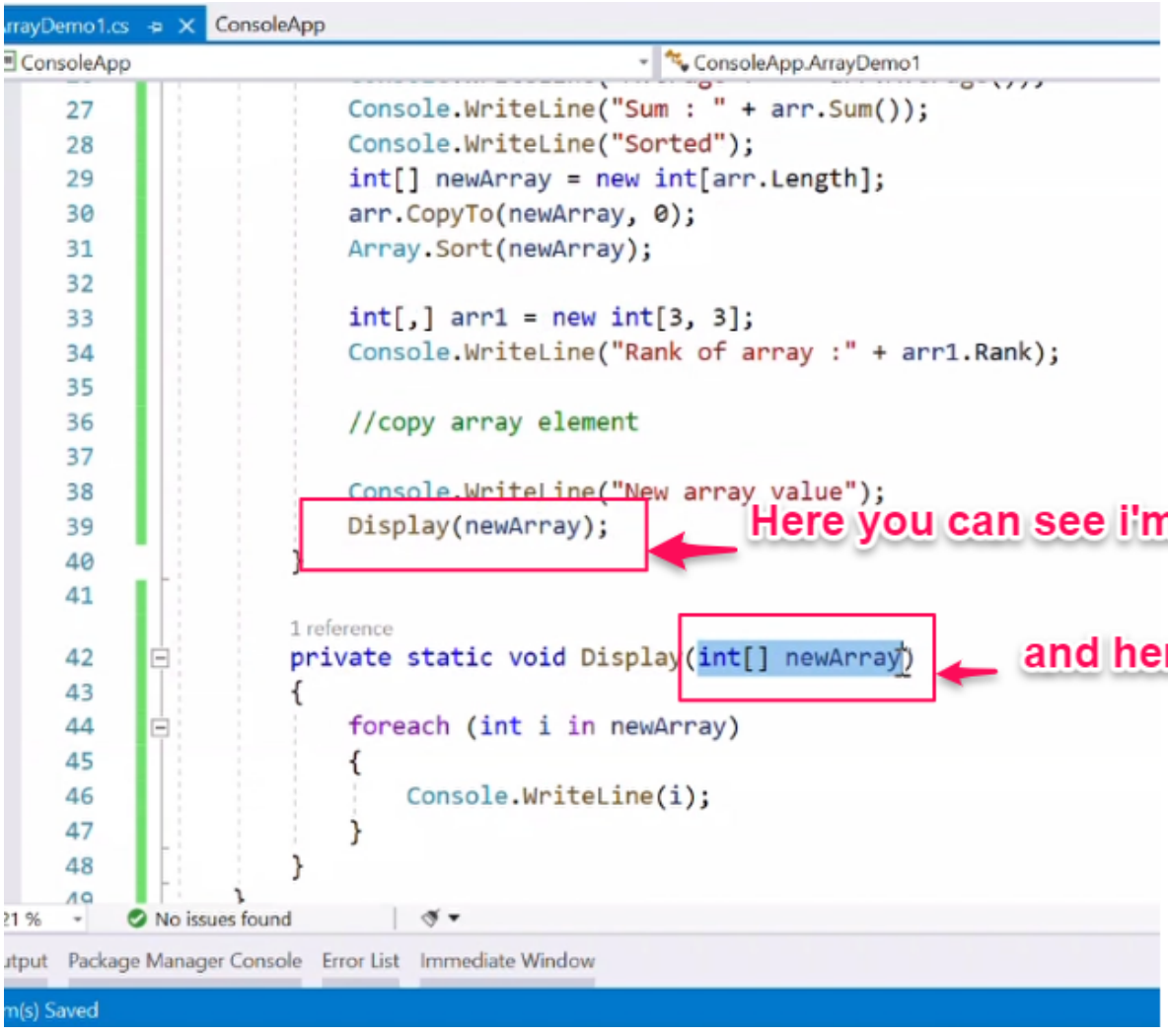
you can see it in preview



just click on this and a new method is created for you


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
ConsoleApp
Debug
Any CPU
Start
ArrayDemo1.cs*
ConsoleApp
ConsoleApp.ArrayDemo1
Main()
Server Explorer
Toolbox
int[,] arr1 = new int[3, 3];
Console.WriteLine("Rank of array :" + arr1.Rank);
//copy array element
Console.WriteLine("New array value");
foreach (var i in newArray)
{
Console.WriteLine(i);
}
}
private static void NewMethod(int[] newArray)
{
foreach (int i in newArray)
{
Console.WriteLine(i);
}
}
}
}
Rename: NewMethod
Modify any highlighted location to
Include comments
Include strings
Preview changes
Rename will update 2 references in
121 %
No issues found
Ln: 32
Ch: 13
Output
Package Manager Console
Error List ...
Immediate Window
Ready
Type here to search

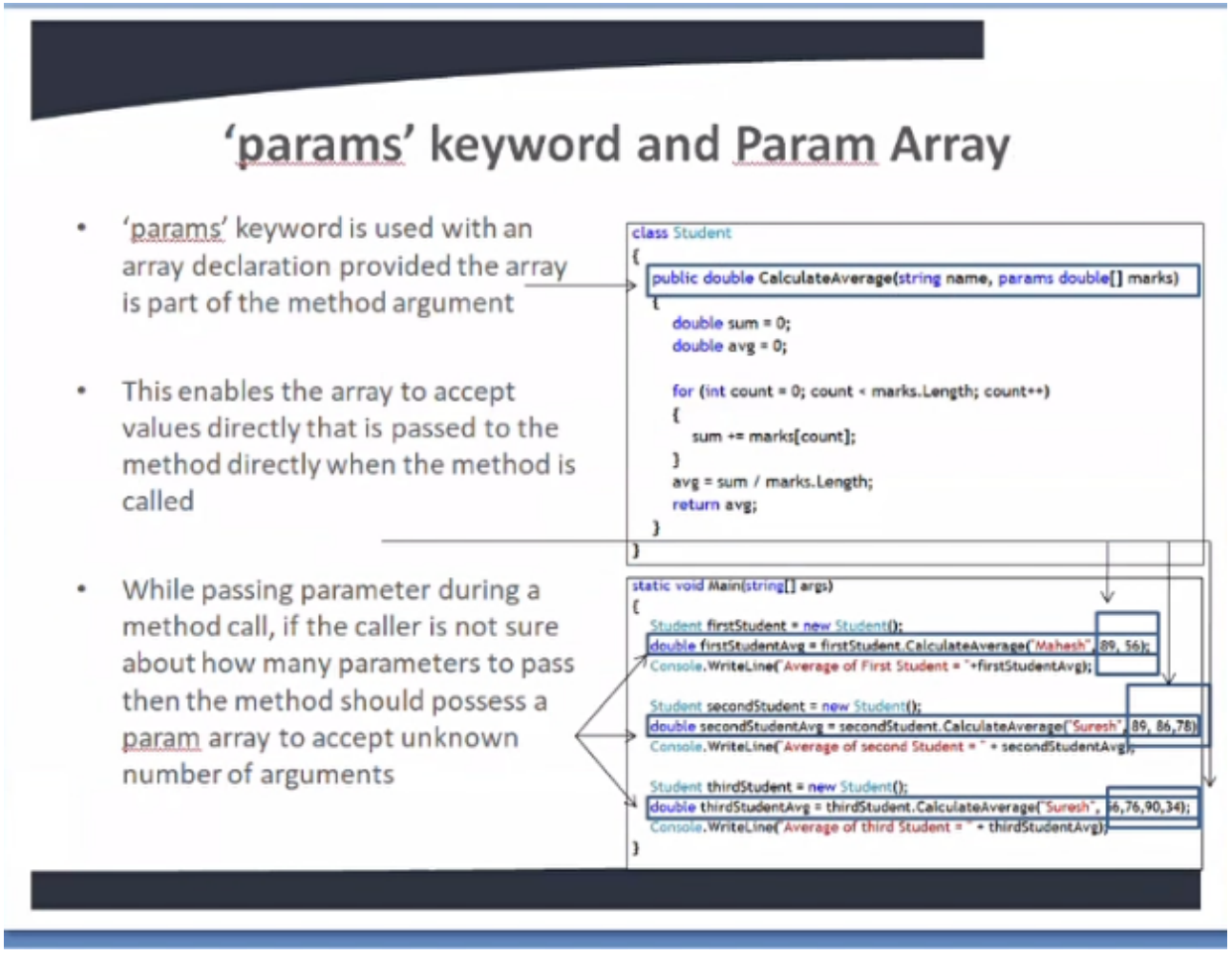

ArrayDemo1.cs
ConsoleApp
ConsoleApp
ConsoleApp.ArrayDemo1
Console.WriteLine("Sum : " + arr.Sum());
Console.WriteLine("Sorted");
int[] newArray = new int[arr.Length];
arr.CopyTo(newArray, 0);
Array.Sort(newArray);
Display(newArray);
int[,] arr1 = new int[3, 3];
Console.WriteLine("Rank of array :" + arr1.Rank);
//copy array element
Console.WriteLine("New array value");
foreach (var i in newArray)
{
Console.WriteLine(i);
}
}
1 reference
private static void Display(int[] newArray)
{
foreach (int i in newArray)
{
121 %
No issues found
now you just have to give name of this new method we named it Display Done hogaya
ye bhi nahi chiye yaha bhi hum display ko call kar lenge



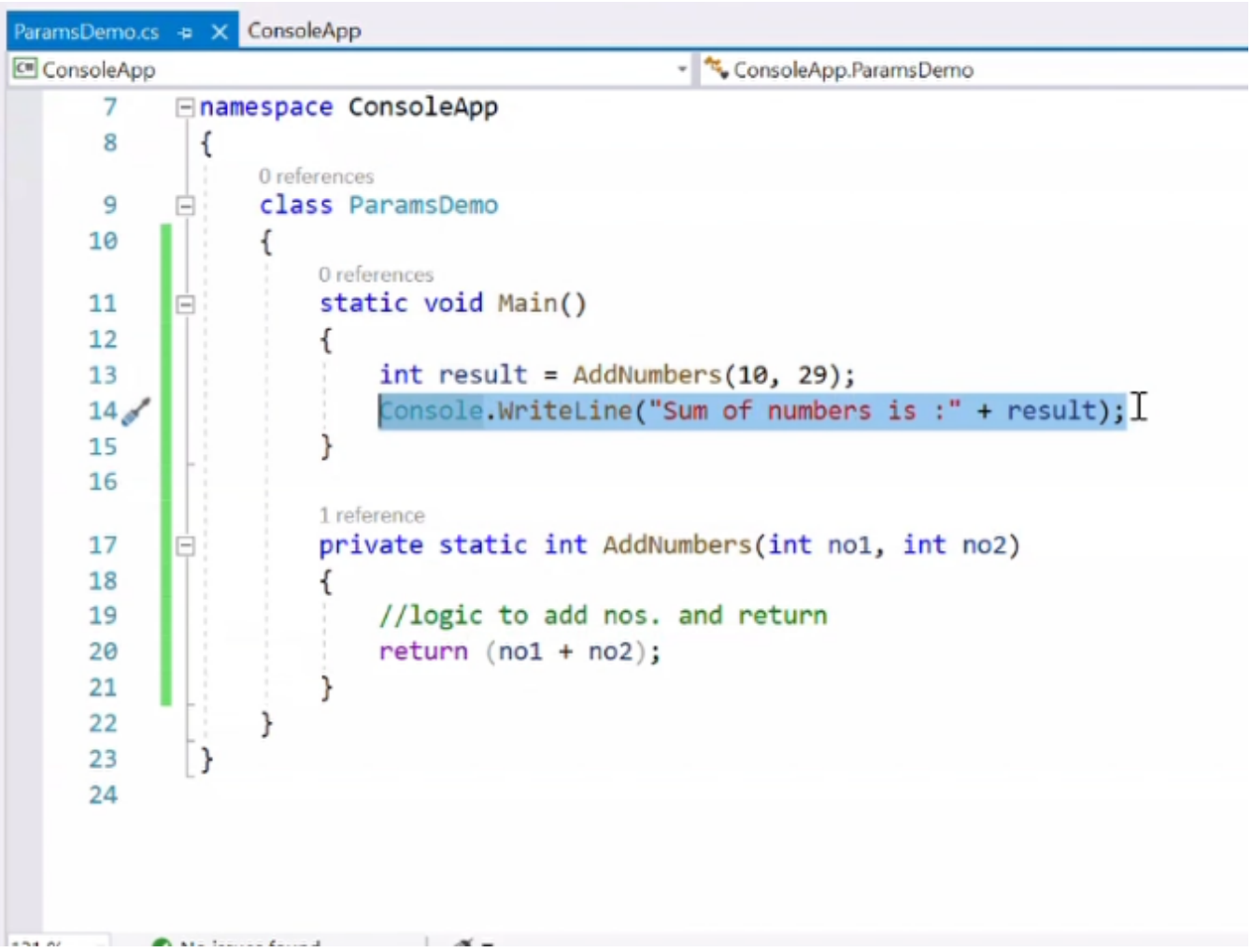
Here you can see i'm passing an array

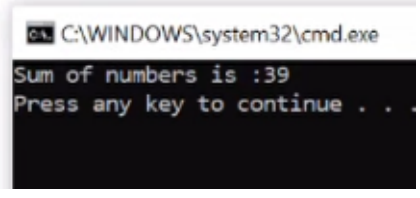
and here we are taking an array

This is how we can pass array from one place to another place

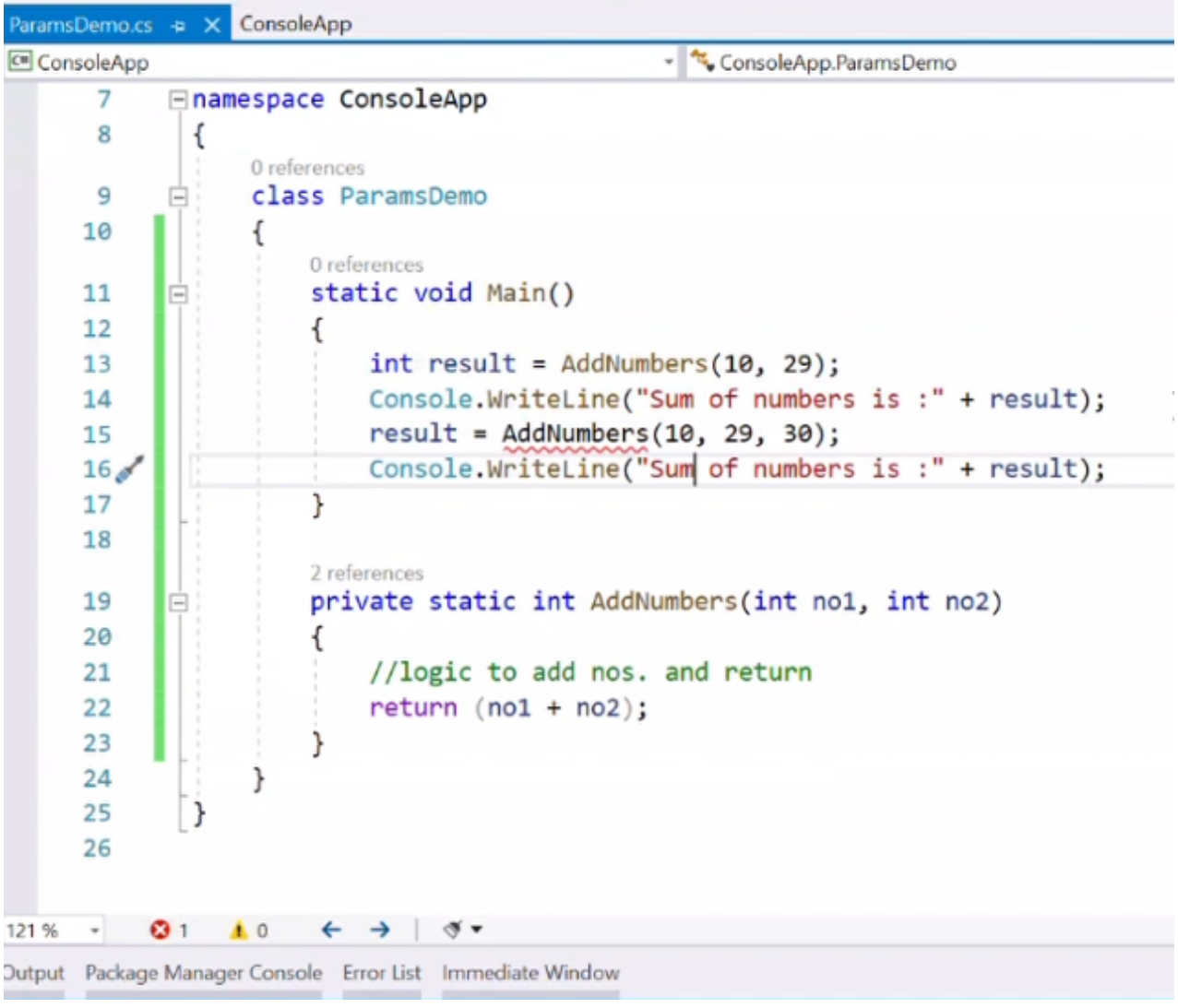


Imagine i have written a particular program and what this program does is, it will return a integer value , minimum how many numbers you can add-----alteast 2

