

Session 3_Part 1

* String AS BuiltIn Class
IS Array of Char

Reference Type

String Name;

① Declare For Reference of TYPE
'String'

② // This Reference 'Name' IS Refering
To Default Value of RT = Null

③ // CLR Will AllocatE

4 Bytes in Stack

↳ For Reference Type 'Name' as
address.

5 char

Name = new String ("Ahmed");

Take 2 Bytes Char + char --

// CLR Will AllocatE 10 Bytes

in Heap

// Will initialize For AllocatE BYT

BY default Value For char

// Call User Defined constructor
then String ("Ahmed")
 ↓ Set in each char value
// Will return address of object
in Heap to stack as 4 Bytes.
Contain Address
Name = "Ahmed"; // Syntax Error

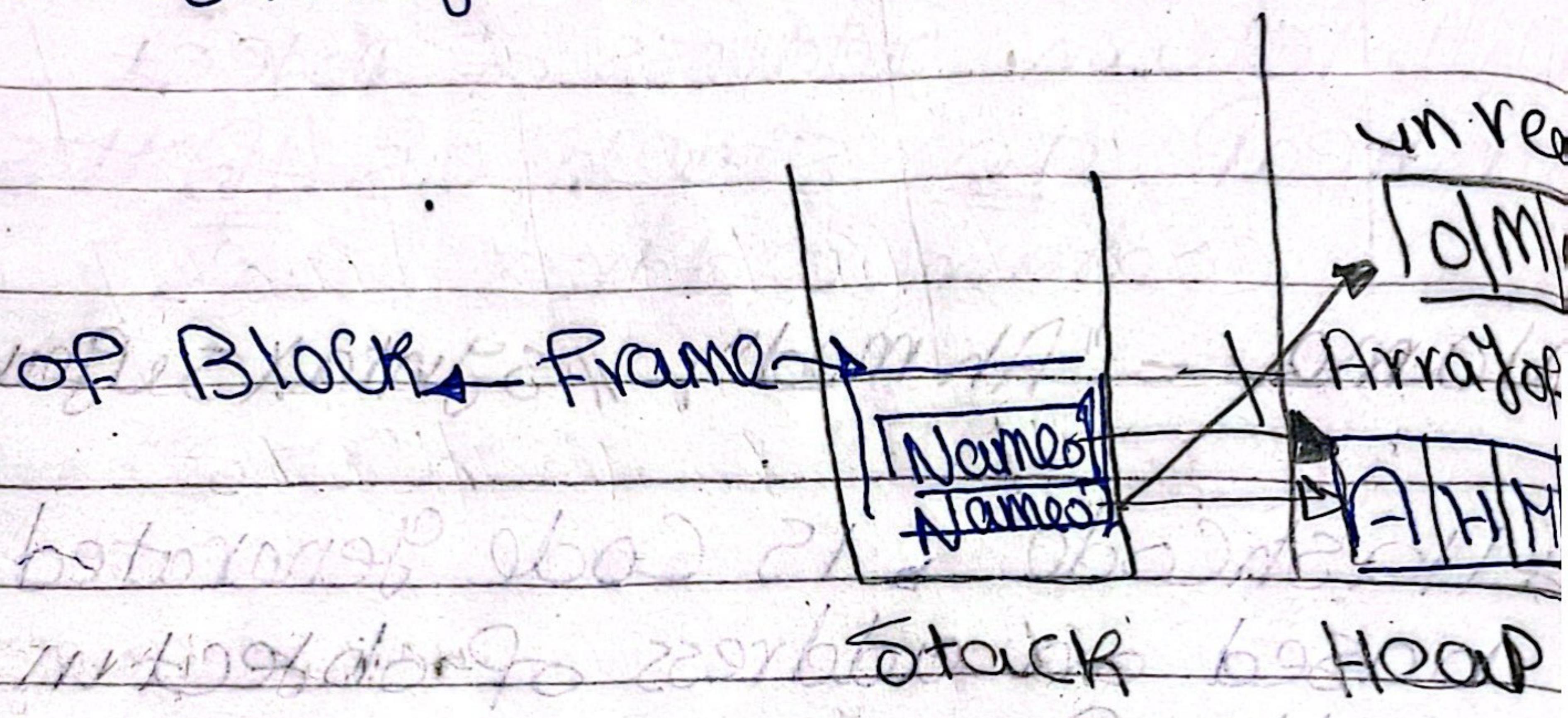
* HashCode → IS Code Generated
based on address of object in
Heap.

