Getting into more Variables

Eng : yousif mohamed

Table of Content

1- Intro to arrays

2- Array with Memory

3- Declaration and initialization

4- Multi-Dimensional array

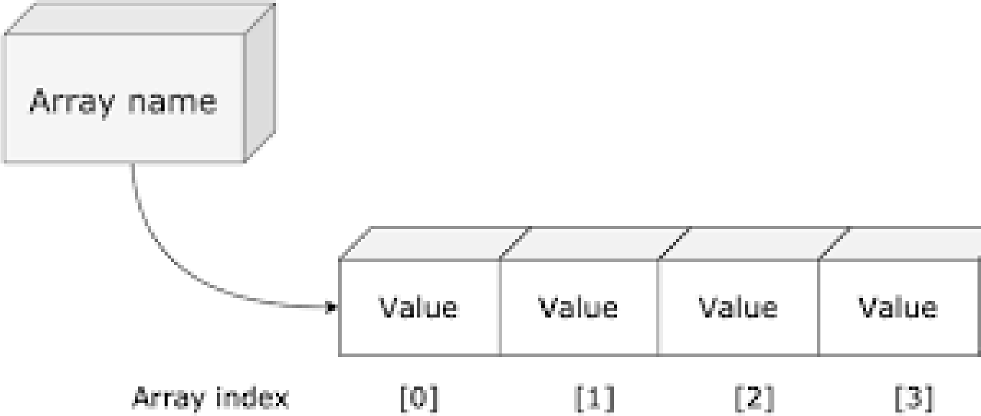
5-jagged arrays

6-indices and Range

7-Bound Checking

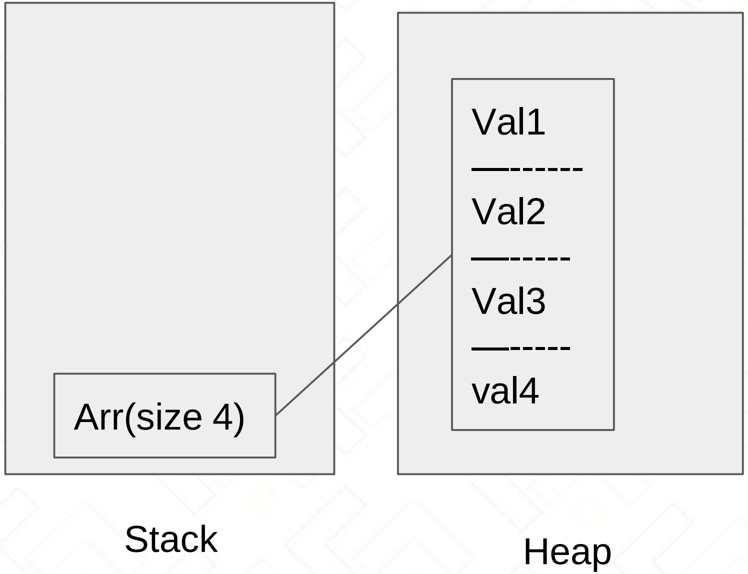
What is an array ?

an array is a collection of elements of the same data type, stored in contiguous memory locations. Arrays are a fundamental data structure that allow for efficient storage and access to collections of data. Array is a reference type



Array with memory

As the array is Reference Data type so its declaration name is stored in stack and values in heap but this time they are stored together as block together and we can access each one of them with it’s **index** (Order)



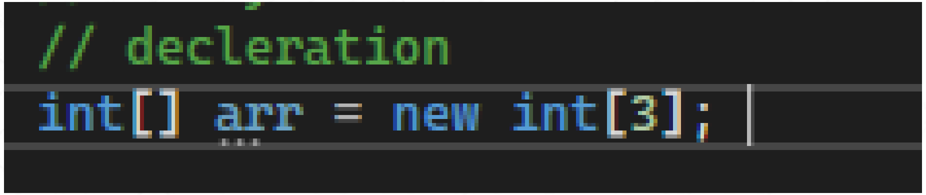
Array Declaration and initialization

As we have said in our Golden Rule declaration of Reference is as Follow :