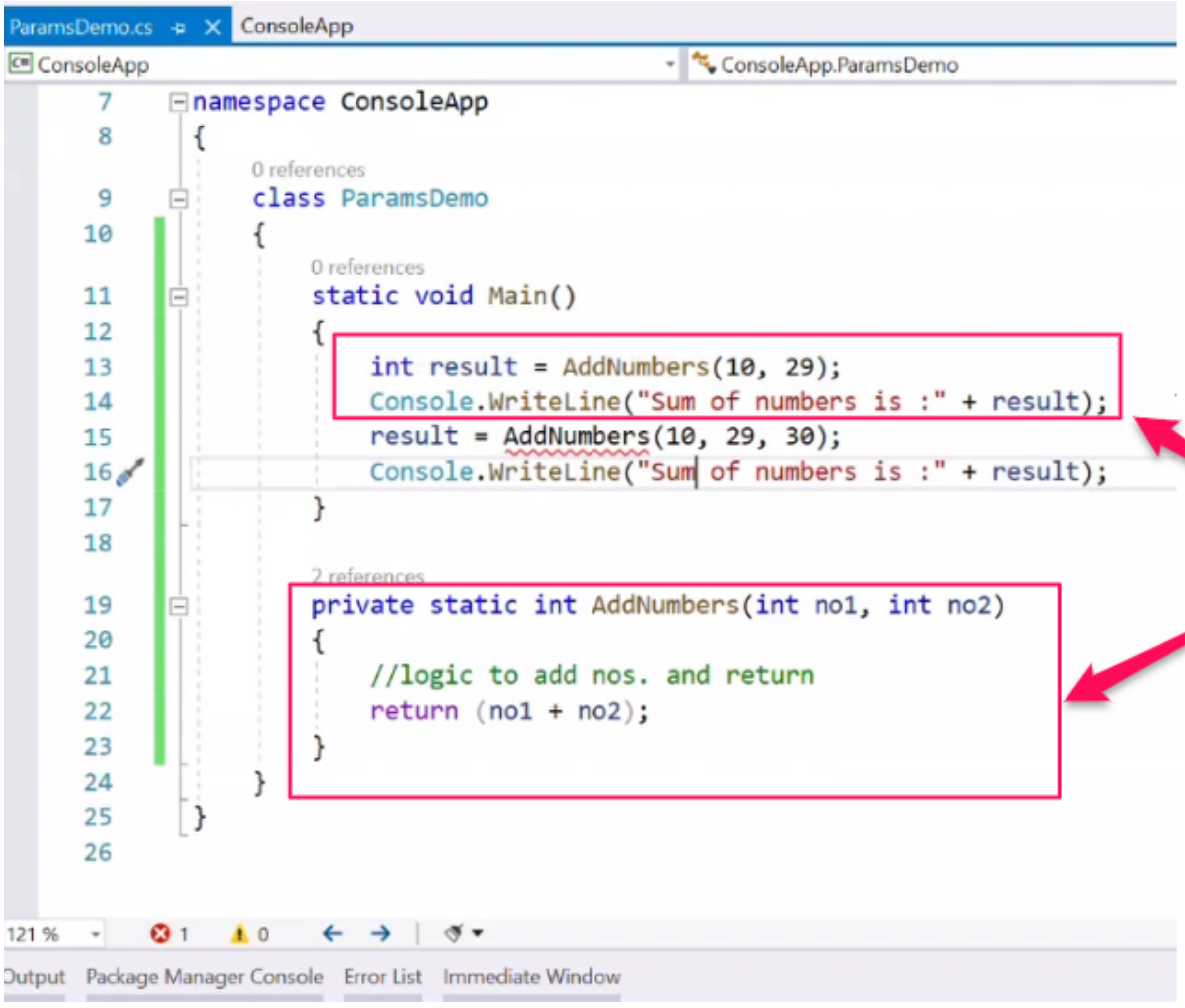

ParamsDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.ParamsDemo
namespace ConsoleApp
{
0 references
class ParamsDemo
{
0 references
static void Main()
{
int result = AddNumbers(10, 29);
Console.WriteLine("Sum of numbers is :" + result);
}
1 reference
private static int AddNumbers(int no1, int no2)
{
//logic to add nos. and return
return (no1 + no2);
}
}
}


C:\WINDOWS\system32\cmd.exe
Sum of numbers is :39
Press any key to continue . . .

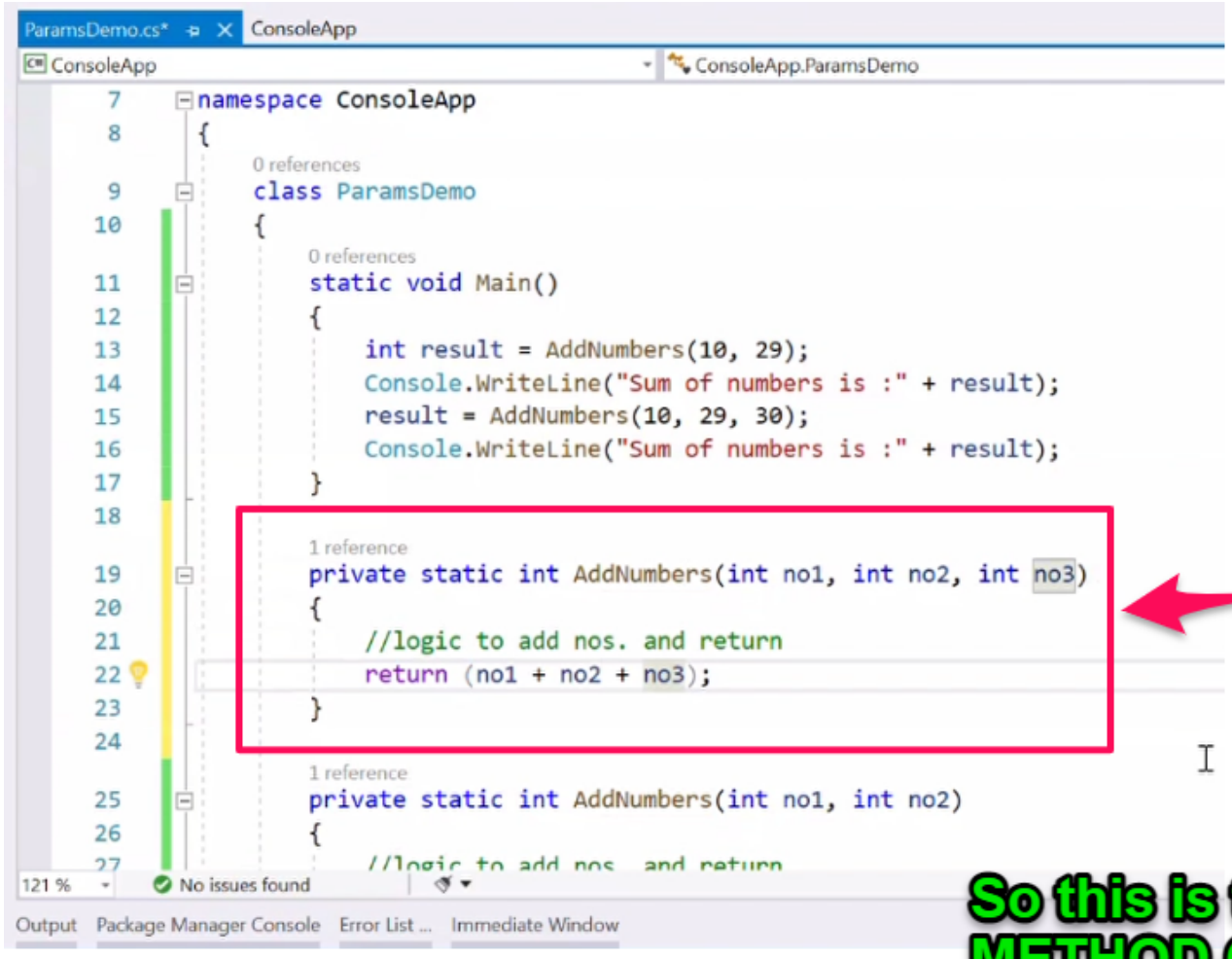
So this code is working perfectly fine, now the bussiness will come and say this is fabulos but i want to add more numbers, i want to add 3 numbers



So how do we do this ?



Now i cannot touch this code because these is already been implemented and already in production



So I need to add one more number here

So this is the concept of METHOD OVERLOADING

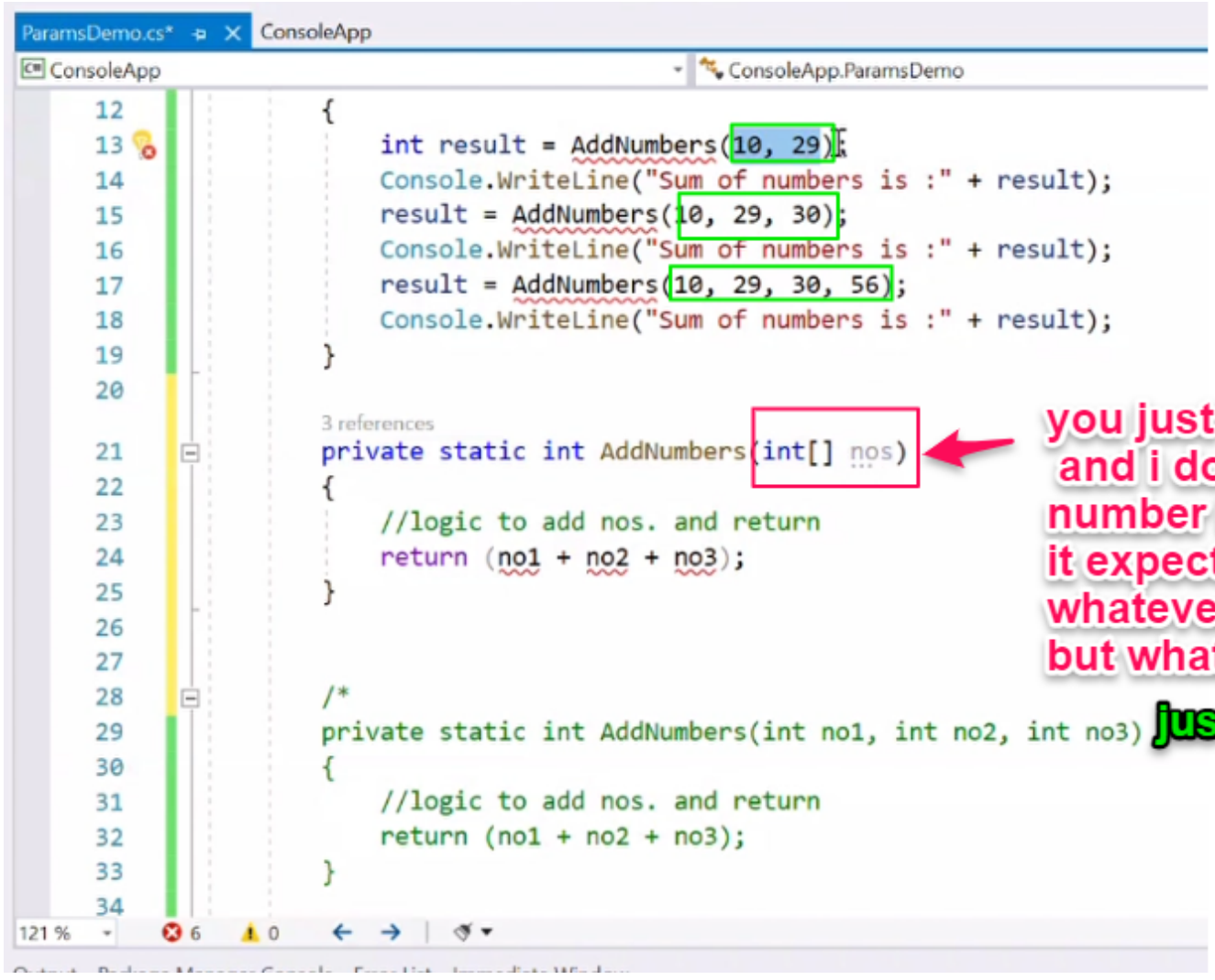
SO this is the concept of

overloading

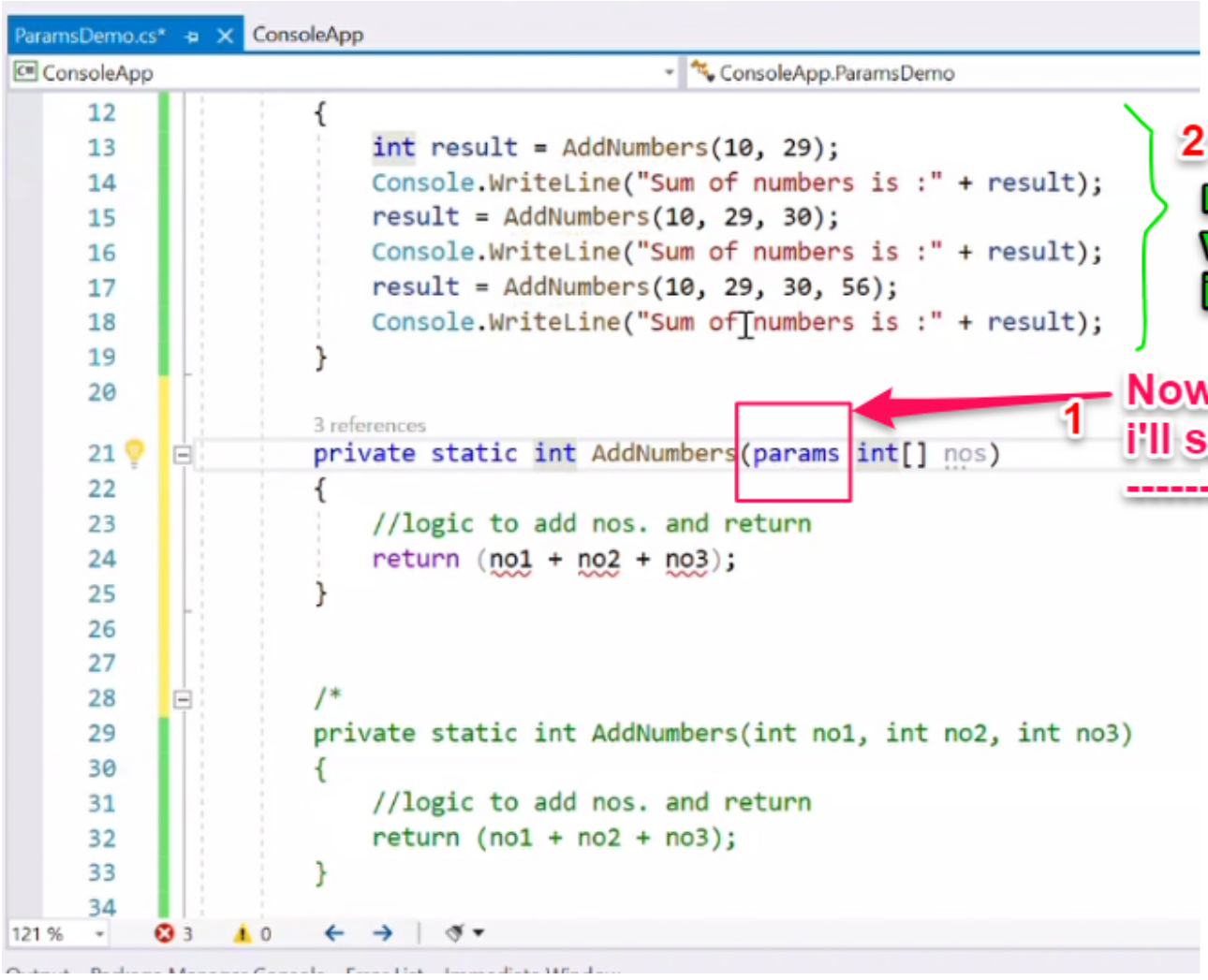
you can have the same method name but with different parameter

So, this is working fine now bussiness will say i want to add 4 numbers ,so you will do the same thing again

So, looking at this kind of code, developers of .NET thought -----boss whenever you have this kind of situation, you just have to do a shortcut



you just say, this is my array
and i don't know how many
number i'll get, now when i do this
it expects to be an array
whatever i'm getting
but what they are passing is
just a numbers