note:

→ با بابا انجاز انت و لا تنسى
algobj لكي يدخل في الحاله obj (في المثلث)
ـ على لو اول و او اول
so :- Contain same HashCode
in IL When check string
make load string
↓
Check if string
object does not have its
signature is not found

String Name\ = "Ahmed"; i

```
String NameOfA = "Omax";
```



Name 02 = Name 01 ;

Name of 1 = "yesmin" i

Name of student Sitiqur Ahmed

حسن جمعیت علی داده امکان ایجاد

Ayyoub Panchasari, Is Ahmed 64

Fixed length

- Create new object To "Yasmin" with new size in Heap

89

String is immutable type

→ String is immutable type
Can't changed

↓ So ~~char~~ ~~String~~ ~~Print~~

Name01 = "Yasmin"

but Name02 ~~is~~ still = "Ahmed"

~~char~~ ~~array~~ ~~of~~ ~~size~~
fixed length

String Message = "Hello";

→ This Array

Message + = "Ahmed"; will
unreach

Hello String Use ~~new~~ ~~size~~ ~~fixed~~
new size ~~new~~ ~~fixed~~ ~~use~~
↳ Hello Ahmed ~~will~~ → ~~new~~ ~~size~~

String ~~also~~ ~~new~~ ~~size~~ ~~use~~ *
"new" ~~new~~ ~~size~~ ~~use~~ ~~new~~ ~~size~~
|| (Garbage) ~~in~~ ~~Heap~~ ||

Garbage Collector

unreachable in Heap || ~~it~~ ~~is~~ ~~not~~ ~~defined~~
no ~~use~~ ~~new~~ ~~size~~ ~~use~~ ~~new~~ ~~size~~
in

Wahid said, And we will do all this
by ~~String~~ ~~String~~ ~~String~~

* String Builder

IS String but internally
~~is string~~ ~~is string~~ ~~is string~~

using System.Text;

~~to~~ of String Build

StringBuilder Msg; ~~String~~

Declare . For Reference TYPE

"StringBuilder"

// Reference Msg is Refering to
default value of RT → Null

// CLR Allocate → 4Bytes in stack
→ 0Bytes in Heap

Message = new StringBuilder("Hello")

Message . APPend ("Ahmed");

↓
// Add from end

existing string class

HashCode (int)

* StringBuilder is Mutable String
internally → is linked list of chars
→ flexible (Var)
size

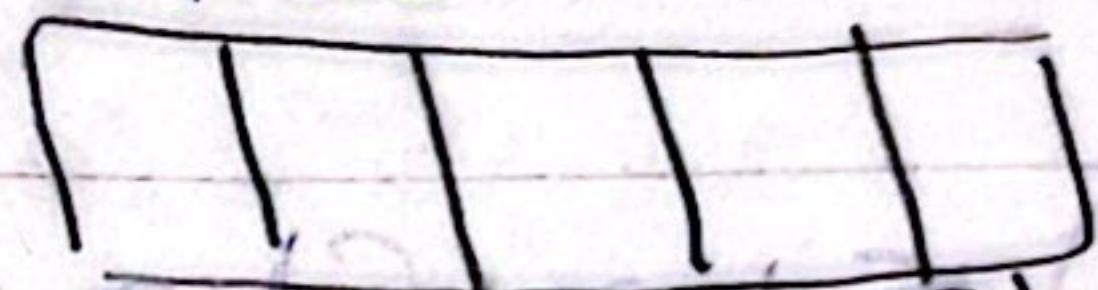
To reach reach in any char

Array

Take one step

O(1)

linked list



To reach char

O(n)