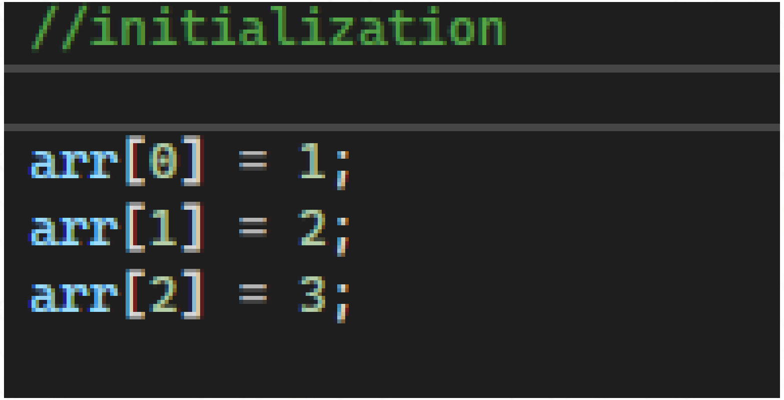
<Datatype>[] <varname> = <new > <datatype[size]>

By typing this declaration statement you will reserve a place in memory (stack and heap) with space acquired to hold values

In array we start ordering our values start from index 0 so if we want to start storing we will start from index varName[0]



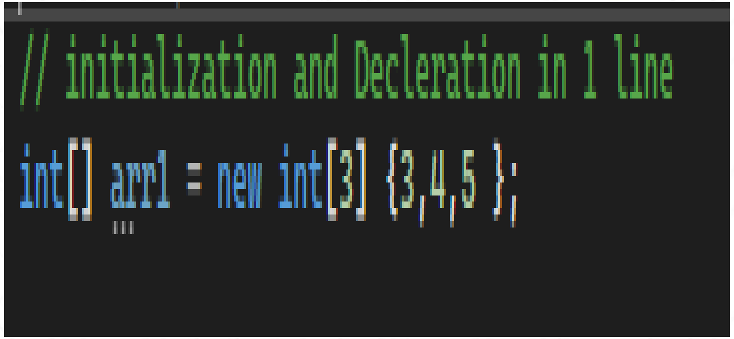
```
// decleration
int[] arr = new int[3];
```



```
//initialization

arr[0] = 1;
arr[1] = 2;
arr[2] = 3;
```

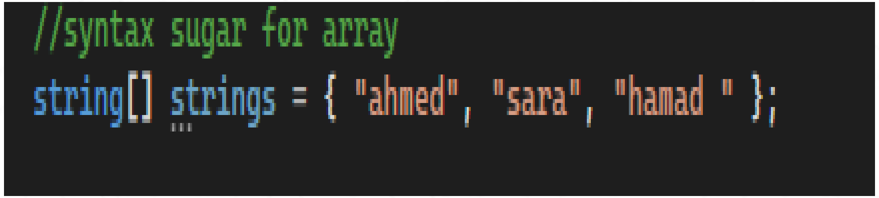
Array Syntax Sugar

Initialization and Declaration in One Line



```
// initialization and Decleration in 1 line
int[] arr1 = new int[3] {3,4,5 };
```

c# team always wants to make it simplified for us so as syntax sugar we can declare array with other form



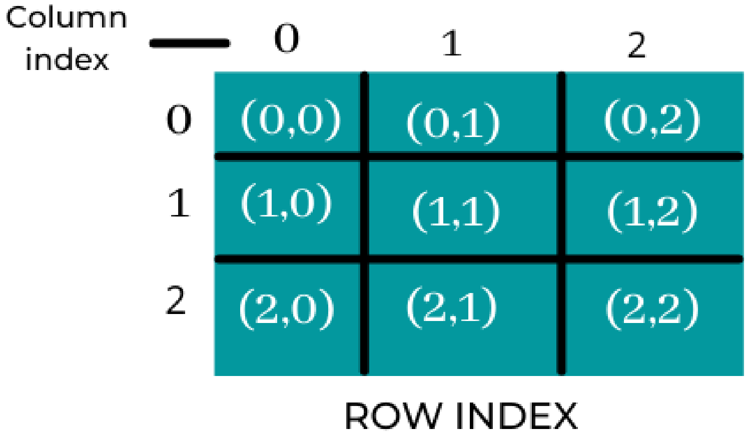
```
//syntax sugar for array
string[] strings = { "ahmed", "sara", "hamad " };
```

