Arrays: Collection of $\underbrace{\text{Homogenous}}_{\text{Same type of data}}$ Elmns.

int a = 10; // Hold single value

Arrays int[] a = {10, 20, 30};

$\downarrow$

1D

String[] st = {"ABC", "BCD"};

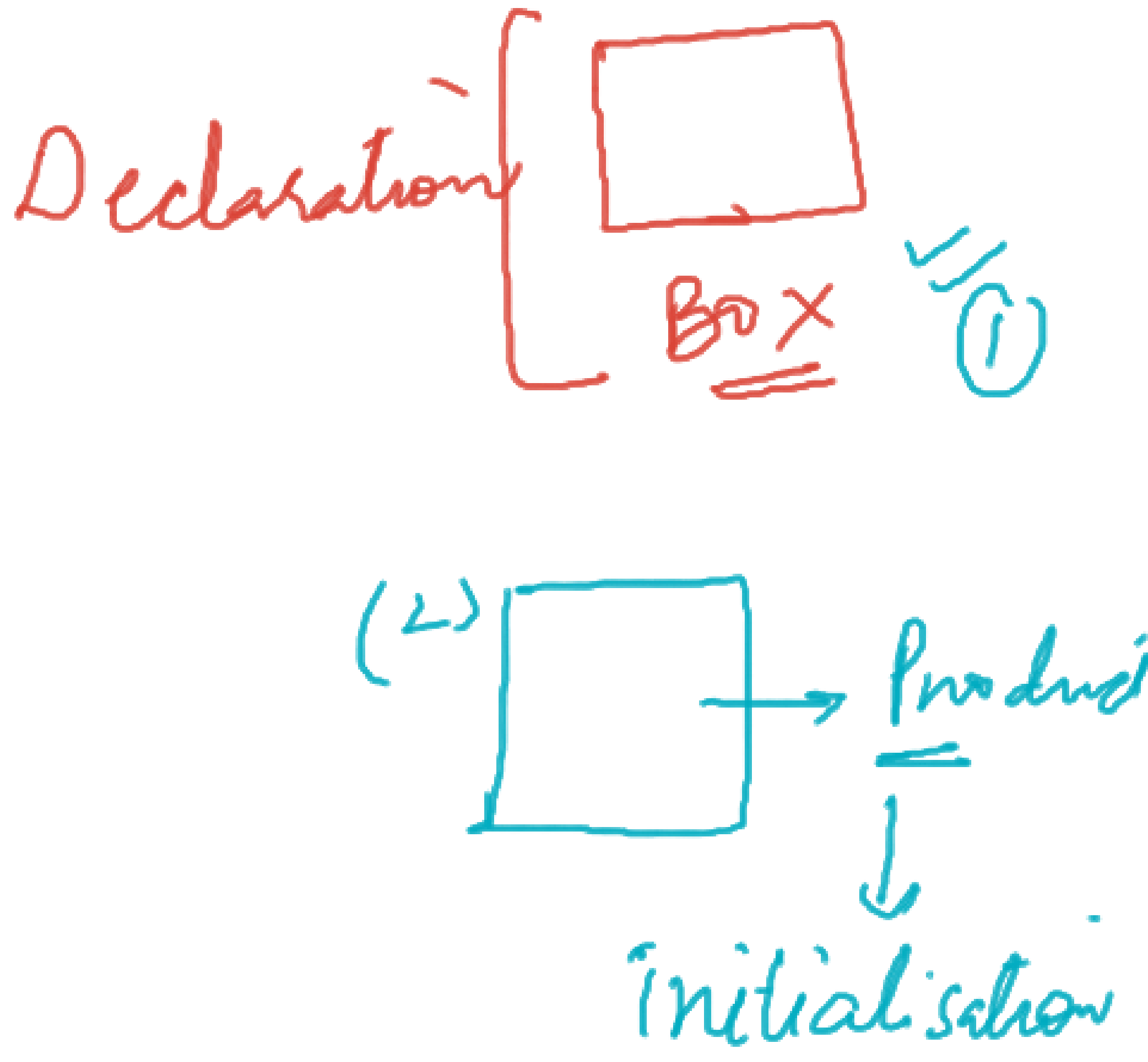
Array Declaration :