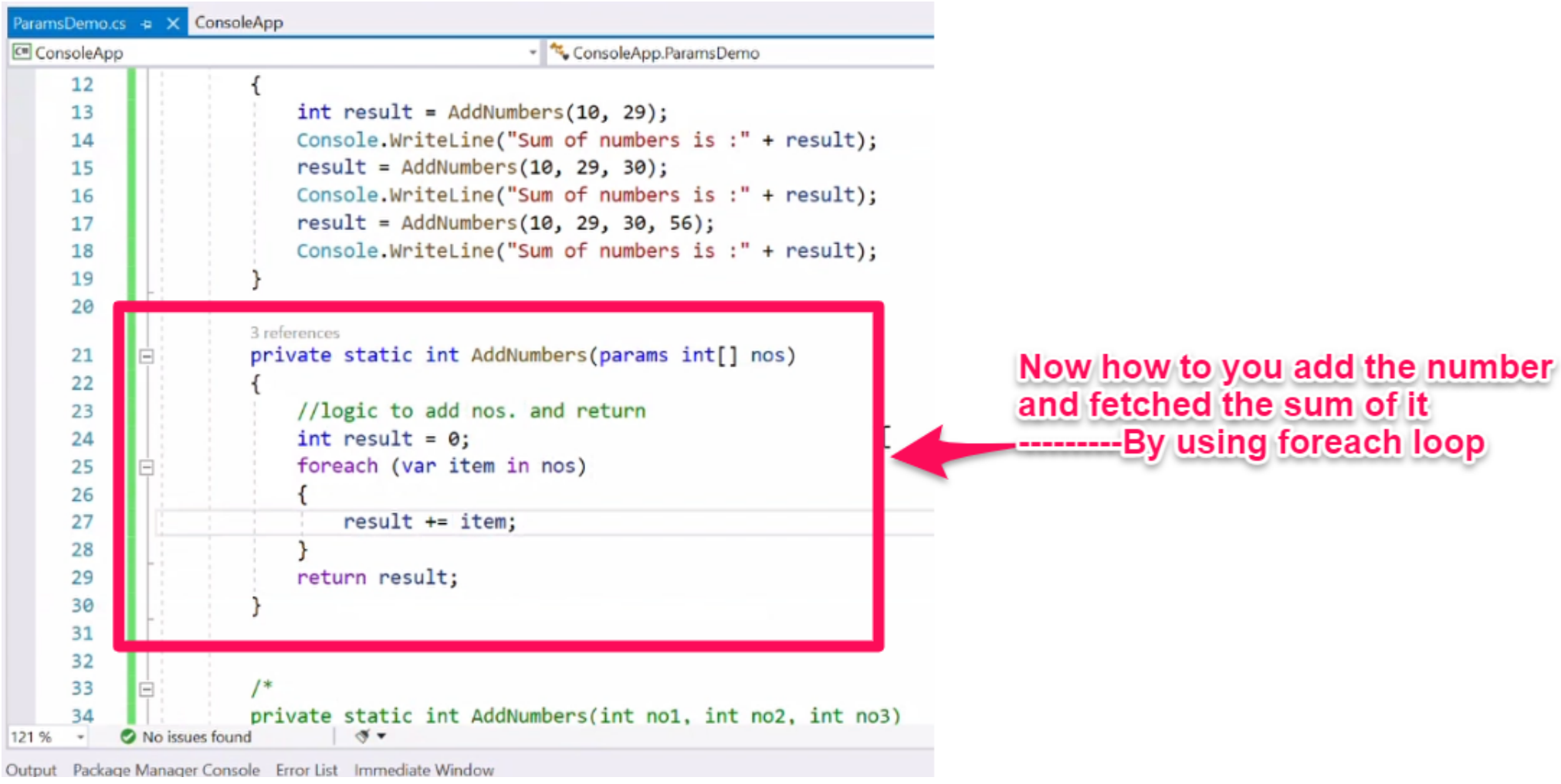
2

now look at this all the error
what i has here
is gone

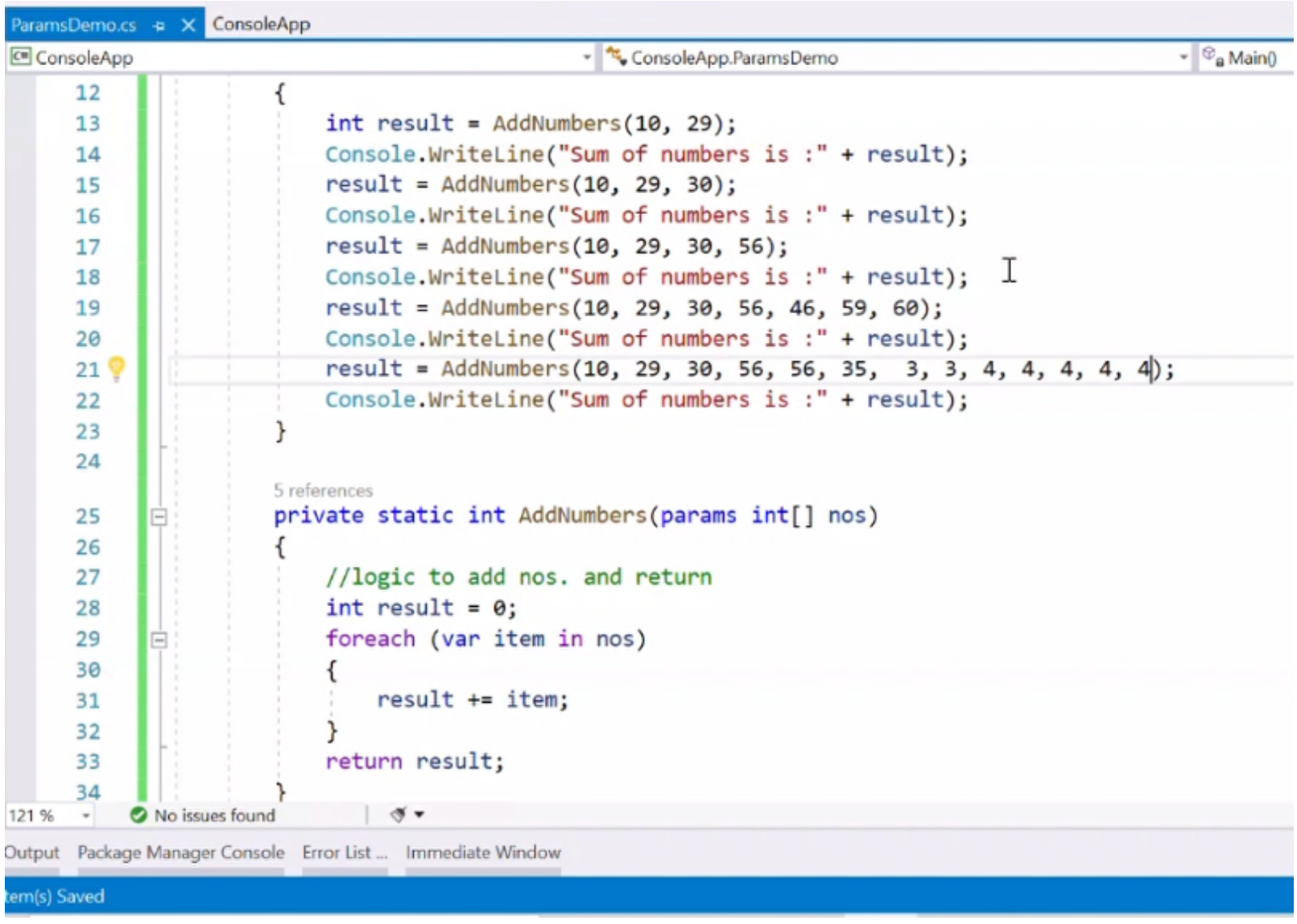
1

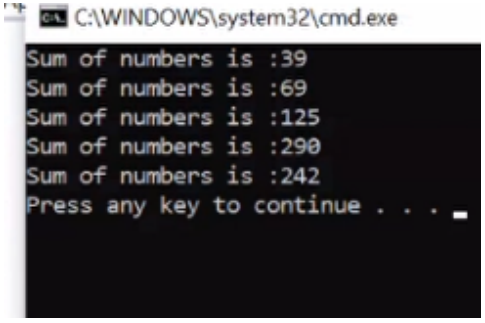
Now what i do here is ,
i'll say boss you pass whatever it is
--------i will converts this to my array

So this will be my
Parameterized Array



Now you add 4 number or 400 number no worries , Bussiness will happy you are also happy


ParamsDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.ParamsDemo
Main()
{
int result = AddNumbers(10, 29);
Console.WriteLine("Sum of numbers is :" + result);
result = AddNumbers(10, 29, 30);
Console.WriteLine("Sum of numbers is :" + result);
result = AddNumbers(10, 29, 30, 56);
Console.WriteLine("Sum of numbers is :" + result);
result = AddNumbers(10, 29, 30, 56, 46, 59, 60);
Console.WriteLine("Sum of numbers is :" + result);
result = AddNumbers(10, 29, 30, 56, 56, 35, 3, 3, 4, 4, 4, 4, 4);
Console.WriteLine("Sum of numbers is :" + result);
}
5 references
private static int AddNumbers(params int[] nos)
{
//logic to add nos. and return
int result = 0;
foreach (var item in nos)
{
result += item;
}
return result;
}
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window
tem(s) Saved



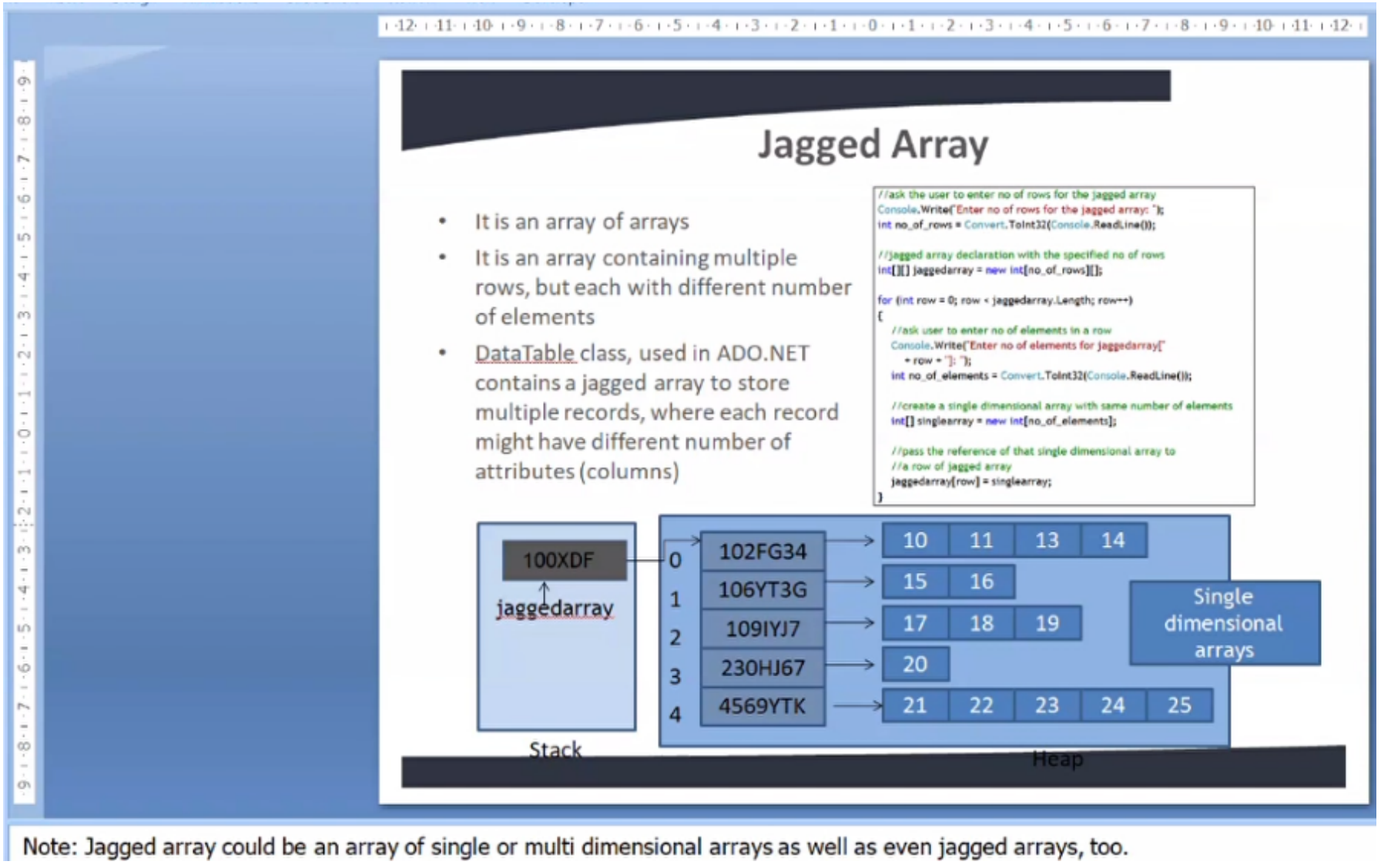
Remember when you don't know how many argument user will pass and that should be of same type

```
result = AddNumbers(10, 29, 30, 56, "Shashi");
```

You cannot do like this, it should be of the same type, when it is of the same type you can always use **PARAMS** , So params are very powerfull

Have we done something like this in java ?

java also supports this, in java we will write **int... nos**



Note: Jagged array could be an array of single or multi dimensional arrays as well as even jagged arrays, too.

Suppose in array

in 1st row i want to store 1 2 3

in 2nd row i want to store 2 4 5 7

int 3rd row i want to store 1 2
Something like this
So, how is this possible ?
So , where you don't have the data it will add 0 automatically

```
1, 2, 3, 0
2, 4, 5, 7
1, 2, 0, 0
```

But i don't want to do it i want this 2 number to be store here and i want this 3 number to be stored here

```
1, 2, 3
2, 4, 5, 7
1, 2
```

so then what i need to do, i need to plan my self
i write
int [ ] [ ] ----------1st box i'll say how many rows i need, 2nd box i'll say how many columns i need
int [1] [3]-------1row with 3 columns

```
int[1][3] = 1, 2, 3
int[1][4] = 2, 4, 5, 7
int[1][2] = 1, 2
```

this is like 3 elements, 3 arrays i'm creating, but i don't want to create 3 array i want to handle them in one array because of the space complexity and all
So, i know how many rows i need, i know that i need 3 rows, but i don't know how many columns i need So, i'm not bothered

```
int[3][]
```

So, this becomes my **Jagged Array**

Remember jagged array are different they are not multidimentional array
i'm not saying here

```
int[3,4]
```

i'm not doing this

```
int[3][4]
```

i'm giving 2 different boxes--------and this clearly says this is a array which will have a array inside it
its like a array inside a array

What will be the rank of this jagged array? --remember rank is dimention so what will be the dimention of this array?

if i say **int [ ]** this is one dimention correct

when i put a comma **int [,]** this is two dimention correct

when i put one more comma **int [, ,]** this is three dimention correct

when i put **int [, , ,]** this is four dimention correct

if i put comma then only dimention will change it will be **2,2 3,3 2,3 .....**it will be like that so this will be a square matrix or a rectangular matrix where the number of rows and the number of columns are fixed

but this **int [ ]** is still single dimentional array the rank will be one right

so for that i'm adding new array int [ ] [ ] So, what will be the rank -----**This is**

Single Dimention

So, in case of jagged array

```
int[1][3] = 1, 2, 3
int[1][4] = 2, 4, 5, 7
int[1][2] = 1, 2
```

No of Rows i know but column i don't, it can be 3 column, it can be 4 column, it can be 2 column

So, in this particular case what it does

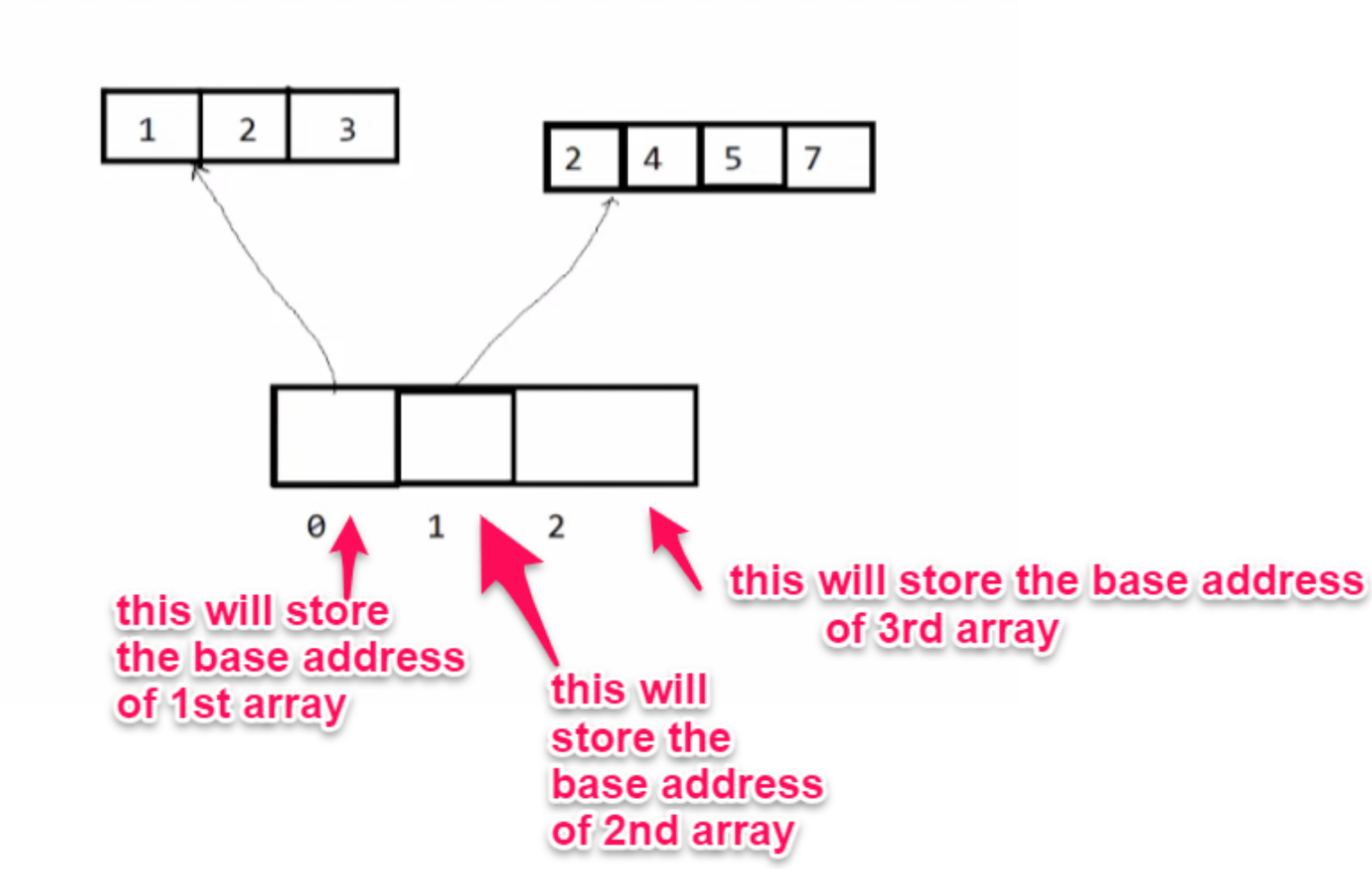
to this particular Array

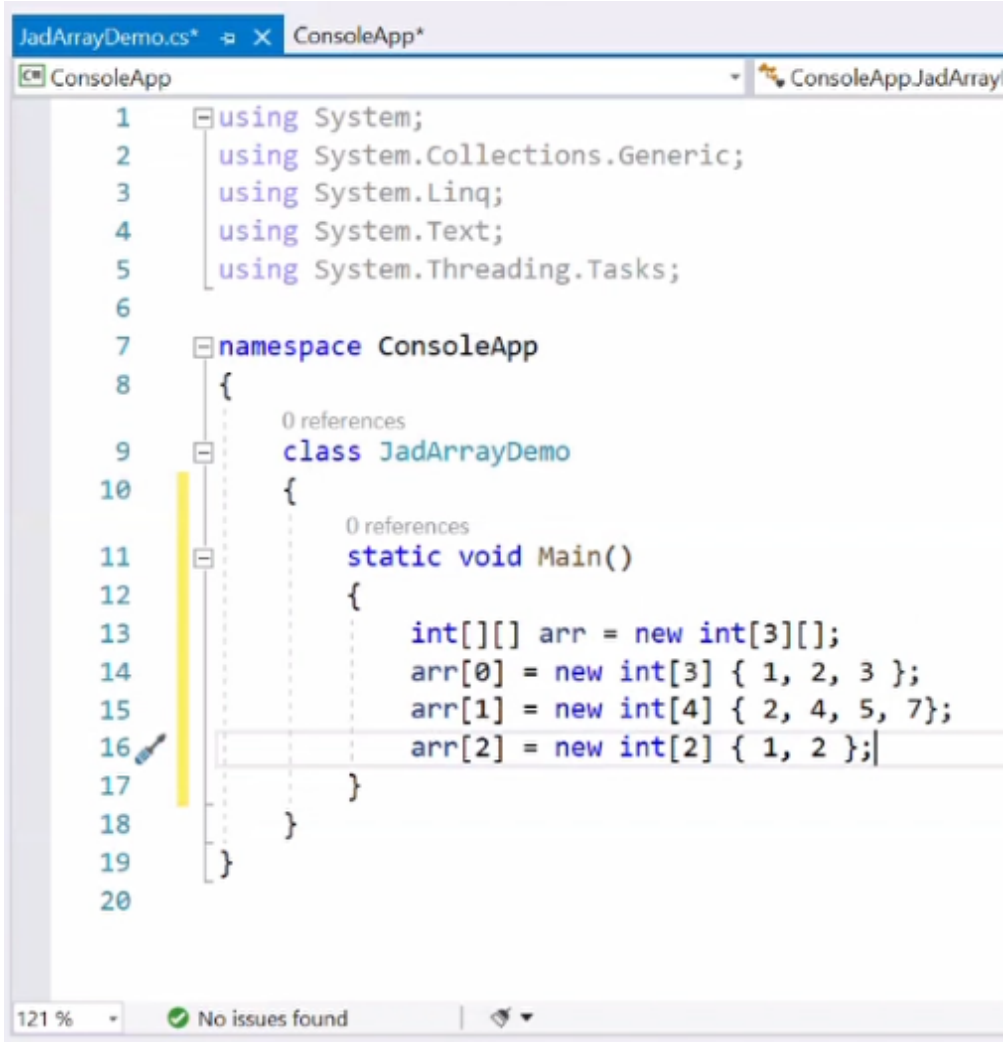
it will add this as Array

it is like pointing to that particular array

```
int[1][3] = 1, 2, 3
int[1][4] = 2, 4, 5, 7
int[1][2] = 1, 2
```