Brain Force warning

- While you are dealing with array take care of your total Size and your elements you want to add
- Array’s should be explicitly Defined (var keyword) we will test it in code in Practice

Multi-Dimensional array

Arrays in C# can be one-dimensional or multidimensional, and can also be jagged arrays, which are arrays of arrays. Here's an example of a two-dimensional array:

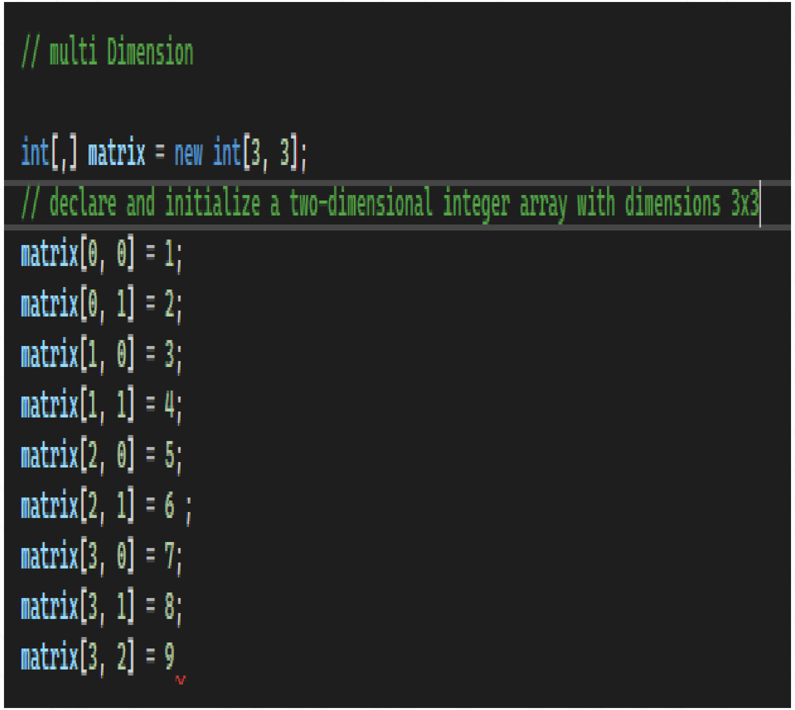


Column index	0	1	2
0	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)
2	(2,0)	(2,1)	(2,2)

ROW INDEX

Multidimensional declaration

To Declare a multi Dimensional array or in other way we call it matrix . we just need to Determine what is our dimensions in our case we need 2 dimensions array (rows and columns) so we can Declare it as follow .

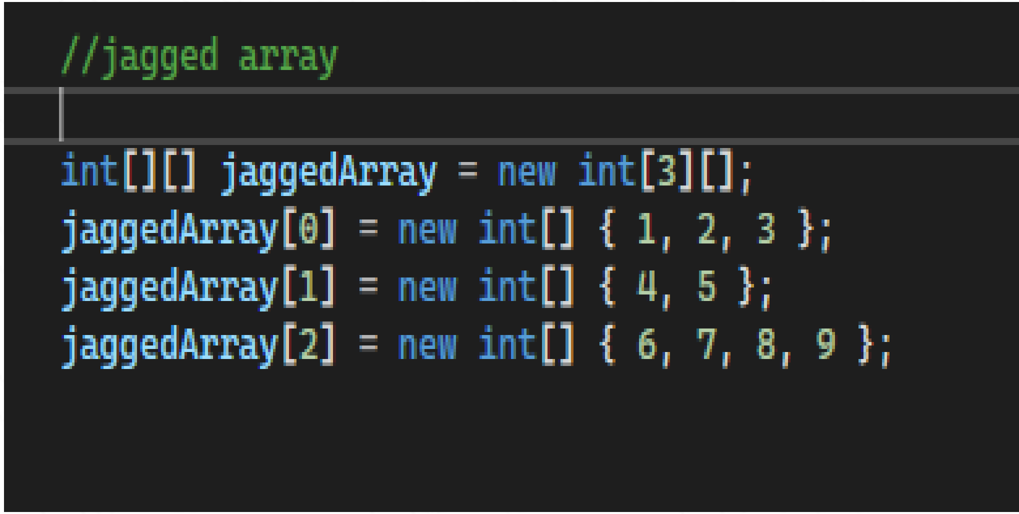


```
// multi Dimension


int[,] matrix = new int[3, 3];
// declare and initialize a two-dimensional integer array with dimensions 3x3
matrix[0, 0] = 1;
matrix[0, 1] = 2;
matrix[1, 0] = 3;
matrix[1, 1] = 4;
matrix[2, 0] = 5;
matrix[2, 1] = 6 ;
matrix[3, 0] = 7;
matrix[3, 1] = 8;
matrix[3, 2] = 9
```