* StringBuilder Methods

① String. APPend (Word);

② String. APPendlne ();

// write then → ~~use line~~ newline

Str. APPend ("Hello Ahmed");

Str. APPendlne (" Nasr");

Str. APPend ("Your Age 22");

// Hello Ahmed Nasr

// Your Age 22

③ Str. Remove (0, 5);
loop From $\underline{4}$ to remove
first

// Hello

④ Str. Insert (0, "Hi");
insert $\underline{0}$ to $\underline{1}$ \rightarrow insert $\underline{1}$ to $\underline{2}$

⑤ Str. Clear();

String $\underline{\underline{1}}$ to $\underline{\underline{1}}$ is good

⑥ Str. AppendFormat();

"{0}:{1}", true, 'A'

: always = 0 until $\underline{\underline{1}}$ to $\underline{\underline{1}}$ left $\underline{\underline{1}}$

// True : A

⑦ Str. AppendJoin();

";, 'Hamda', 'Ahm'

utopia is $\underline{\underline{1}}$ to $\underline{\underline{1}}$

Concatenated \rightarrow

⑧ Str. replace()

⑧ Str.length() // Actual length

⑨ Str.GetChunks()

Read only memory

IS extract data of Constant
BuilderString IS Modified while
Chunk enumeration IS incomplete.
result is undefined

chunks of characters

represented in Readonly Memory

Created From This String Builder

Instance.

* Array one D

int [] Numbers = {1, 2, 3}

If you want updated

Numbers = {3, 4, 5} X blc

Must ~~87 + m = 88~~ ~~88~~

.. Numbers = new int [] {3, 4, 5};

Numbers.length

// return length of array

Numbers.Rank

// return dimension of array

// Array IS Class - Reference
TYPE

int[] numbers;

// Declare For Reference of
"Array of integer".

// Referring to Default Value of

RT => null

// This Reference (numbers) can
refer to an object TYPE
of integer Array.

// CLR will allocate

4 Bytes

in Stack as

address

0Bytes in
Heap

Numbers = new int[3];