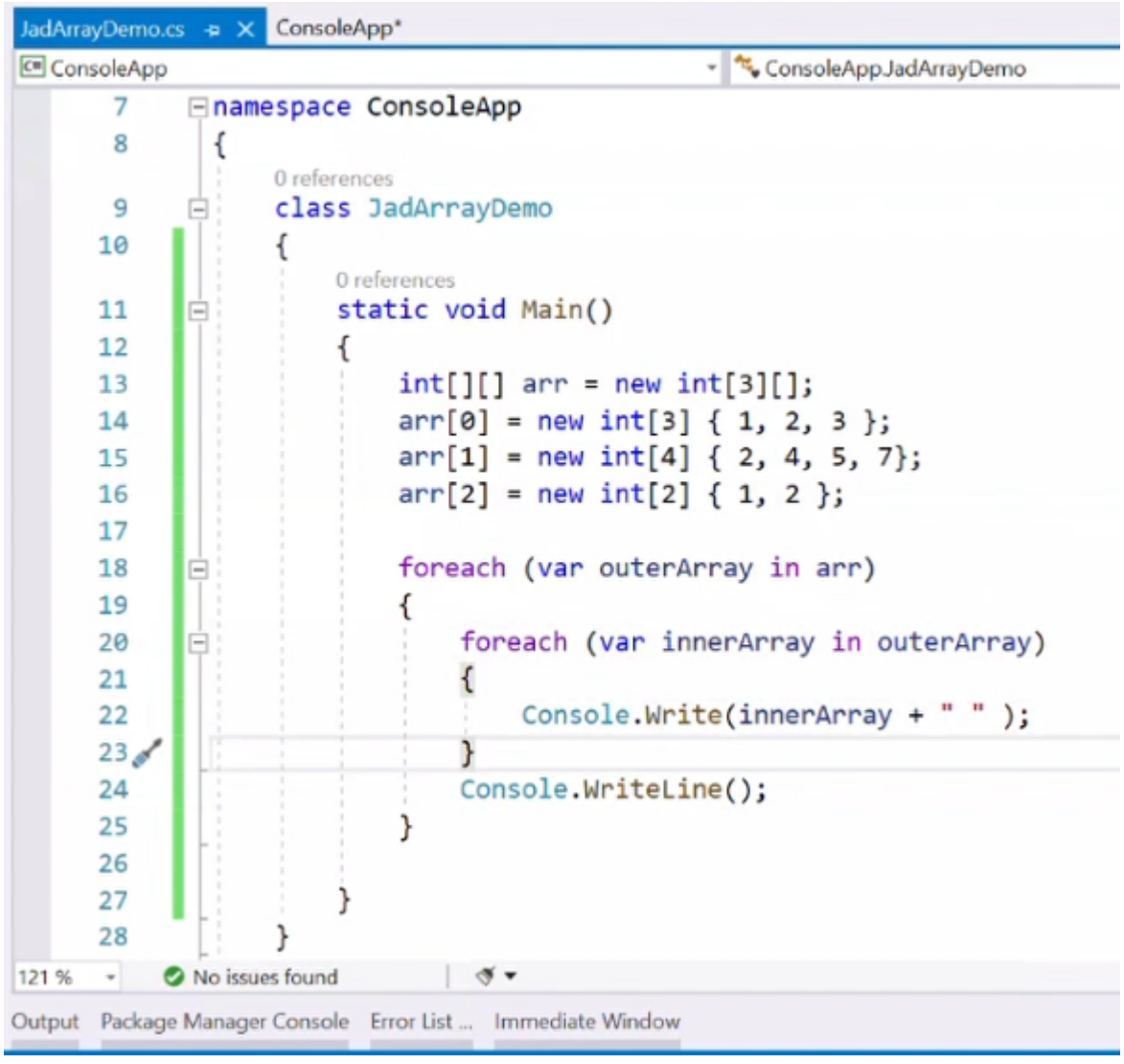
int [3] [ ] will be look like this

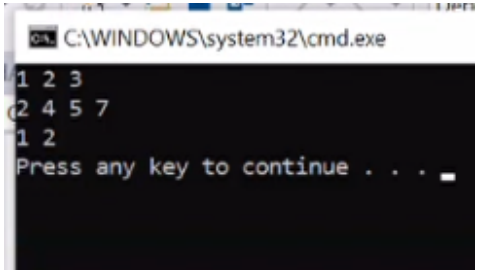




this is how we create jagged array

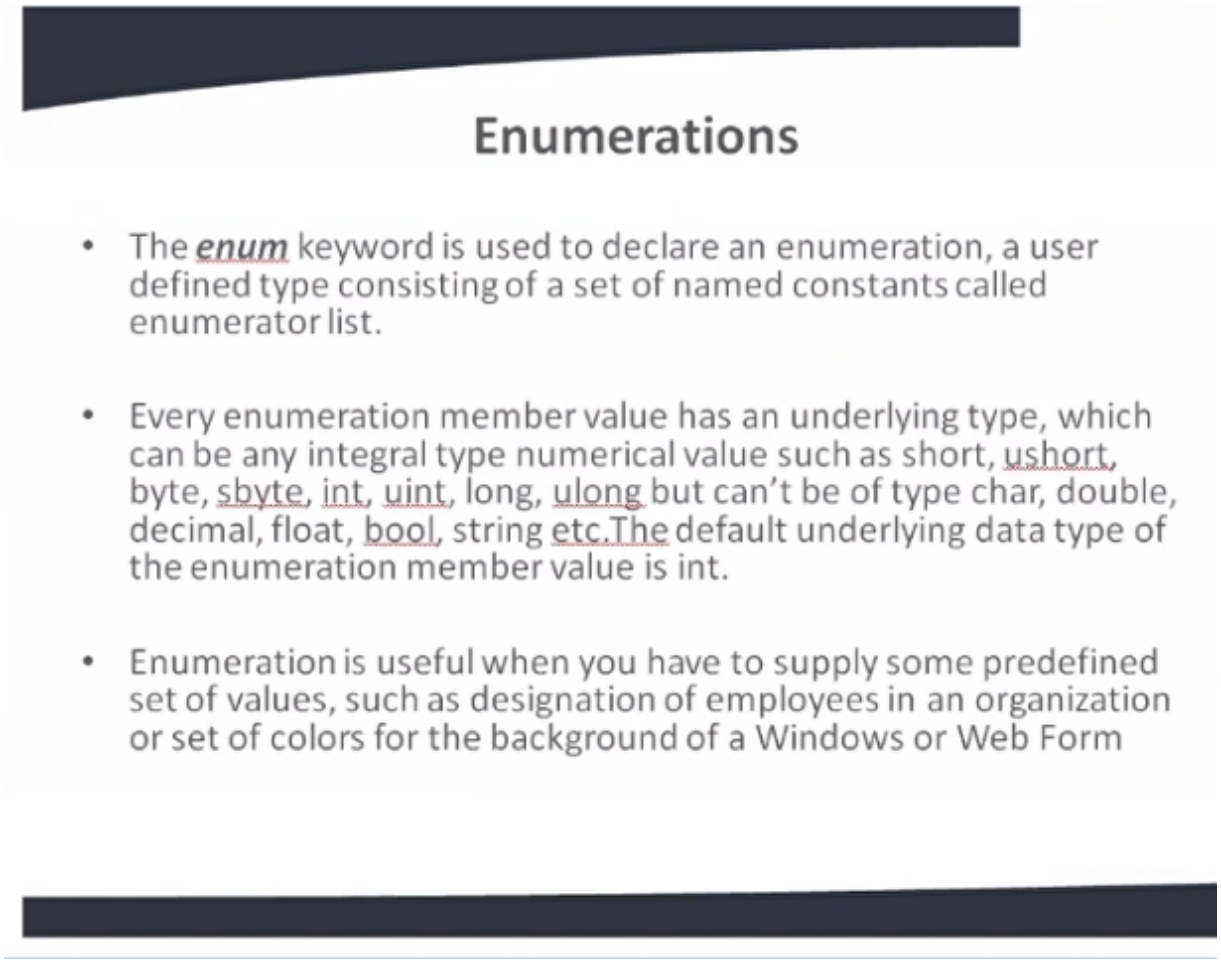
Now, how do you read the data from this array so, this is 2 different array ----------array inside array


JadArrayDemo.cs
ConsoleApp*
ConsoleApp
ConsoleApp.JadArrayDemo
namespace ConsoleApp
{
0 references
class JadArrayDemo
{
0 references
static void Main()
{
int[][] arr = new int[3][];
arr[0] = new int[3] { 1, 2, 3 };
arr[1] = new int[4] { 2, 4, 5, 7};
arr[2] = new int[2] { 1, 2 };
foreach (var outerArray in arr)
{
foreach (var innerArray in outerArray)
{
Console.Write(innerArray + " ");
}
Console.WriteLine();
}
}
}
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window

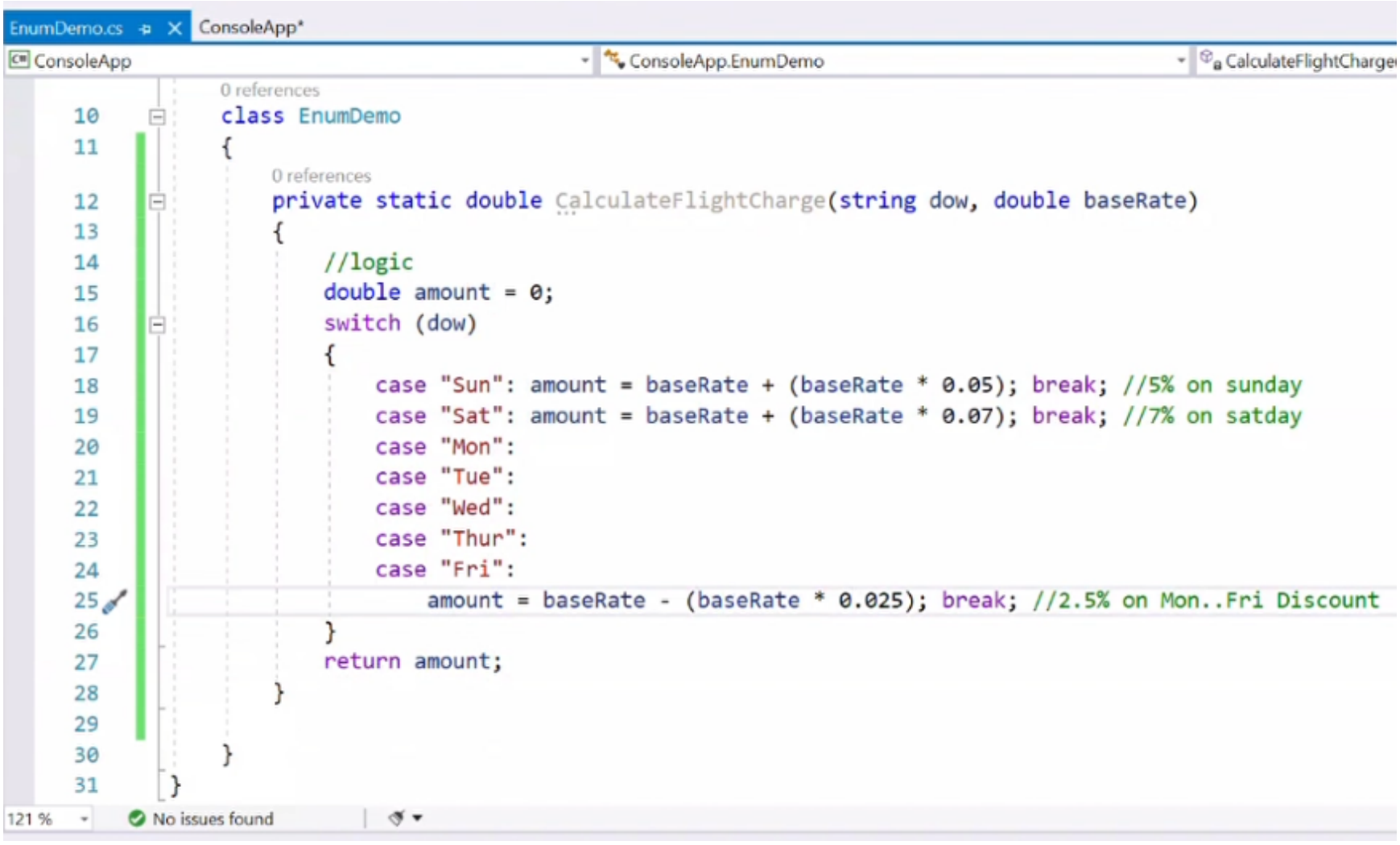


Entering Items Into and Displaying Items From Jagged Array

```
for (int row = 0; row < jaggedarray.Length; row++)
{
    for (int column = 0; column < jaggedarray[row].Length;
        column++)
    {
        //ask the user to enter item in particular column of
        //particular row
        Console.Write("Enter item in jaggedarray["
            + row + "][" + column + "]: ");
        jaggedarray[row][column] = Convert.ToInt32(
            Console.ReadLine());
    }
    Console.WriteLine("\n");
}

for (int row = 0; row < jaggedarray.Length; row++)
{
    for (int column = 0; column < jaggedarray[row].Length;
        column++)
    {
        //display to the user an item from particular column from
        //particular row
        Console.Write("Item at jaggedarray["
            + row + "][" + column + "]: "
            + jaggedarray[row][column]);
    }
    Console.WriteLine("\n");
}
```



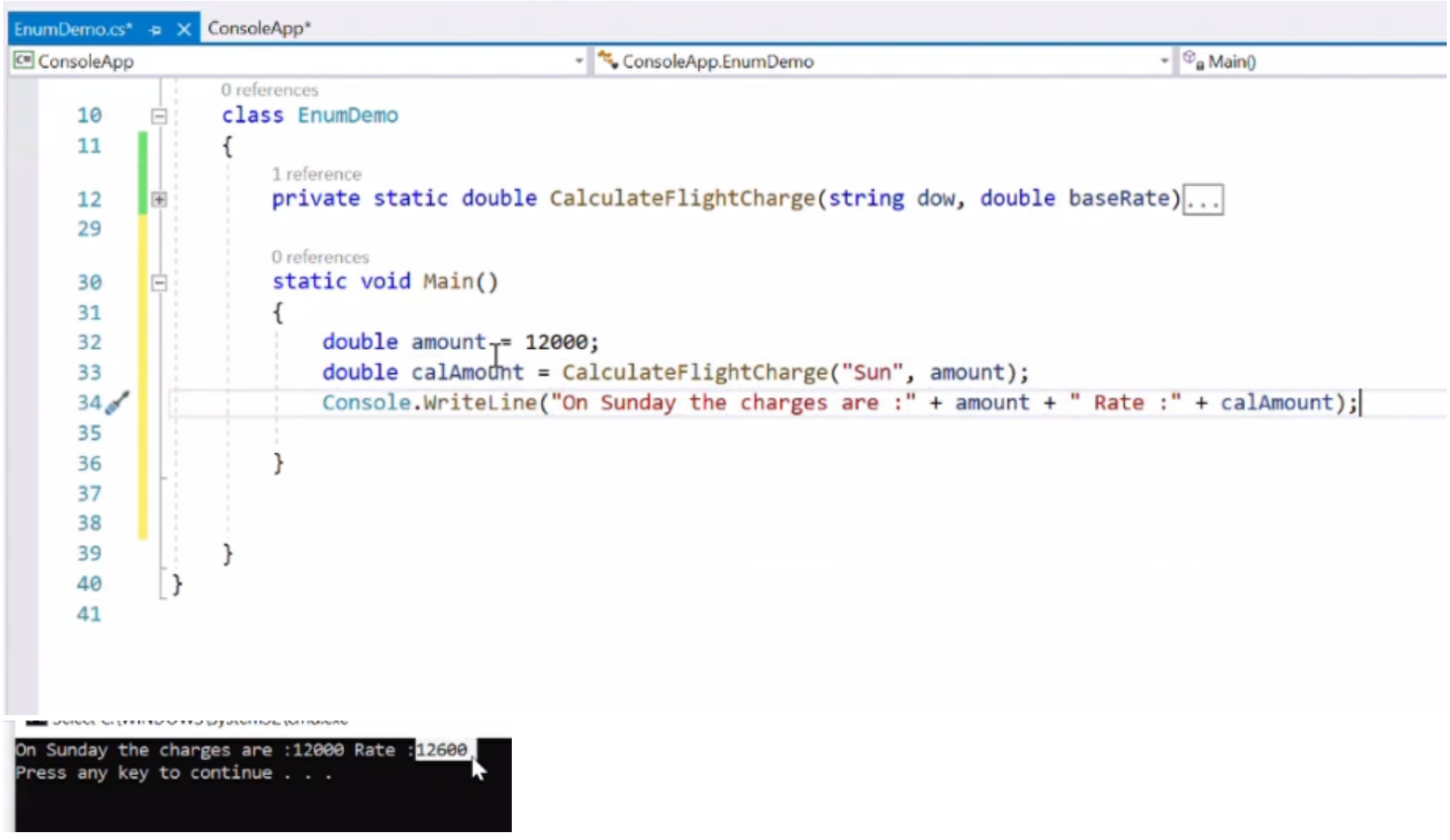
Suppose i am a particular programmer who takes a particular day of the week
lets take a real time flight example
if it is weekends then the flight charges will be little heigh
but where as in week days it will be less
and if it is a weekend + any festival it will more heigh



now we have a logic which we have already written ,

now somebody will use our program

we know that any code you write is not for you , mostly someone else will use it

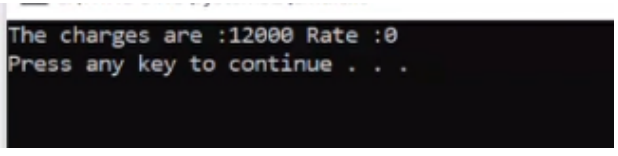


what if it is wednesday

EnumDemo.cs | ConsoleApp

ConsoleApp | ConsoleApp.EnumDemo | Main()

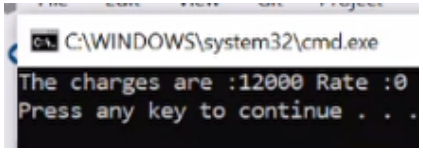
```
0 references
class EnumDemo
{
    1 reference
    private static double CalculateFlightCharge(string dow, double baseRate)...

    0 references
    static void Main()
    {
        double amount = 12000;
        double calAmount = CalculateFlightCharge("wed", amount);
        Console.WriteLine("The charges are :" + amount + " Rate :" + calAmount);
    }
}
}
```



Why the rate is 0

What if its a sunday?