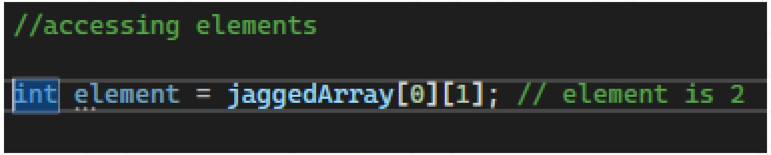
Jagged Array

a jagged array is an array of arrays, where each array can have a different length. Jagged arrays are useful for representing data that has irregular or varying dimensions.

To create a jagged array, you first declare the array variable, and then you can initialize each sub-array with its own set of values

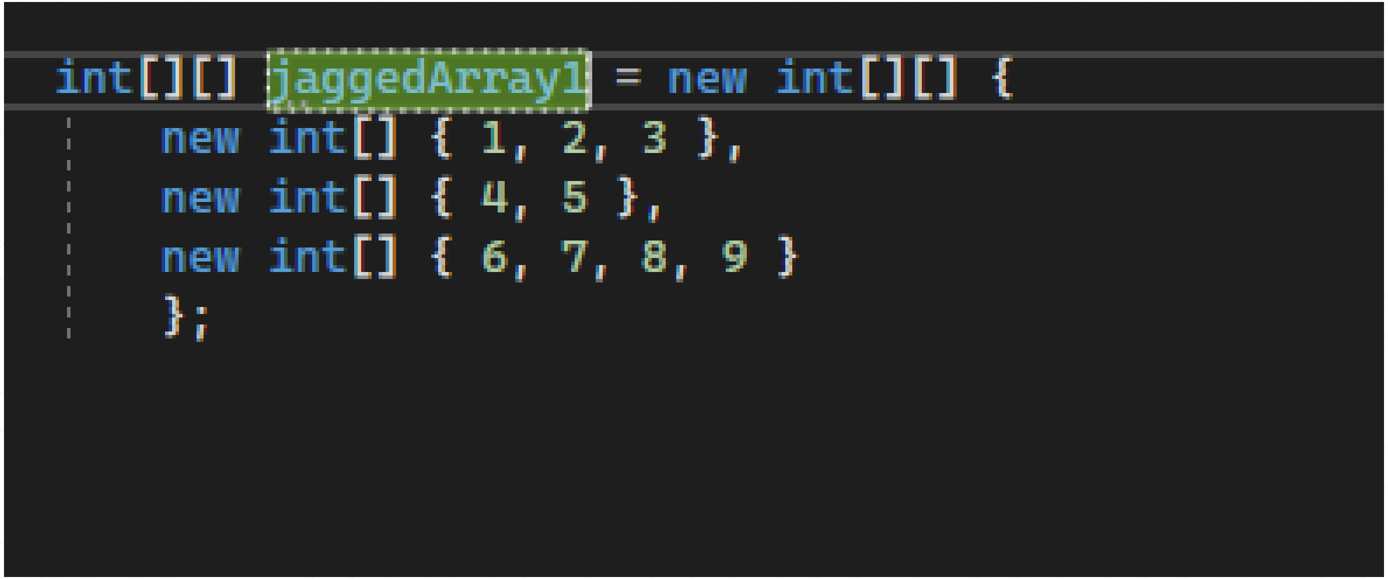


```
//jagged array

int[][] jaggedArray = new int[3][];
jaggedArray[0] = new int[] { 1, 2, 3 };
jaggedArray[1] = new int[] { 4, 5 };
jaggedArray[2] = new int[] { 6, 7, 8, 9 };
```



```
//accessing elements

int element = jaggedArray[0][1]; // element is 2
```