// CLR will Allocate

12 Bytes in Heap,

// initialized with Default Value
of int = 0

// Make object refer to Address

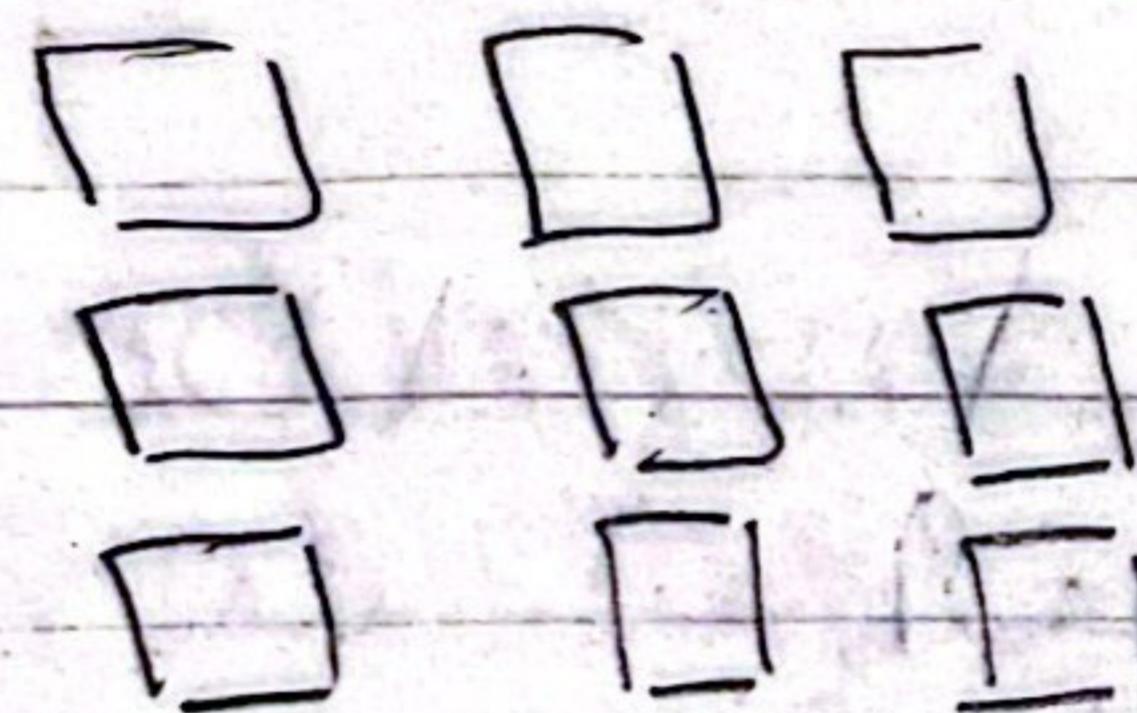
* Array Two D

① Rectangle Array

② Judged Array

* Rectangle Array

A 5 Table



of Columns

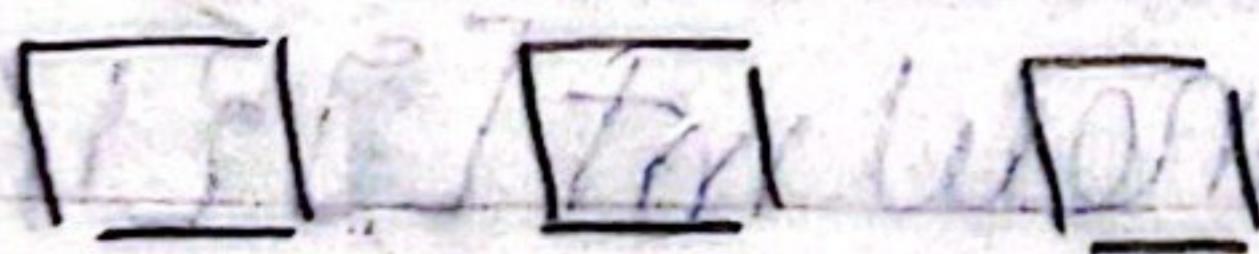
in each

row equals

Null's value, no cell is

Column = 3 Space is 2

* Judged Array AS Connection
AS Document



of Columns

in each

row not

equal

Null's value, no cell is

space بونس اصل

Array کوئی جگہ

int [] Numbers;

Numbers = new int [3] {1, 2, 3}
allocate ↕ ↕ ↕

or

Numbers [0] = 1

Numbers [1] = 2

Numbers [2] = 3

or

int [] Numbers = new int [3]

use allocate ↕ ↕ ↕

or

int [] Numbers = new int [] {1, 2, 3};
use allocate ↕ ↕ ↕ size given

or

int [] Numbers = {1, 2, 3};
// Syntax Sugar

→ For Represent array using
① For loop, ② foreach

```
For (int i=0; i<Numbers.length; i++)  
    {  
        Console.WriteLine(Numbers[i]);  
    }
```

* Advantage of Array: That Take one step to reach for any element.