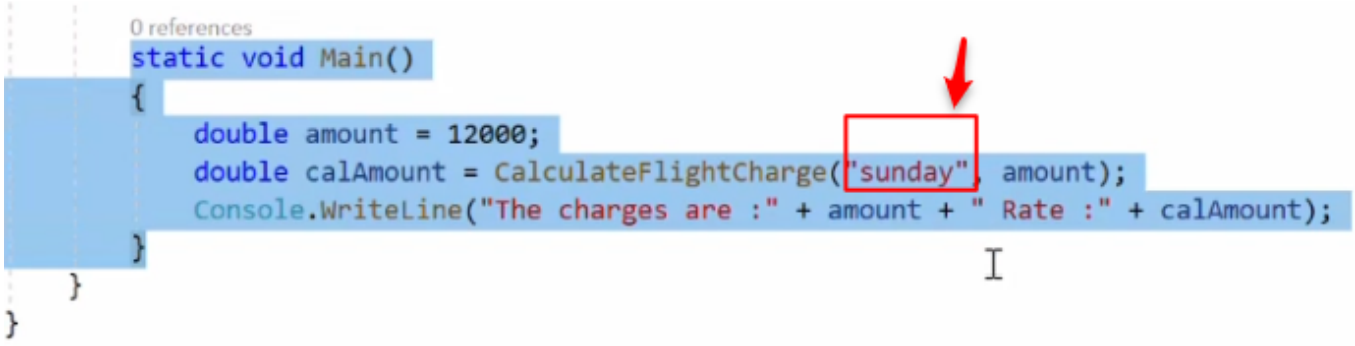
```
numDemo.cs  ConsoleApp
ConsoleApp                ConsoleApp.EnumDemo                Main()
        0 references
10      class EnumDemo
11      {
            1 reference
12          private static double CalculateFlightCharge(string dow, double baseRate)...
29
            0 references
30          static void Main()
31          {
32              double amount = 12000;
33              double calAmount = CalculateFlightCharge("sunday", amount);
34              Console.WriteLine("The charges are :" + amount + " Rate :" + calAmount);
35          }
36      }
37  }
38
```



Why the rate is again 0

Because the ediot who is using our code

he passed sunday but or logic is for Sun



```
        0 references
        static void Main()
        {
            double amount = 12000;
            double calAmount = CalculateFlightCharge("sunday", amount);
            Console.WriteLine("The charges are :" + amount + " Rate :" + calAmount);
        }
    }
}
```

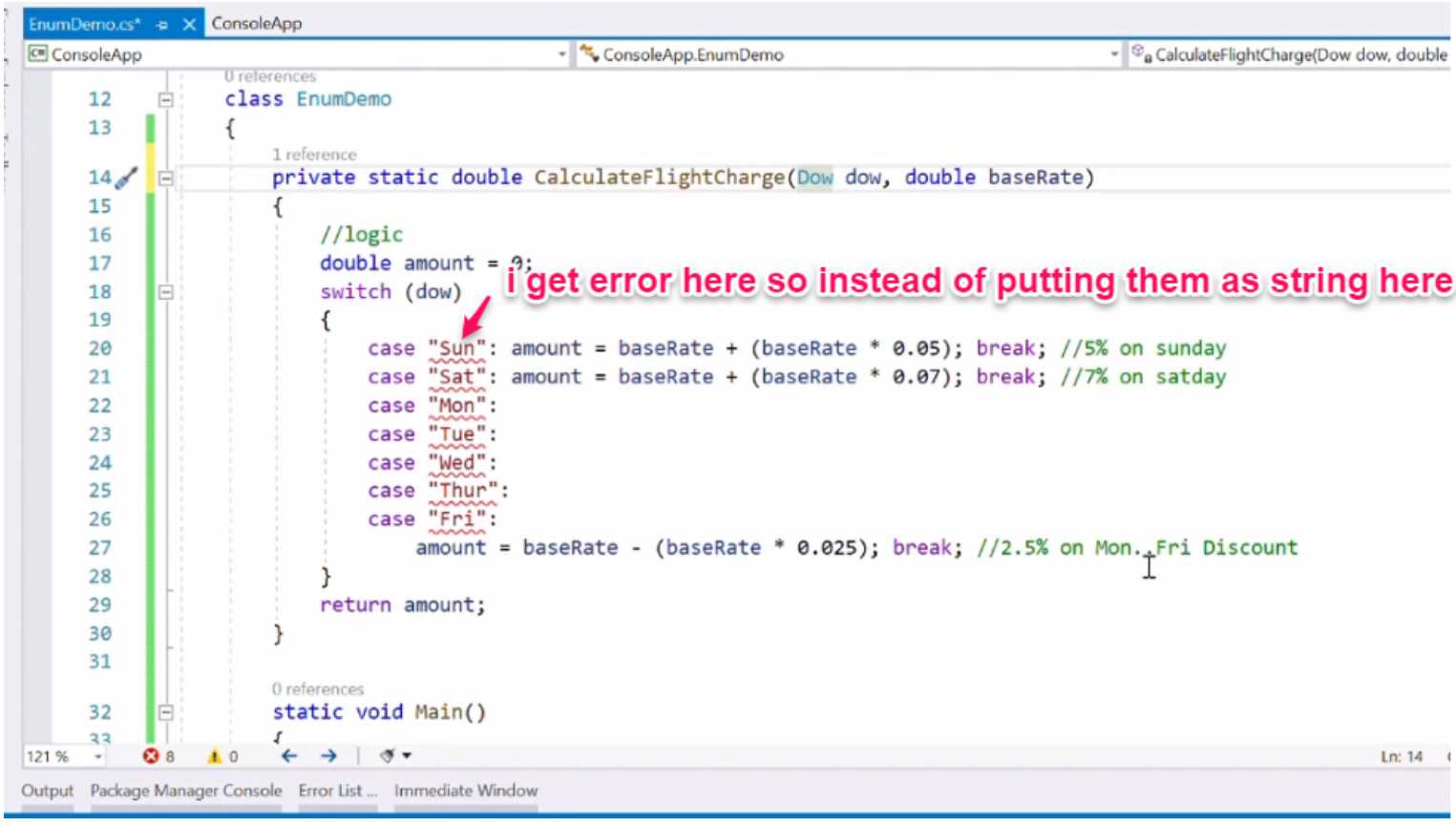
So, how will you make sure that the code what ever you have written will not break as it is breaking now

So, for that i need to have an agreement with this guy who is using my code you have to pass this particular value only, i'm restricting him

So, how do you do that?

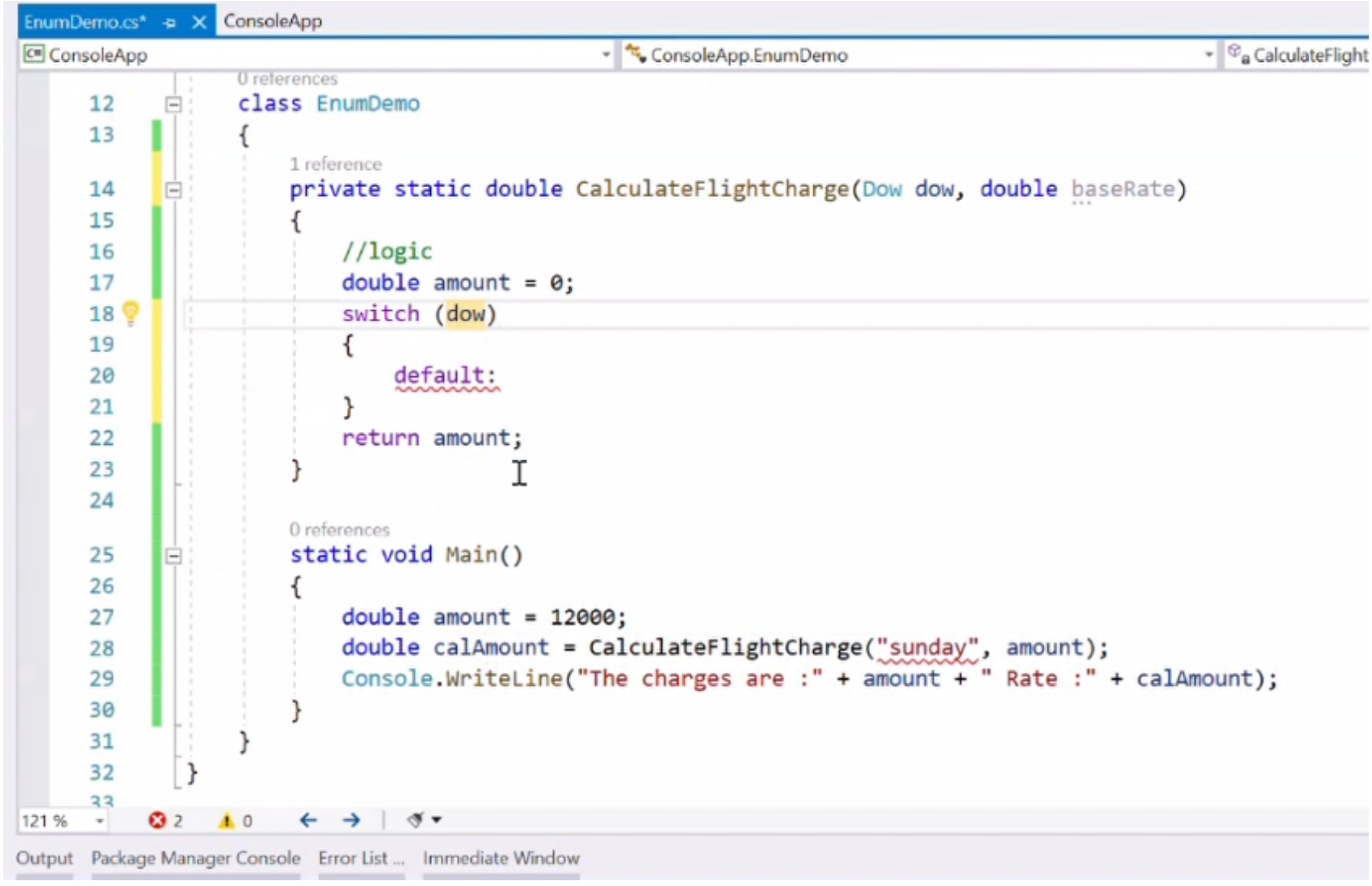
----I'll just create a ENUM


EnumDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.EnumDemo
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
enum Dow {Sun, Mon, Tue, Wed, Thur, Fri, Sat};
//Flight
0 references
class EnumDemo
{
1 reference
private static double CalculateFlightCharge(string dow, double baseRate)...
0 references
static void Main()
{
double amount = 12000;
double calAmount = CalculateFlightCharge("sunday", amount);
Console.WriteLine("The charges are :" + amount + " Ra
(local variable) double amount
}
}
}
now instead of calling this as string i call this as Dow

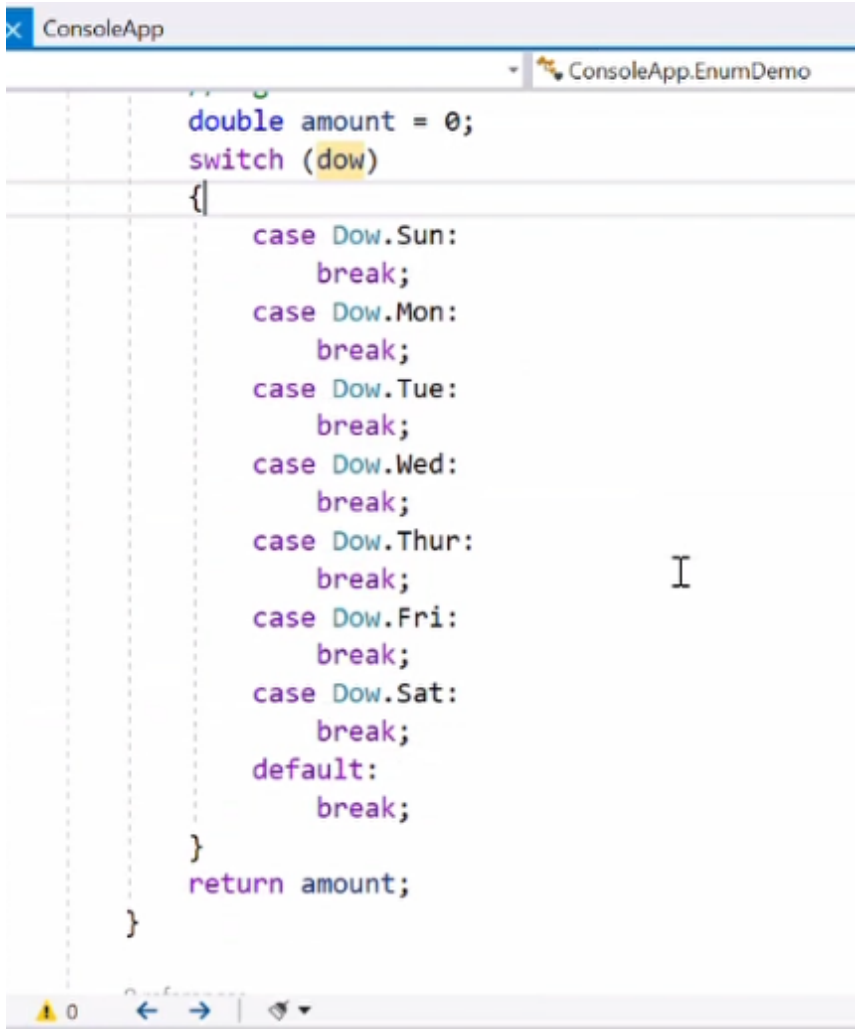


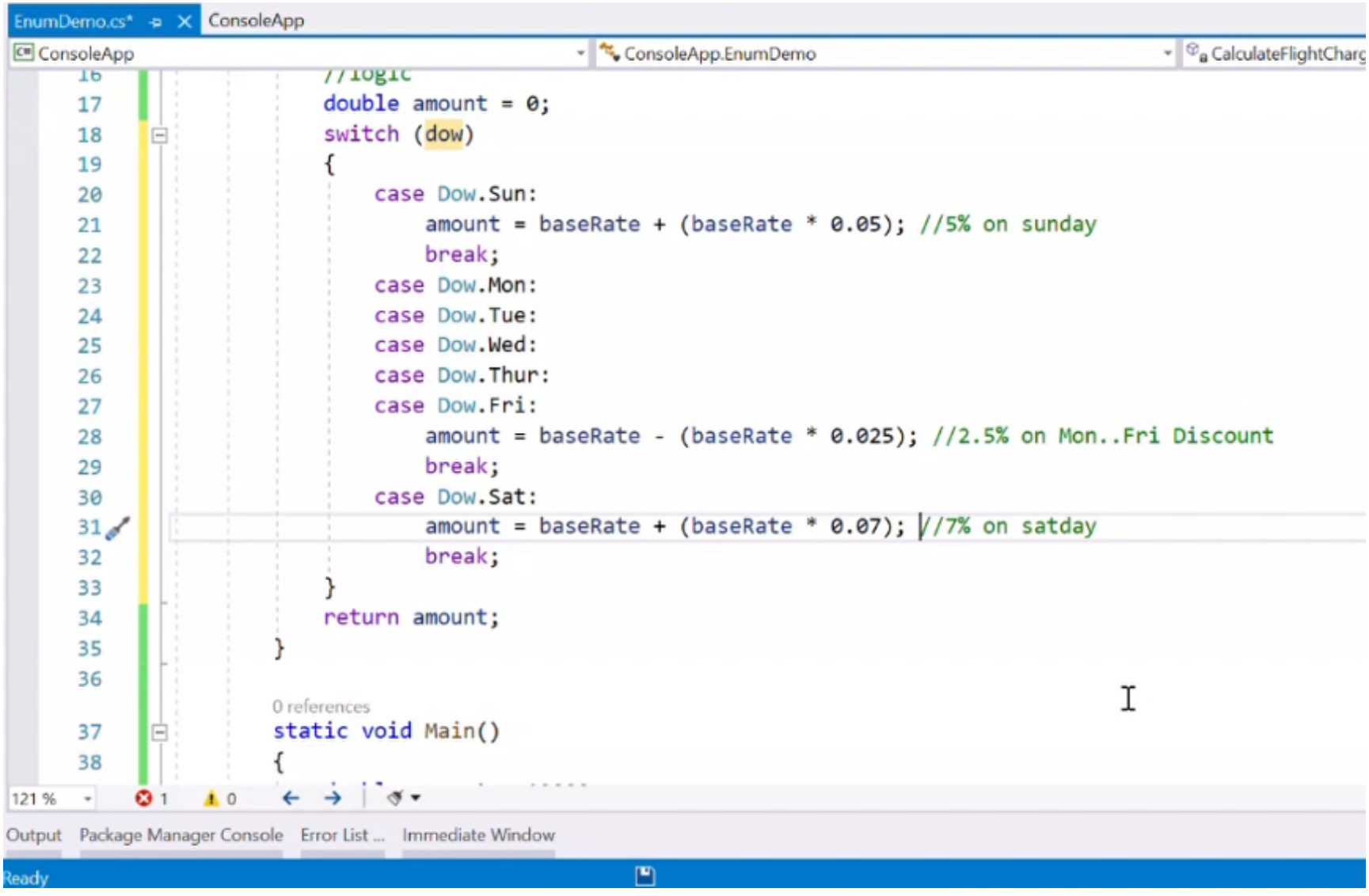
to do this look at the magical thing i'll write switch ctrl + space