Jagged array using initializer list

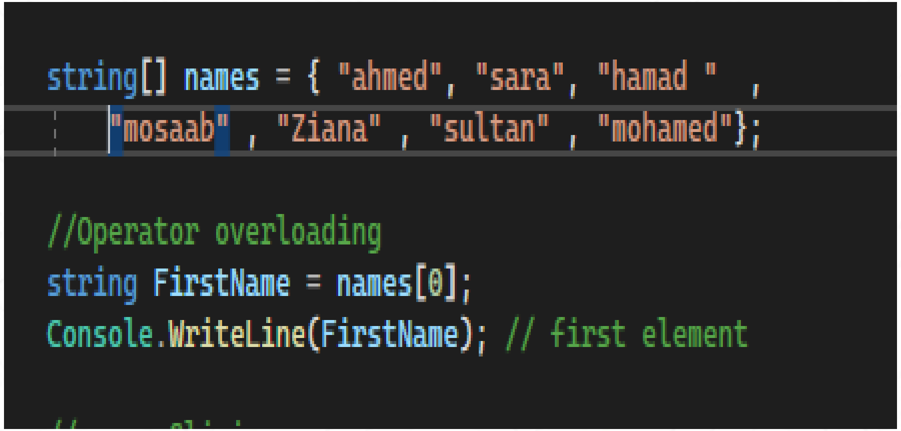


```
int[][] jaggedArray1 = new int[][] {
        new int[] { 1, 2, 3 },
        new int[] { 4, 5 },
        new int[] { 6, 7, 8, 9 }
        };
```

Indices and Ranges

As we have talked together you can access values inside the array by it’s Index and our index start from 0 .

But what if i want to take a range of values to another array or to print them out , that is what array slicing is Let’s Discuss it together in the next Slide



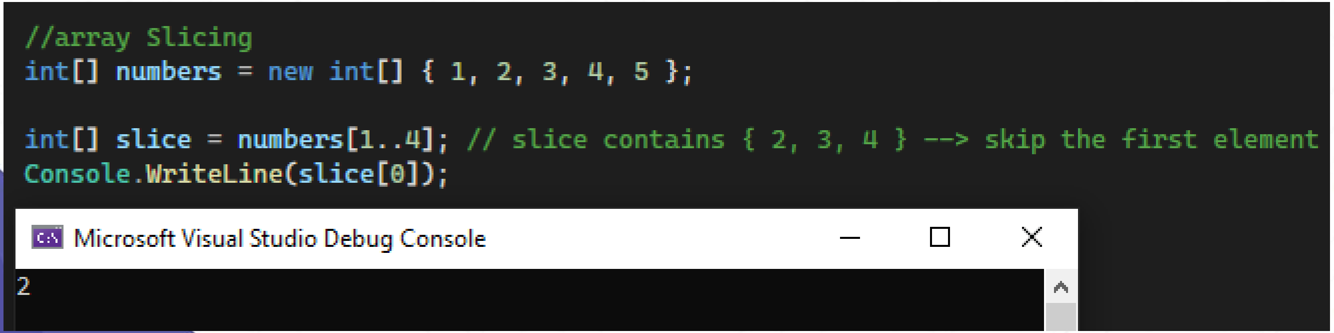
```
string[] names = { "ahmed", "sara", "hamad "   ,
    "mosaab" , "Ziana" , "sultan" , "mohamed"};

//Operator overloading
string FirstName = names[0];
Console.WriteLine(FirstName); // first element
```

Array slicing

Array slicing is used for destructing your array into smaller pieces and it is done in c#using range operator

Range operator [from **..** to] used to slice and your array to sub parts so let’s Discover Together how we can manage it together .