* Disadvantage: → Array is fixed length

```
int[] Numbers;
```

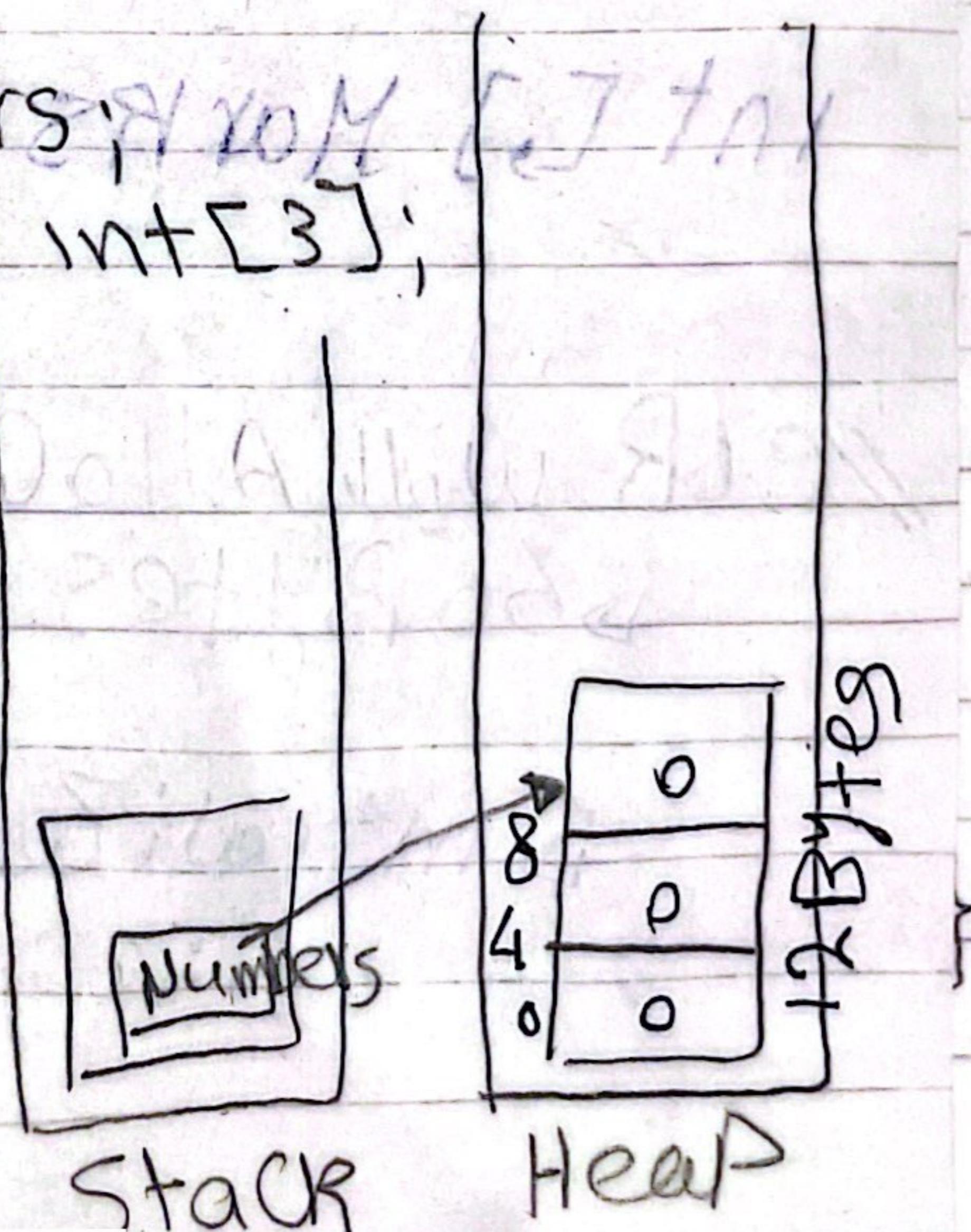
```
Numbers = new int[3];
```

↓
指向
指向元素

Number [2];

指向 0

0 1 2 3



Numbers = new int [3];

Number [3]; //will throw exception

→ If you want add to Array
will Create new Array with
doubtlicate size, delete old
array.