→ Array contains homogenous Elmns ·

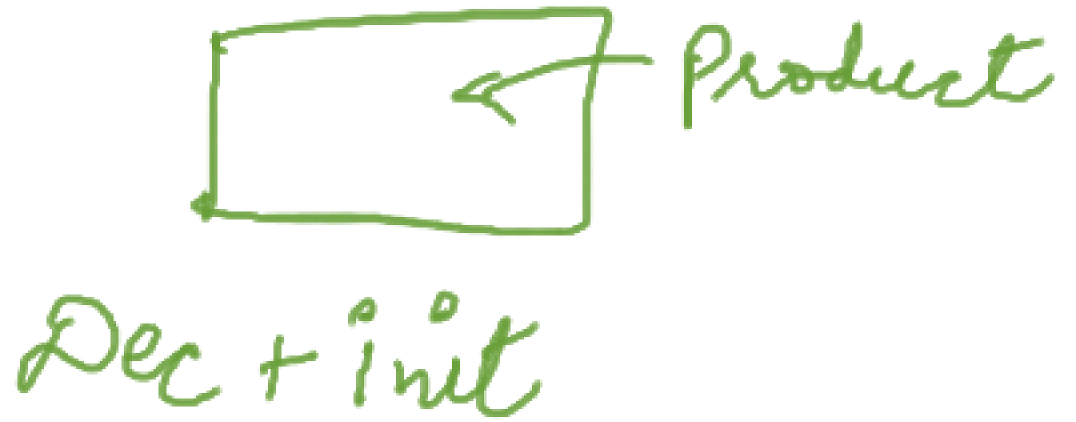
→ Size is fixed or we need to specify the size of an array.

Syntax

```
                              size of an Array.
int[ ]   arr  = new int [5] ;
  ↓        ↘
1D Array    var name
```



NewBox

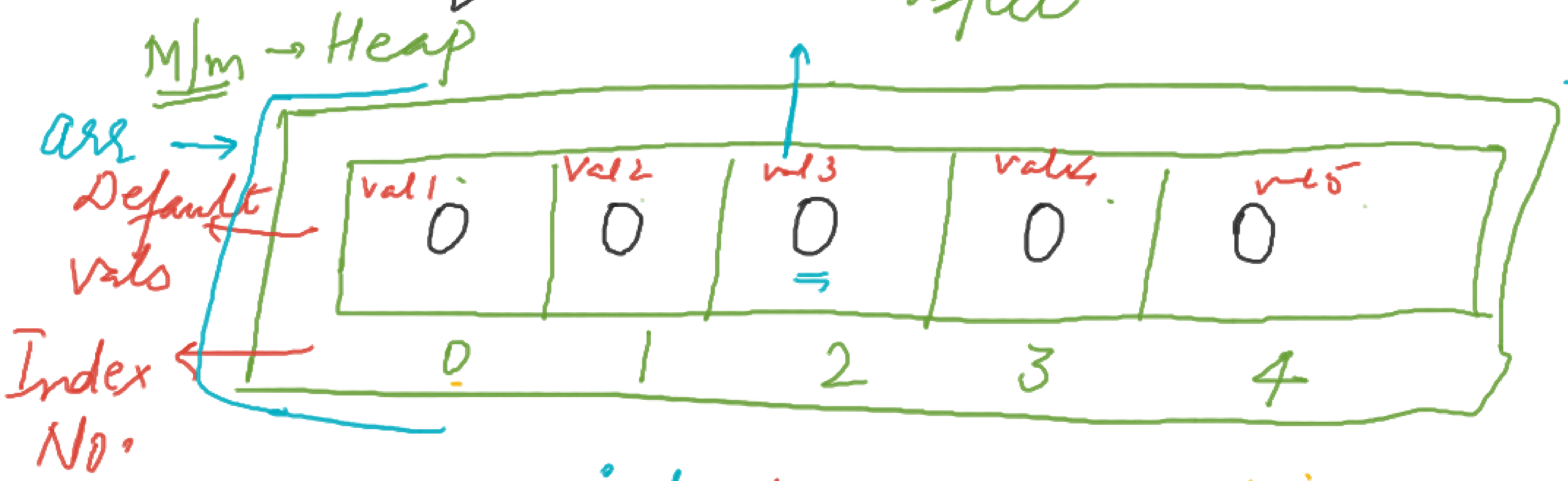


Varname / Array Name

```
int [] arr = new int [5];
```

object

→ Array act as an object.



→ Array always starts with index No - 0

```
arr[2];
```

→ indexNo.