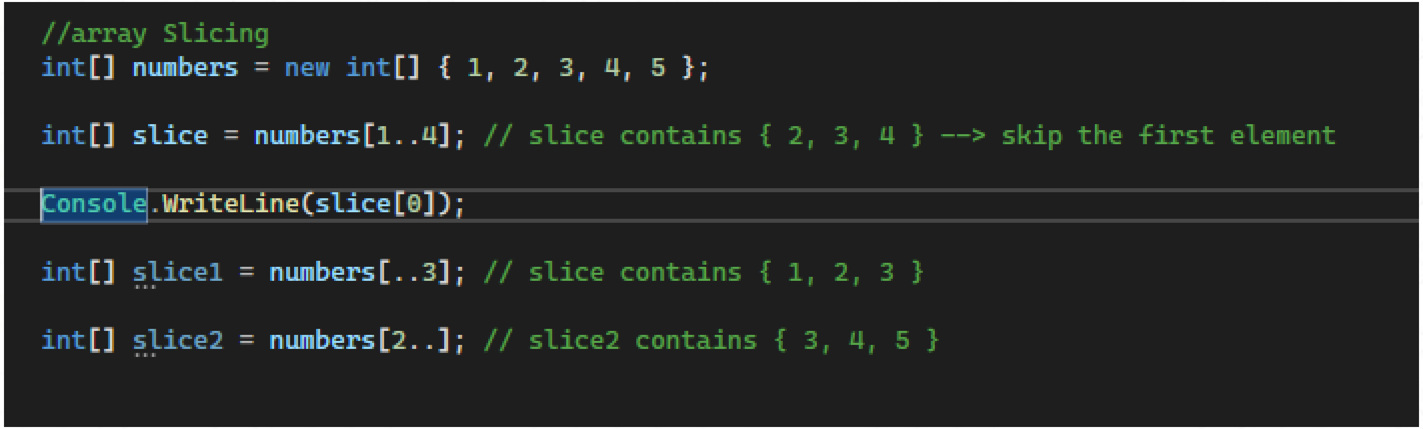
If you didn’t provide from by default compiler will consider it from 0 index



```
//array Slicing
int[] numbers = new int[] { 1, 2, 3, 4, 5 };

int[] slice = numbers[1..4]; // slice contains { 2, 3, 4 } --> skip the first element
Console.WriteLine(slice[0]);
```

```
2
```

Array Slicing mode Advance



```
//array Slicing
int[] numbers = new int[] { 1, 2, 3, 4, 5 };

int[] slice = numbers[1..4]; // slice contains { 2, 3, 4 } --> skip the first element

Console.WriteLine(slice[0]);

int[] slice1 = numbers[..3]; // slice contains { 1, 2, 3 }

int[] slice2 = numbers[2..]; // slice2 contains { 3, 4, 5 }
```

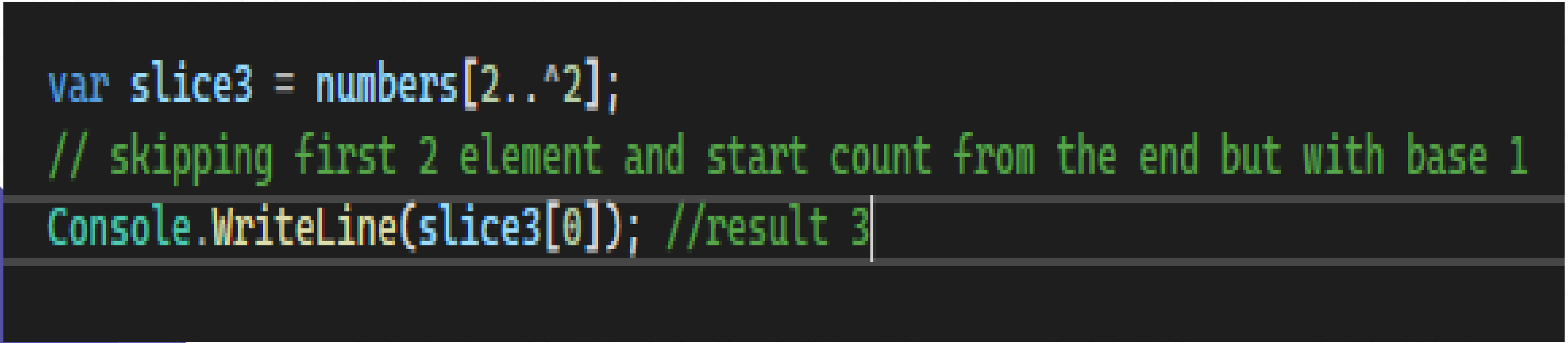
slice 1 : we didn’t provide the start so by default he took it till the end index