then dow and i press down arrow



look at this the code is written for me

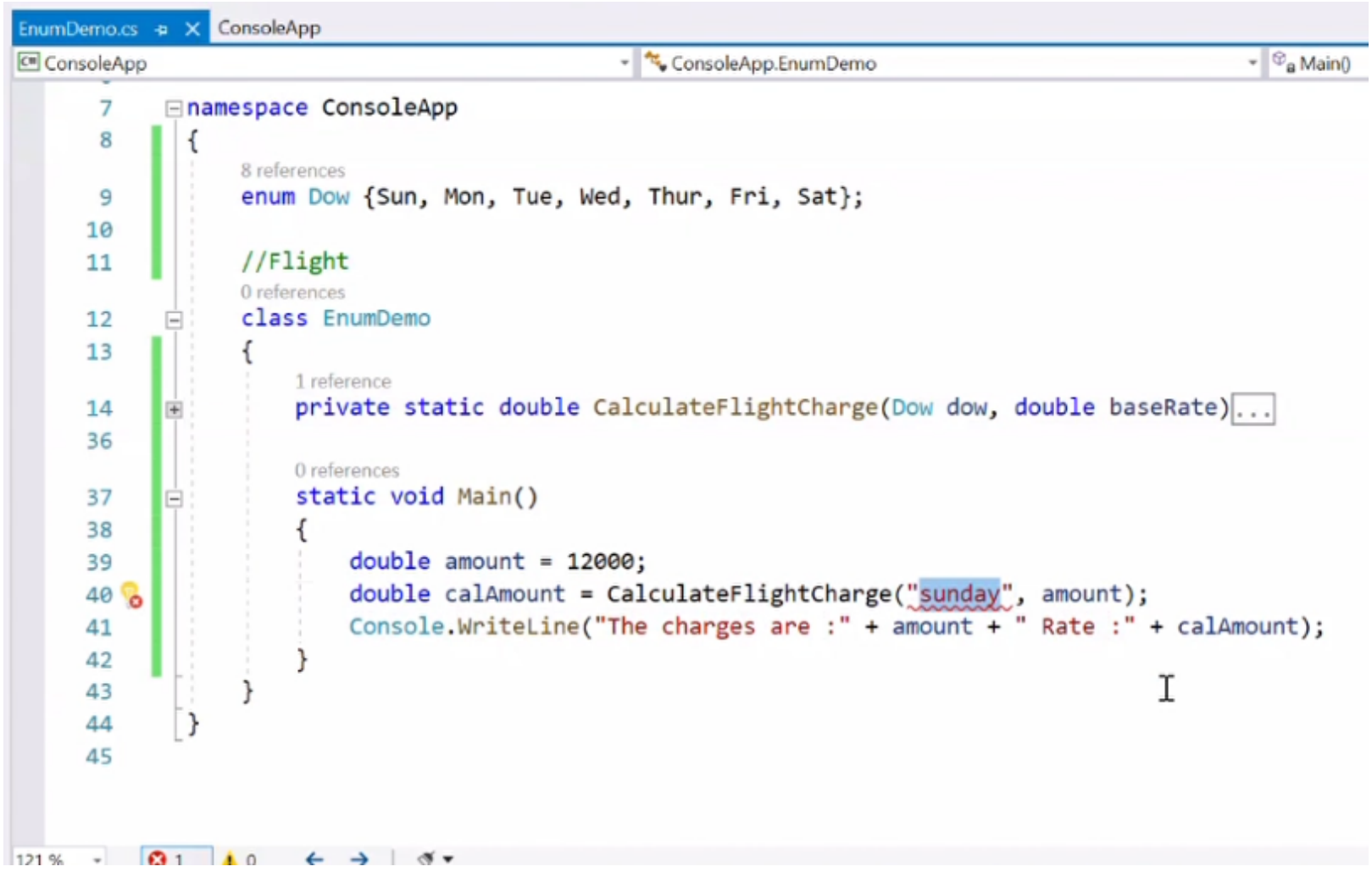

ConsoleApp
ConsoleApp.EnumDemo
double amount = 0;
switch (dow)
{
case Dow.Sun:
break;
case Dow.Mon:
break;
case Dow.Tue:
break;
case Dow.Wed:
break;
case Dow.Thur:
break;
case Dow.Fri:
break;
case Dow.Sat:
break;
default:
break;
}
return amount;
}
0



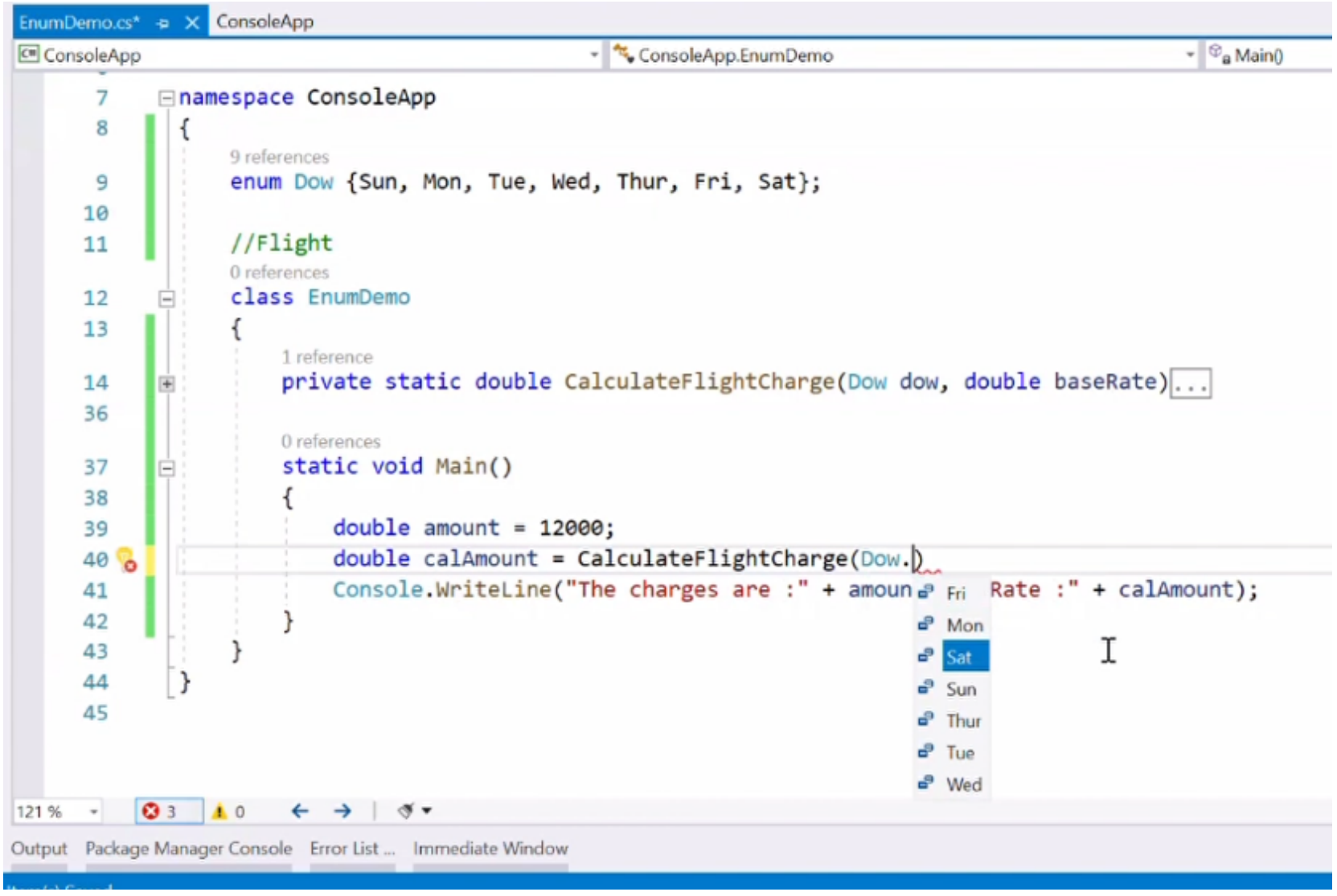
Now what's the advantage we'll get with enum

Now the enum what ever i'm creating is my user defined type

now look at the magic -------this guy cannot pass-------- sunday it wil give him error



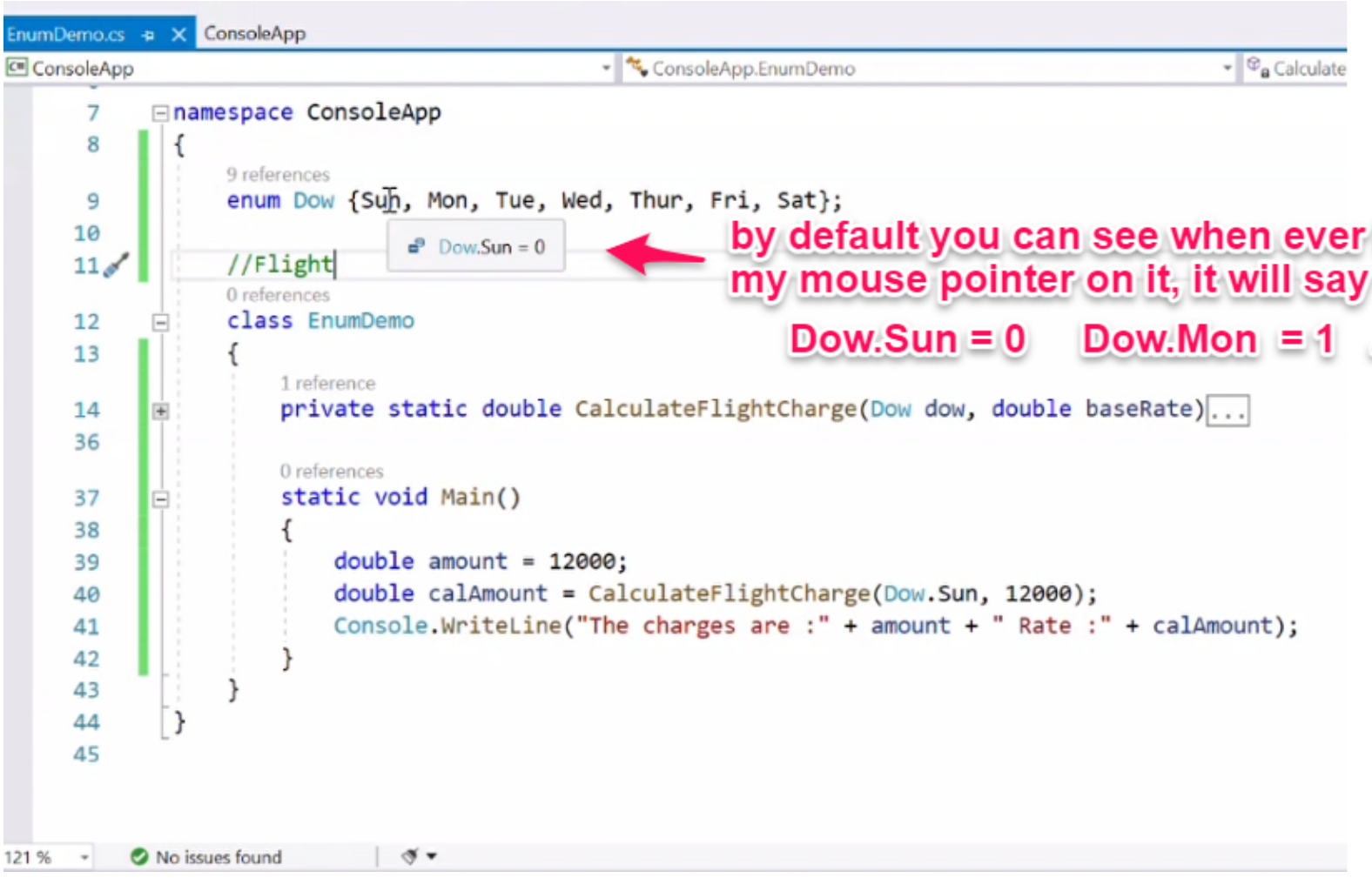
he just need to write Dow. and he will get the suggestion



he'll pass Sun or whatever and he'll pass the amount

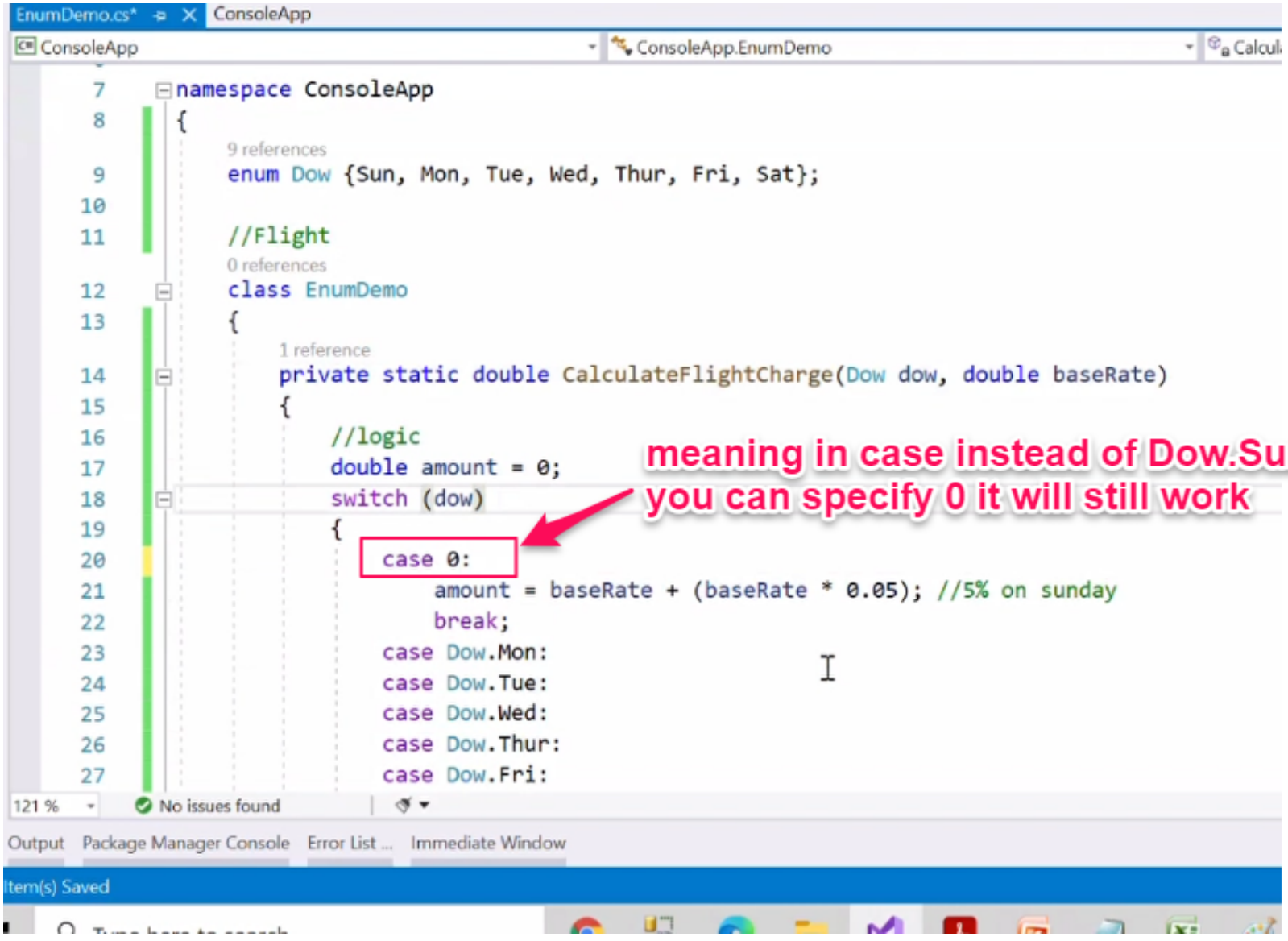
Now we don't have any error which he or she can make, when ever they are using my method this is the main purpose of using enum

you can initialize your enums

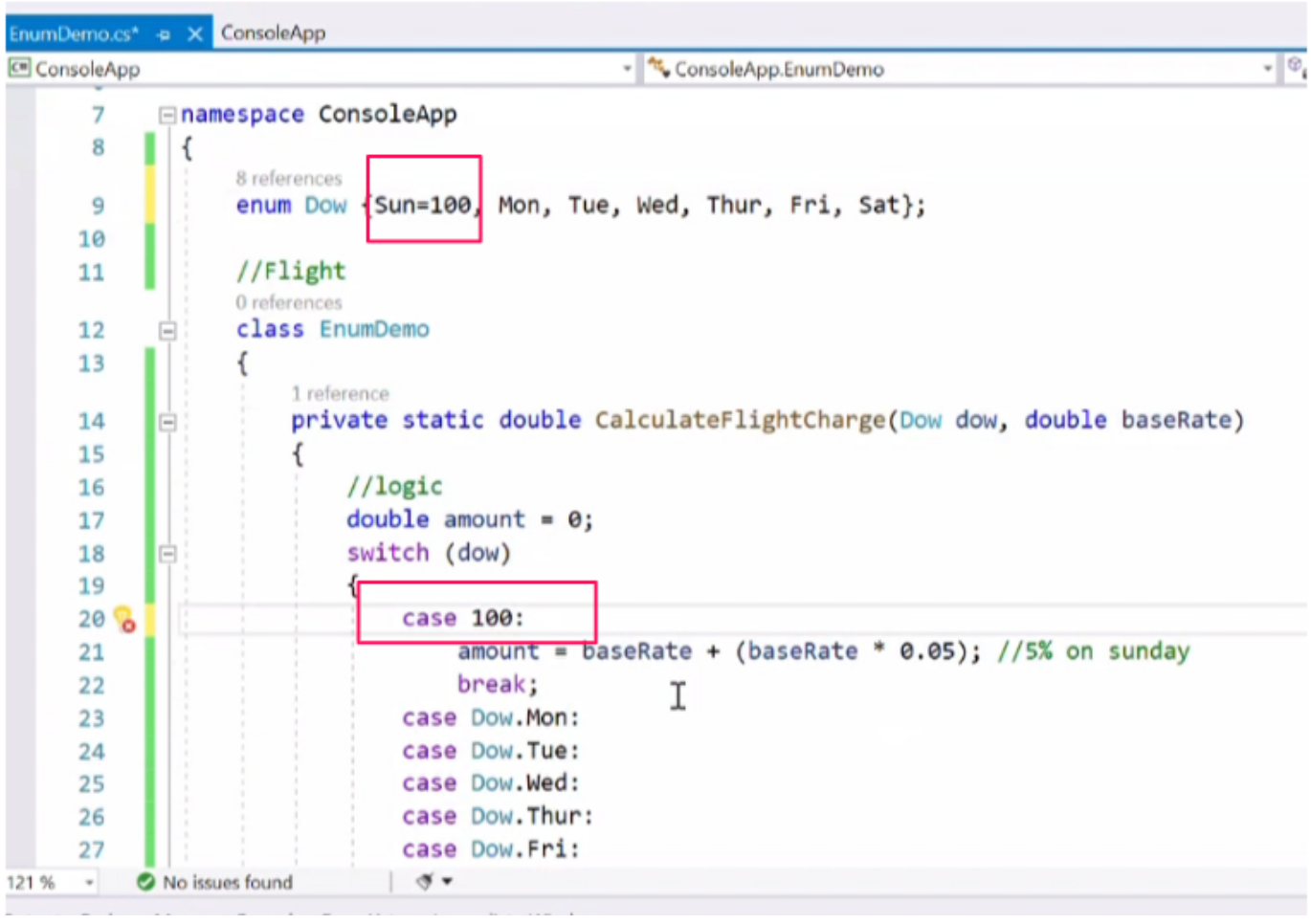


by default you can see when ever i move my mouse pointer on it, it will say

Dow.Sun = 0 Dow.Mon = 1 .....soon.....