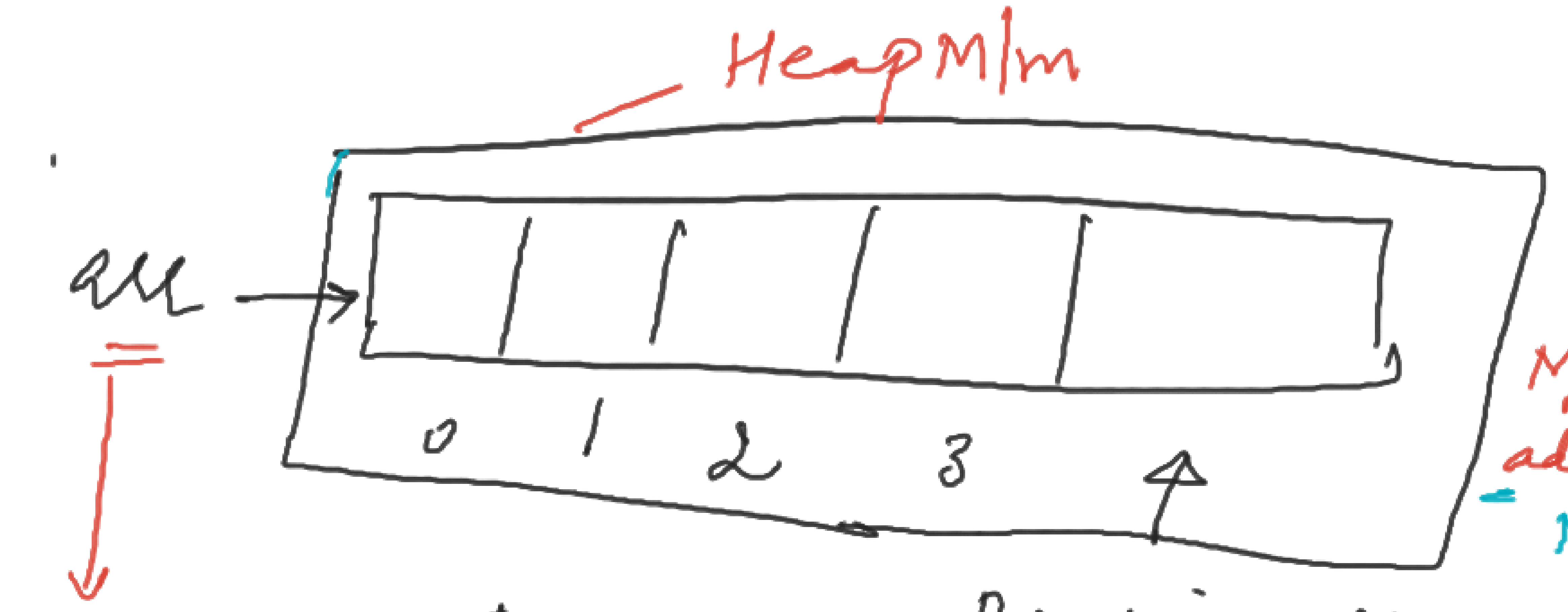
arr_name

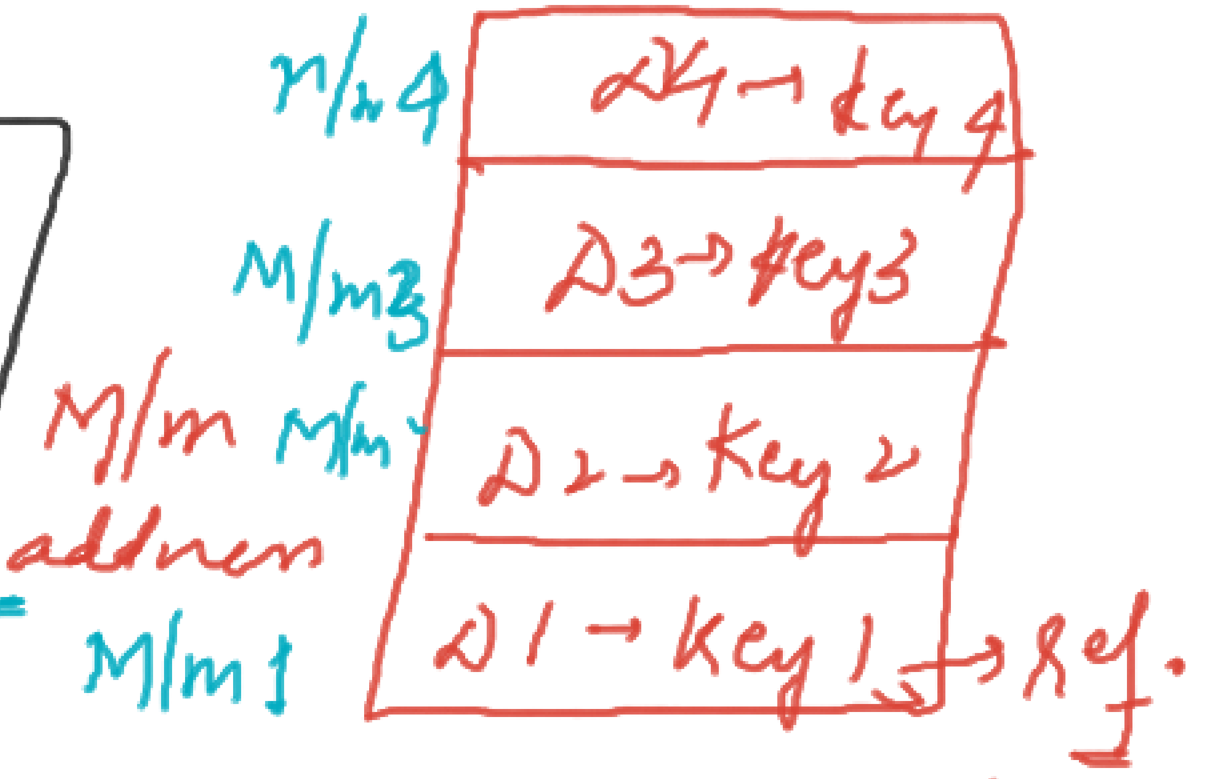
0 - 9