Slice2 : when we provided a start he will skip it and start from the start+1

Reverse Slicing ‘ ^ ’ chapeau notation :

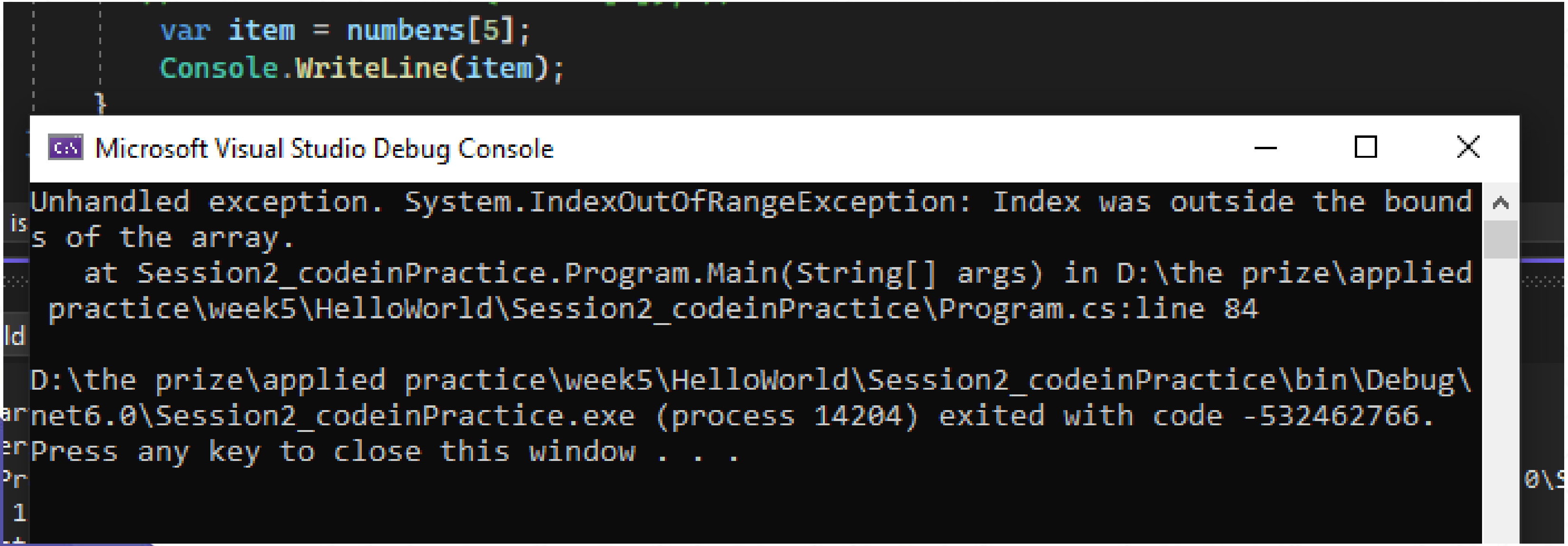
If we want to skip from the beginning of the array and count from the Last we can use chapeau “ ^ ” .

When compiler see the ^ notation compiler will start to detect your target from the end let’s take an example



```
var slice3 = numbers[2..^2];
// skipping first 2 element and start count from the end but with base 1
Console.WriteLine(slice3[0]); //result 3
```

Out of Boundary Exception



```
var item = numbers[5];
Console.WriteLine(item);
}
```

```
Unhandled exception. System.IndexOutOfRangeException: Index was outside the bound
s of the array.
   at Session2_codeinPractice.Program.Main(String[] args) in D:\the prize\applied
 practice\week5\HelloWorld\Session2_codeinPractice\Program.cs:line 84

D:\the prize\applied practice\week5\HelloWorld\Session2_codeinPractice\bin\Debug\
net6.0\Session2_codeinPractice.exe (process 14204) exited with code -532462766.
Press any key to close this window . . .
```

Let’s CodeinPractice Together

Session Recap

Any Questions ?

Thank You

Email : youssef.mohamed@amit-learning.com

Linked-in : https://www.linkedin.com/in/youssif-mohamed-450795157/