EnumDemo.cs*
ConsoleApp
ConsoleApp
ConsoleApp.EnumDemo
namespace ConsoleApp
{
9 references
enum Dow {Sun, Mon, Tue, Wed, Thur, Fri, Sat};
//Flight
0 references
class EnumDemo
{
1 reference
private static double CalculateFlightCharge(Dow dow, double baseRate)
{
//logic
double amount = 0;
switch (dow)
{
case 0:
amount = baseRate + (baseRate * 0.05); //5% on sunday
break;
case Dow.Mon:
case Dow.Tue:
case Dow.Wed:
case Dow.Thur:
case Dow.Fri:
meaning in case instead of Dow.Sun
you can specify 0 it will still work
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window
Item(s) Saved

Now, i don't want to pass 0 , i want to pass 100, so you can always initialize it by saying Sun = 100



But usually we'll not do this , we'll always go with the standard format, but if you want you can always do it

Using Enumeration

- Defining an Enumeration Type

```
enum Color { Red, Green, Blue }
```

- Using an Enumeration Type

```
Color colorPalette = Color.Red;
```

- Displaying an Enumeration Variable

```
Console.WriteLine("{0}", colorPalette); // Displays Red
```

- Displaying an Enumeration Variable's hidden value

```
Console.WriteLine("{0}", Convert.ToInt32( colorPalette));
        // Displays hidden numerical value of member Red
```

Some Important Points To Remember About Enum

- First member of any enumeration has default value 0, if not assigned anything else
- Values for next members get incremented by 1 automatically

```
enum Color { Red, Green, Blue } //Red=0, Green=1, Blue=2
```

- You can assign value to first member and remaining members gets their values automatically, incremented by 1.

```
enum Color { Red = 100, Green, Blue } //Green=101,Blue=102
```

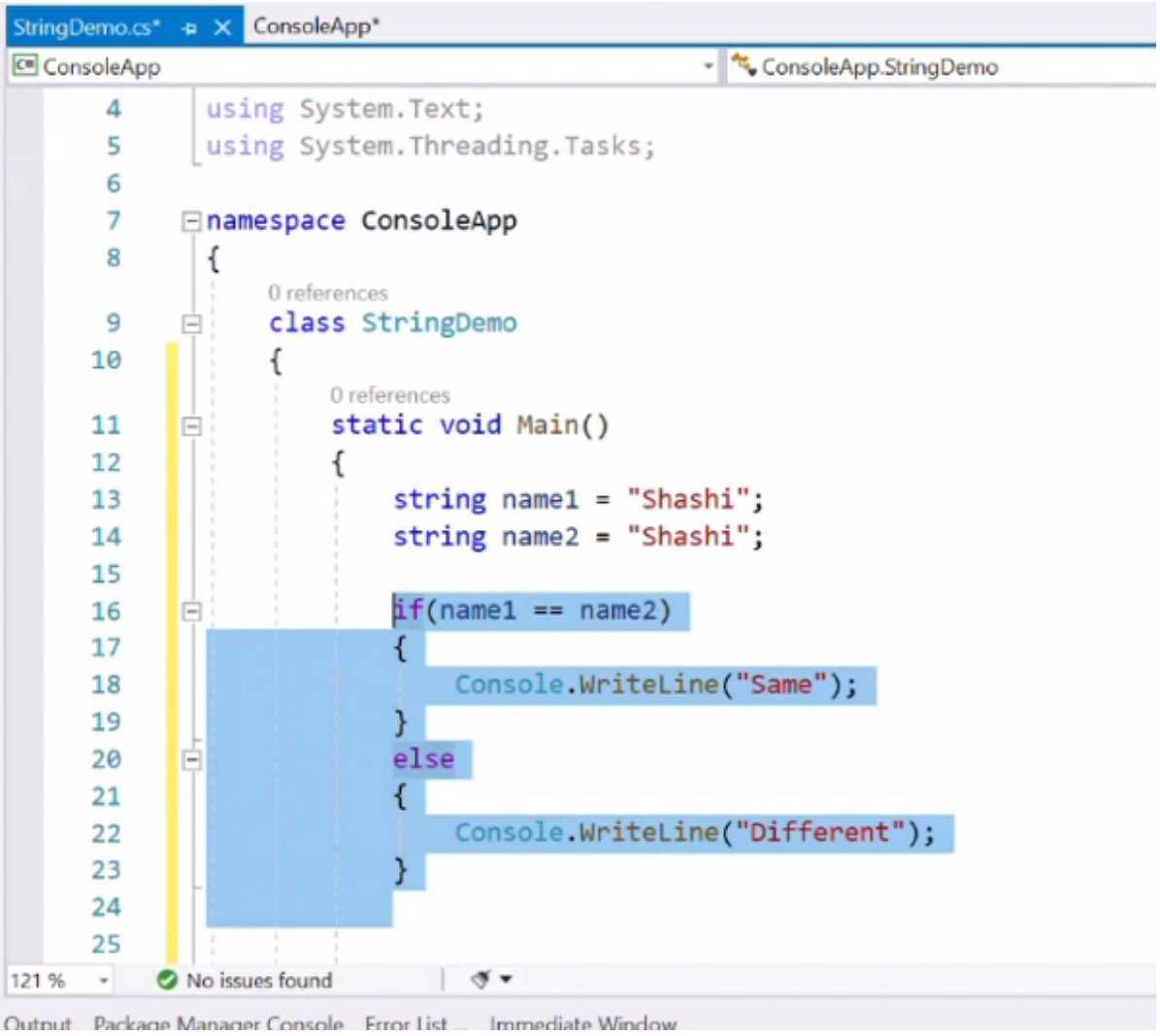
- You can assign distinct values to each and every member,

```
enum Color { Red = 100, Green = 200, Blue = 300 }
```

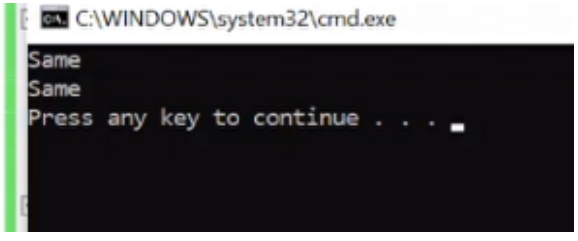
Strings in C#

- Strings are objects in .NET Framework; therefore represented through **System.String** class.
- **System.String** class provides string related functions.
- It also has keyword representation ***string*** *(blue colored)* which is said to be an alias for **System.String**
- The **string** type represents a string of Unicode characters.
- A string is a reference type.

```
Console.WriteLine("""Enter a statement for Vowels and Consonents count""");
string statement = Console.ReadLine();
int vowals_count = 0; int consonents_count = 0;
char[] arr = statement.ToCharArray();
for (int i = 0; i < arr.Length; i++)
{
   switch (arr[i])
   {
      case 'e':
      case 'i':
      case 'o':
      case 'u': vowals_count++; break;
      default: consonents_count++; break;
   }
} Console.WriteLine("No. of Vowels :" + vowals_count);
Console.WriteLine("No. of Consonents :" + consonents_count);
Console.Read();
```

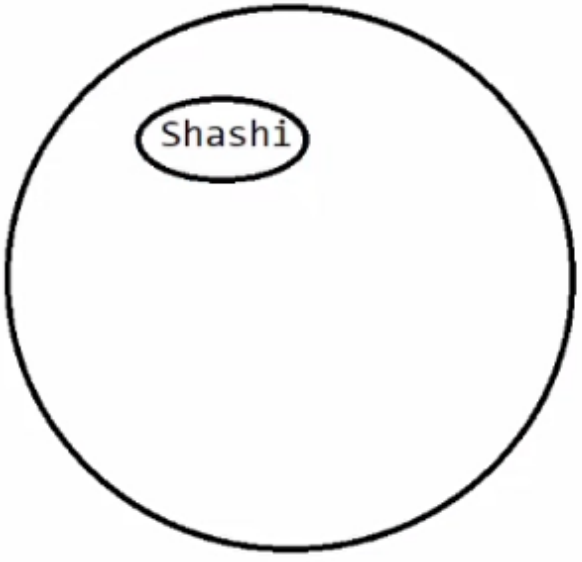

StringDemo.cs*
ConsoleApp*
ConsoleApp
ConsoleApp.StringDemo
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class StringDemo
{
0 references
static void Main()
{
string name1 = "Shashi";
string name2 = "Shashi";
if(name1 == name2)
{
Console.WriteLine("Same");
}
else
{
Console.WriteLine("Different");
}
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window


StringDemo.cs
ConsoleApp*
ConsoleApp
ConsoleApp.StringDem
0 references
static void Main()
{
string name1 = "Shashi";
string name2 = "Shashi";
if(name1 == name2)
{
Console.WriteLine("Same");
}
else
{
Console.WriteLine("Different");
}
if (name1.Equals(name2))
{
Console.WriteLine("Same");
}
else
{
Console.WriteLine("Different");
}
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window


C:\WINDOWS\system32\cmd.exe
Same
Same
Press any key to continue . . .

So when ever you work with string in java, whenever you create a string so string will always be stored in a string pool

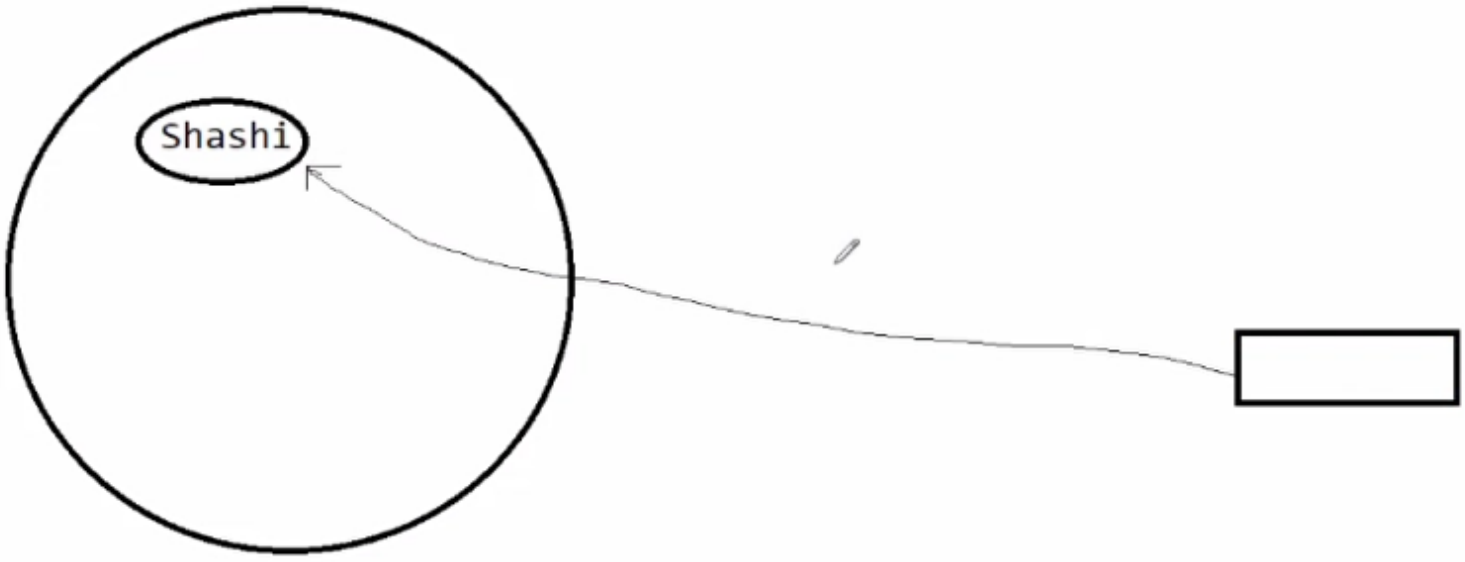
when i specify string name = "Shashi" so what it does is ,it has a string pool , so in the string pool --- shashi will be created and this is one of the pool instance



when every you say

```
string name1 = "Shashi";
```

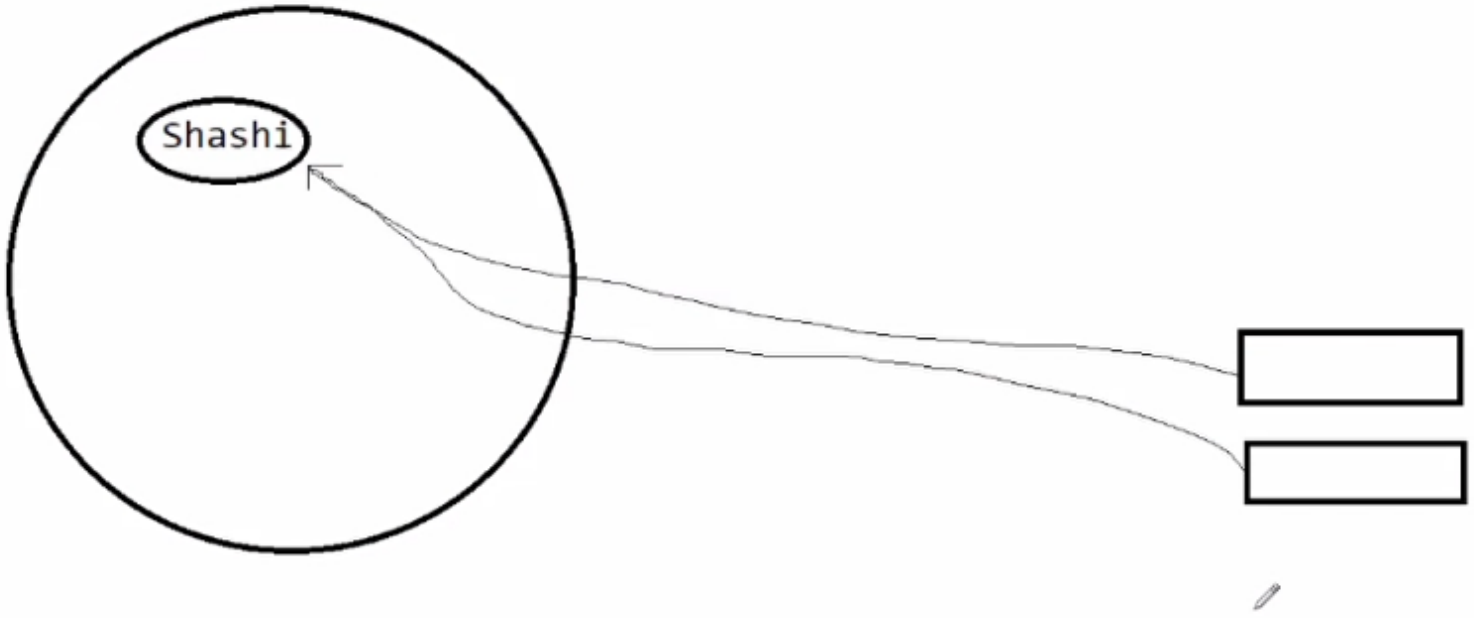
So , it will gone to this name1 = shashi refered from this particular code



Now when you are saying

```
string name2 = "Shashi";
```

then again name2 will be created here and name2 will also refer to the same string



thats why when you write

```
if(name1 == name2)
```

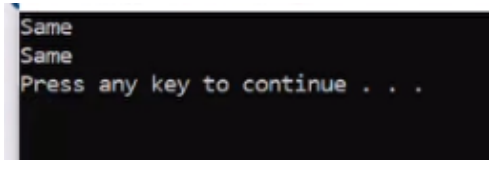
it is checking for the data , it is checking weather both of them are reffering to the same location
So, yes they have the same data so you can see the
outputs of this become true and true

StringDemo.cs | ConsoleApp

ConsoleApp | ConsoleApp.StringDemo

```
0 references
static void Main()
{
    char[] name = { 'S', 'h', 'a', 's', 'h', 'i' };
    string name1 = new string(name);
    string name2 = new string(name);

    if(name1 == name2)
    {
        Console.WriteLine("Same");
    }
    else
    {
        Console.WriteLine("Different");
    }

    if (name1.Equals(name2))
    {
        Console.WriteLine("Same");
    }
    else
    {
        Console.WriteLine("Different");
    }
```

121 % | No issues found

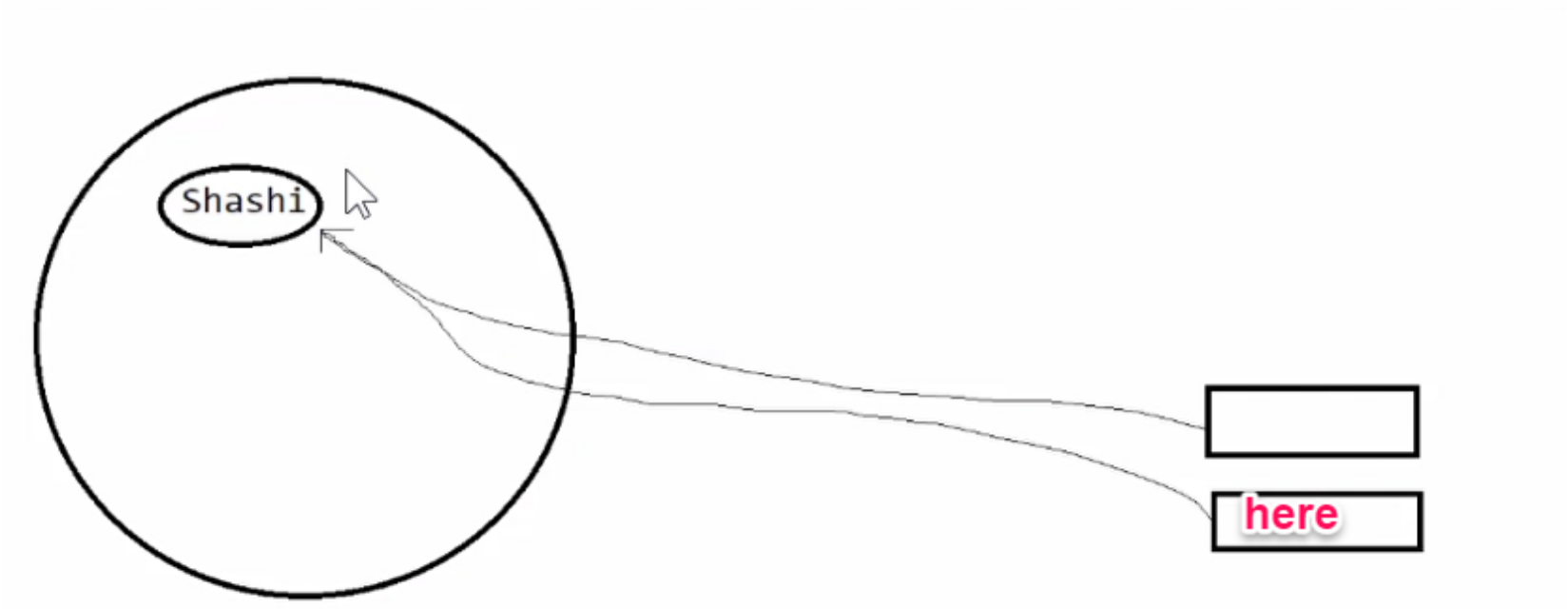


Assignment :- Read What is the start index and end index ?