* Two Dimension Array

Rectangle Array

int [,] Marks

→ Two D Array of int

→ Three Dimensional

int [,] Marks = new int [3,5];
rows ↘ ↗

// CLR will Allocate

→ 60 Bytes at Heap (15 elements
* 4 Bytes)

→ Initialized with default int

Marks[1,2],
J 0 2 3 4

Marks[0,0]=100,*

Marks[0,1]=80,

Marks[0,2]=90;

int[,] Marks = new int[3,5]

↙ { {1,2,3,4,5} } { {6,7,8,9} } { {0,1,2,3,4} } ;

Columns of row1 row2 row3
of row1

↳ Marks.length[i]

// to show length of array

Marks.Rank

// to show Dimension of array

→ For (int i=0; i<3; i++)

↙ For (int j=0; j<5; j++)

↙

~~Code written~~

~~Marks[i,j] = int.Parse(Read);~~

$O(n)^2$

Console. Beep(3223, 3000);

Frequencies ↗

Time of sec ↗

* Issues solved

① Marks. GetLength(0) Based
return ↗ D ↗
length of Dimension ↗ row

Marks. GetLength(1)

→ DGL

Columns

Will become not static

② Issue

Marks[i,j] = int.Parse(Console.
Read());

solve

Flag = int.TryParse(Console.ReadLine(),
out Marks[i,j]);

int.TryParse

* Array with using one loop

→ If you have 2D Array

int[,] Marks = new int[3,5];

i Row(i) Columns (i)

		Column no	Column of	no column
0	0	row0		
1	0			1
2	0	data		2
3	0			3
4	0	row0		4
5	1			
6	1	row1		1
7	1			2
8	1			3
9	1			4
10	2			0
11	2			1
12	2	row2		2
13	2			3
14	2			4

Row information

* Display elements of 2D Array by one loop by

$$\text{row} = i / \# \text{ of Columns}$$

$$\text{columns} = i \% \# \text{ of Columns}$$

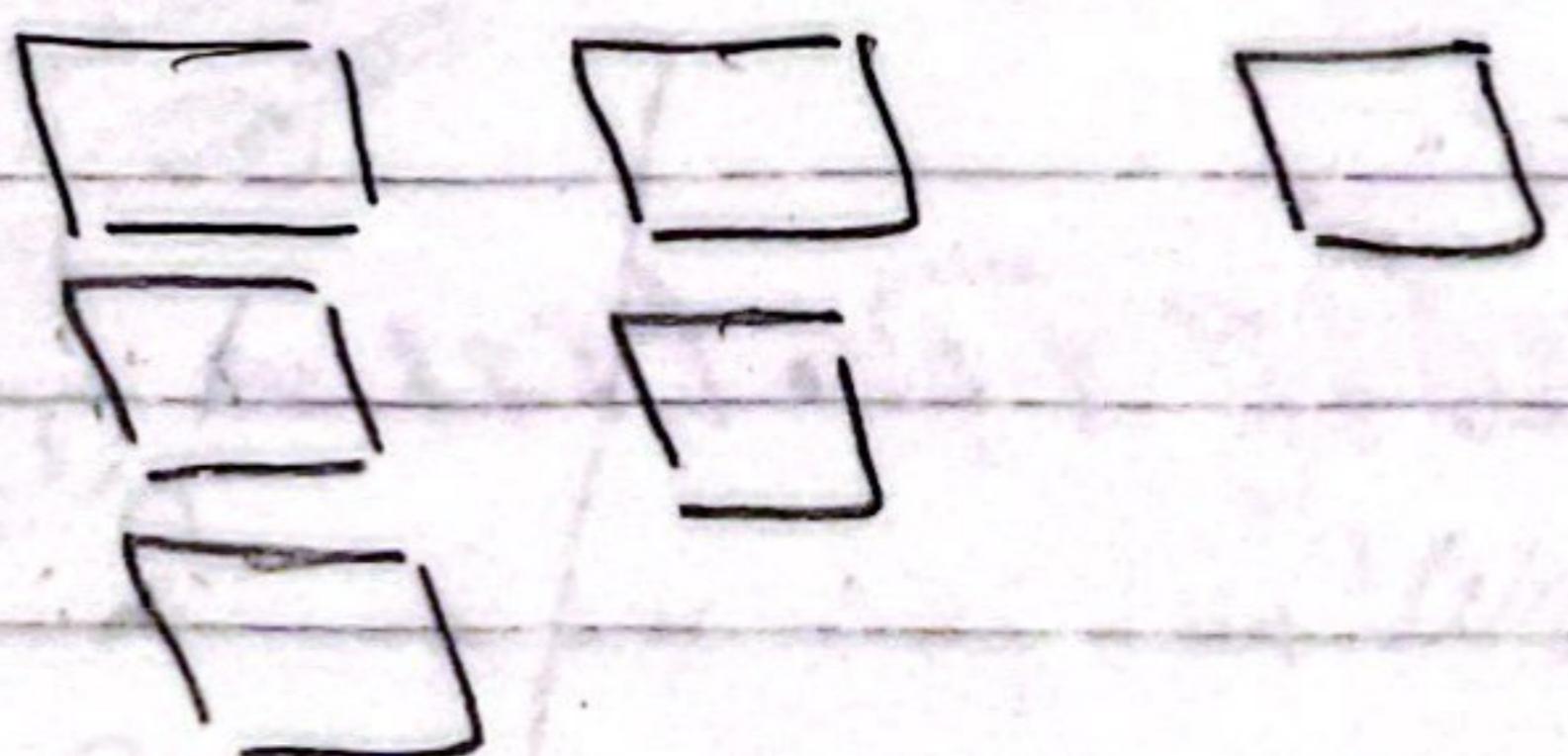
```
For (int i=0; i < Marks.length; i++)
```

```
{  
    Console.WriteLine(Marks[i].GetLength(1),  
                      i % GetLength(1))  
}
```

O(n) improves complexity from
O(n²) to O(n)

* Jagged Array

of column not equal
in each row



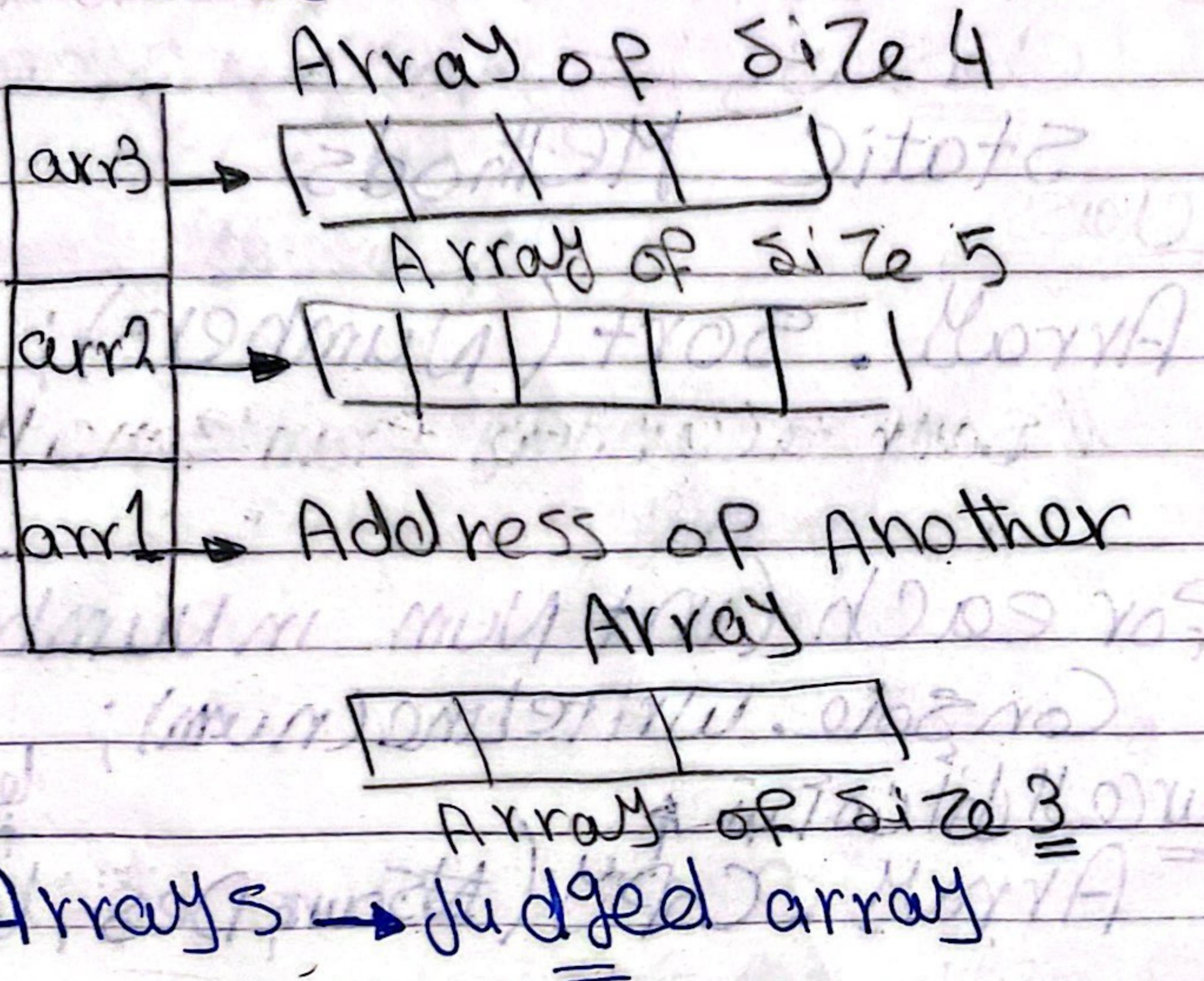
বিজ্ঞান বিজ্ঞান
বিজ্ঞান
বিজ্ঞান

Internally \rightarrow is 1D Array

\downarrow have

Multi Array

* 1 D Array → each element consider reference for array & with any size → 1D



int [] [] Marks = new int [3] [];

Arrays → arrays ↳ array
array of 3

Marks [1] = new int [2] {1, 2};

Marks [0] = new int [3] {3, 4, 5};

Marks [2] = new int [1] (5);

* Array Methods

① Class Member Methods

Class Isignle Brdggo Method

Static Methods

Class

Array. Sort (Numbers);

// Sort according from small to

big

For each (int Num in Numbers)

Console.WriteLine (num);

Source Jpg Cofn

= Array. Copy (~~source~~, Des, length)

(big COPY) <= (small)

int [] Arro1 = { 1, 5, 6, 7 };

int [] Arro2 = ~~new~~ int [2];

Array. Copy (Arro1, Arro2, 2);

Array. Clear (Arrro1) ; Arrro1 . Arrro1
default 11 element جیسے 11 ہیں