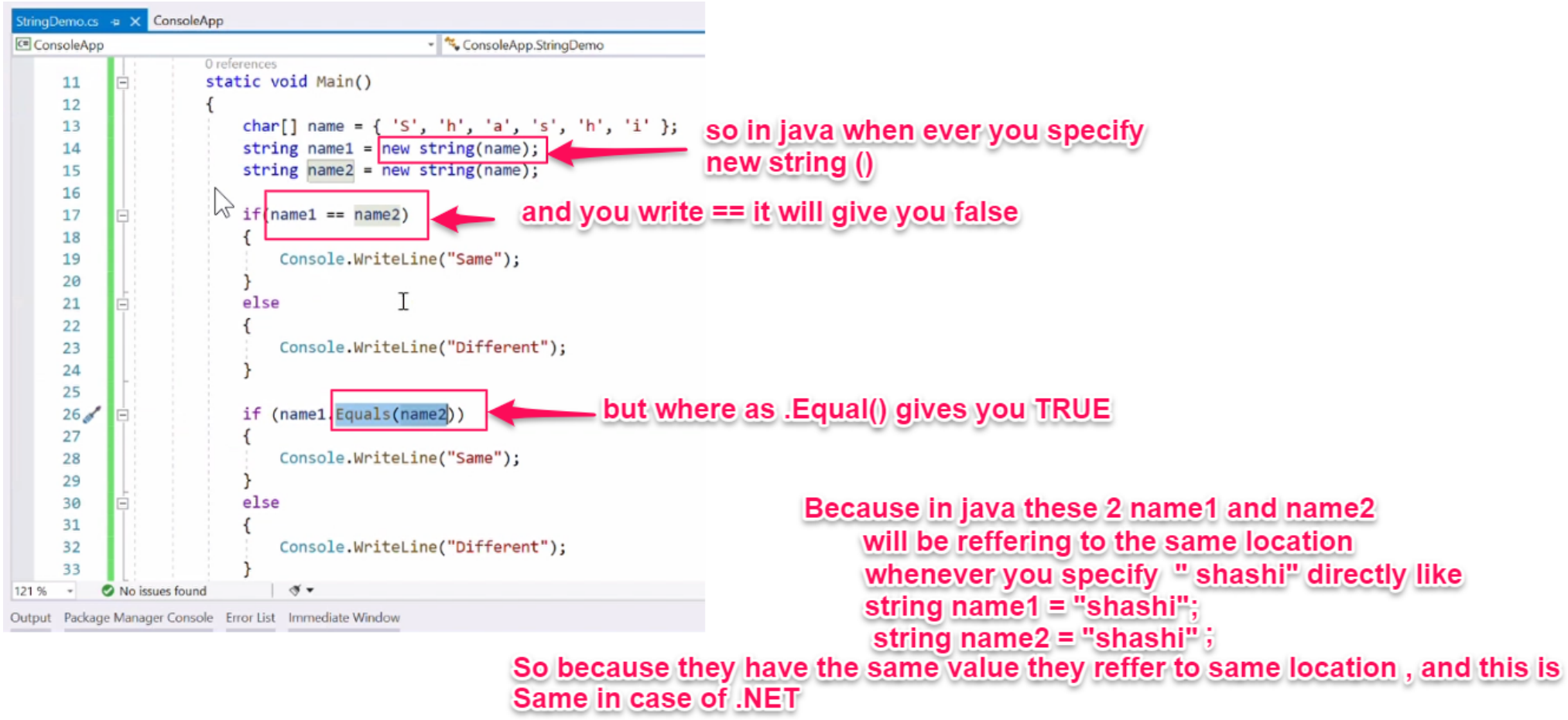
So, strings are **Muteable**

what are Mutable ?

Once the string is been allocated if you try to modify this particular string , so it will allocate the new location it will not be changed in the same place and the new location address will be stored here



We have done the same thing in java also,

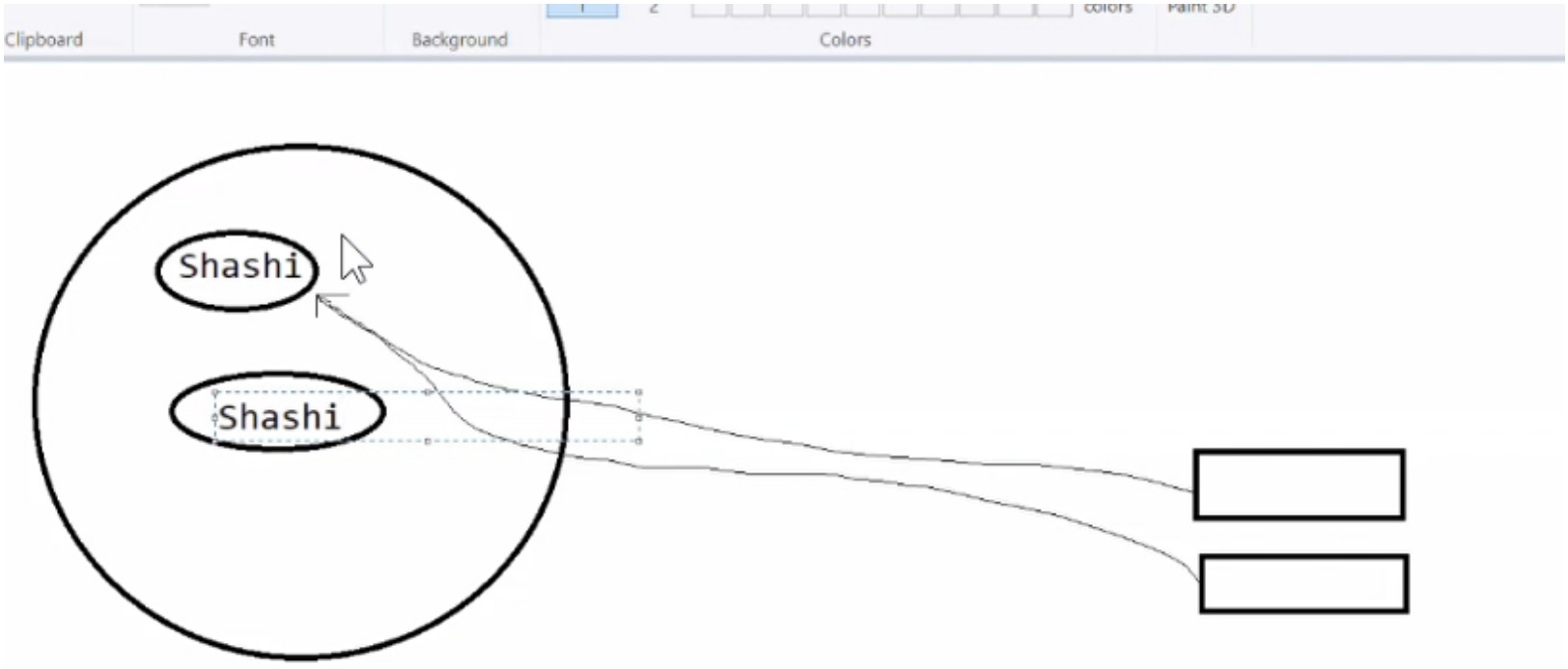


Now here the main difference in java and .NET would be

in java when ever you specify name = new string() and name is of string type ,so there will be lot of change

What will be the change there?

So, it will be create 2 different instances in java



and whenever you compare it you say == it will give you FALSE at that time when you create new instance in java

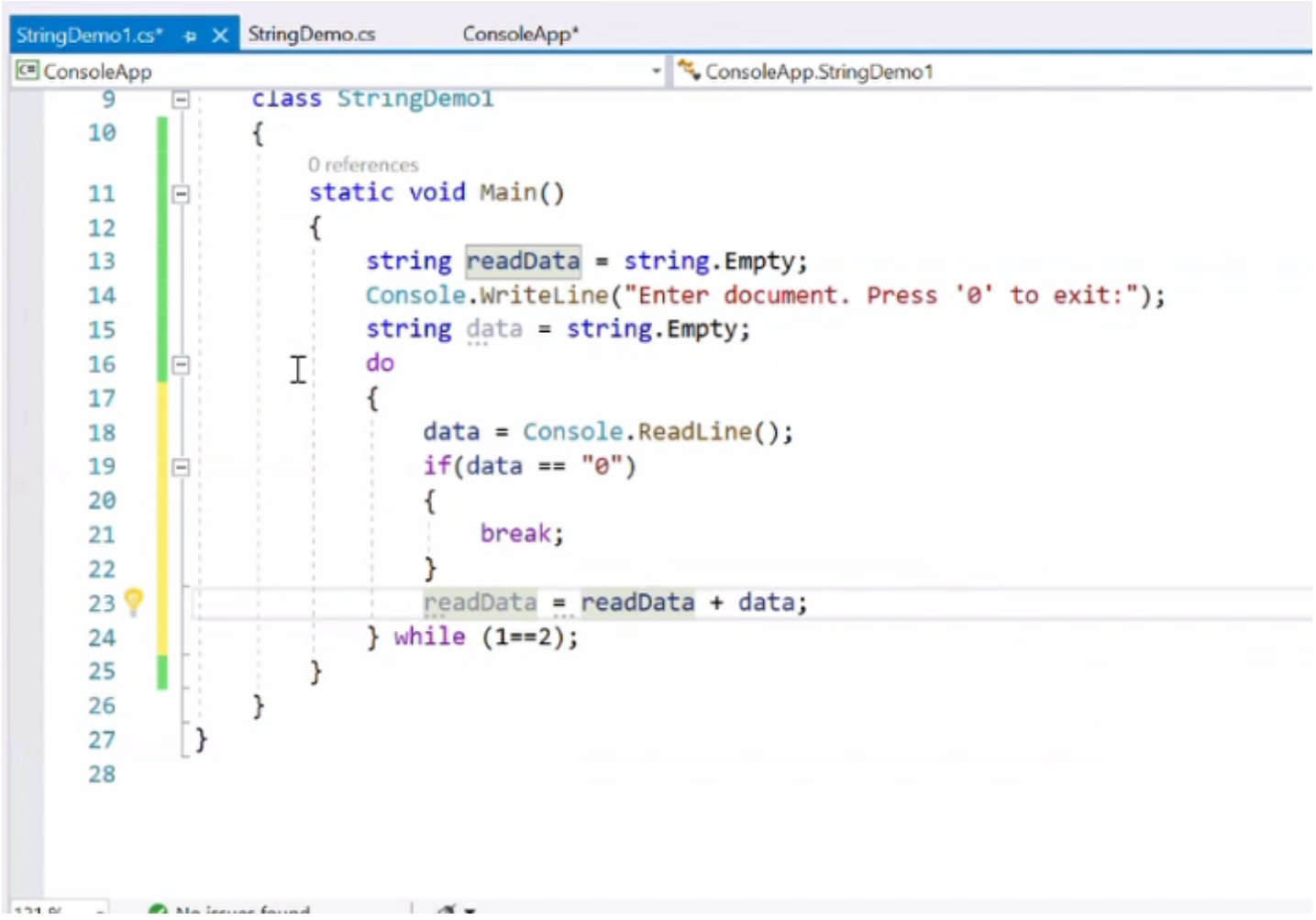
That's a reason we have **.Equal()** method using which we compare in java world because we want to compare not the instances rather the value

So, to compare value in java we always go with the .Equals () method

But where as when you come down to .NET that's not the case there you use .Equals() or == it always compare the values

But in .Net when you modify string it will always break, it will always store data in new reference , Every time you keep adding the data it will treat that as a new string

That's a reason if you are asking the user to enter some string and trying to store that in a string

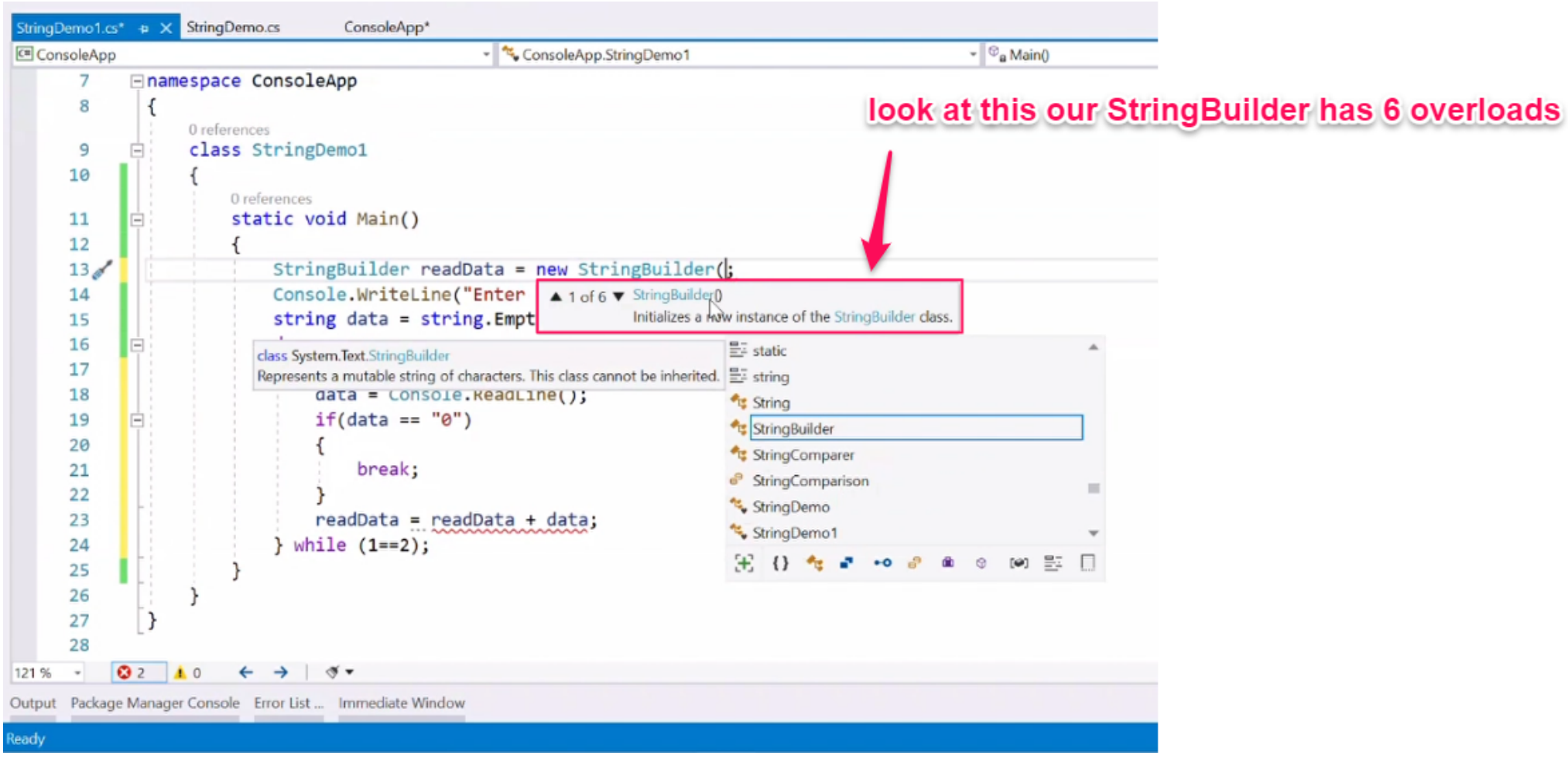


So, if you have such senario never use string

Maybe you are writing a particular program which will read a particular data from the file, you are reading lot of content from the file and you are trying to store it in a string variable, so you should not use a string
because as and when you are reading some content it keeps growing and that will be a **Performance head**
So, instead of putting that to a string you can you can always put it in a

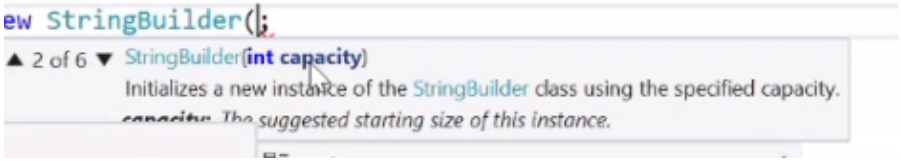
StringBuilder

We can take one string or multiple string and put it to a string builder



StringBuilder without any parameter, So this is a constructor

So **Constructor** **are those special methods which are automatically been called whenever you create a new instance/ new object**



you can specify what will be the initial capacity, initial capacity meaning like if you know how many word you can capture so you can pass that as a initial capacity