لہوں، ہر ایک position

Array. Constrained COPY ()
Jسے لے کر it is to J
element کو COPY

Array. ConstrainedCOPY (Arrro1, 1, Arrro2, 0, 2)
Arrro1 کو index 1 جیسے 1 ہے
index element 1 جیسے 1 ہے ←
Arrro2 کو 0 ہے ←

Array. CreateInstance (TYPEOF (int), 10);
1D of int جیسے length
//Create instance (object) From array

Generic OF TYPES

arrro3 = (int[]) Array.CreateInstance
of 1D جیسے 1D

Casting

Ex For 2D

int[,] Array2D =

(int[,]) Array.CreateInstance
(TYPEOF (int), 2, 4);

int index

Array . IndexOf (Numbers , 10);

العنوان في الارای 10 میں اے index اے جس کا

جس کا سائز 10 ہے اسے چھوڑوں

لے لیں index

Object Member Methods

Method Called through object
Non static Methods

int [] arr02 = { 1, 2, 3 } ;

int [] arr01 , index ;

arr02 . COPYTO (arr01 , 1);

arr02 میں COPY کیا جائے تو arr01 میں 1

arr01 میں 1

الاماں کی جائیں

arr01 . GetLength () ;

// return DIMENSION

arr01 . GetValue (1);

≡ // arr01 [1] ;

arr01 . SetValue (100 , 1);

≡ // arr01 [1] = 100 ;