Obj, String - null

float, Double - 0.0

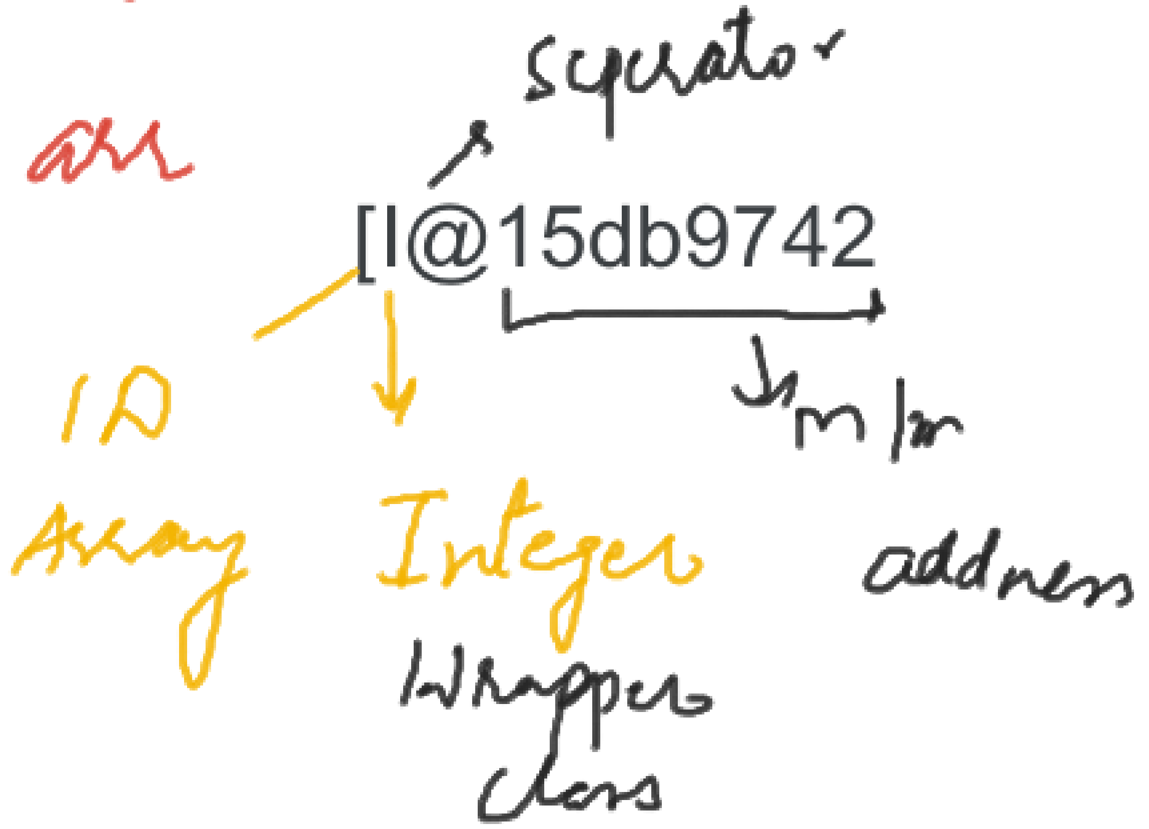
byte, Short, int, long → 0



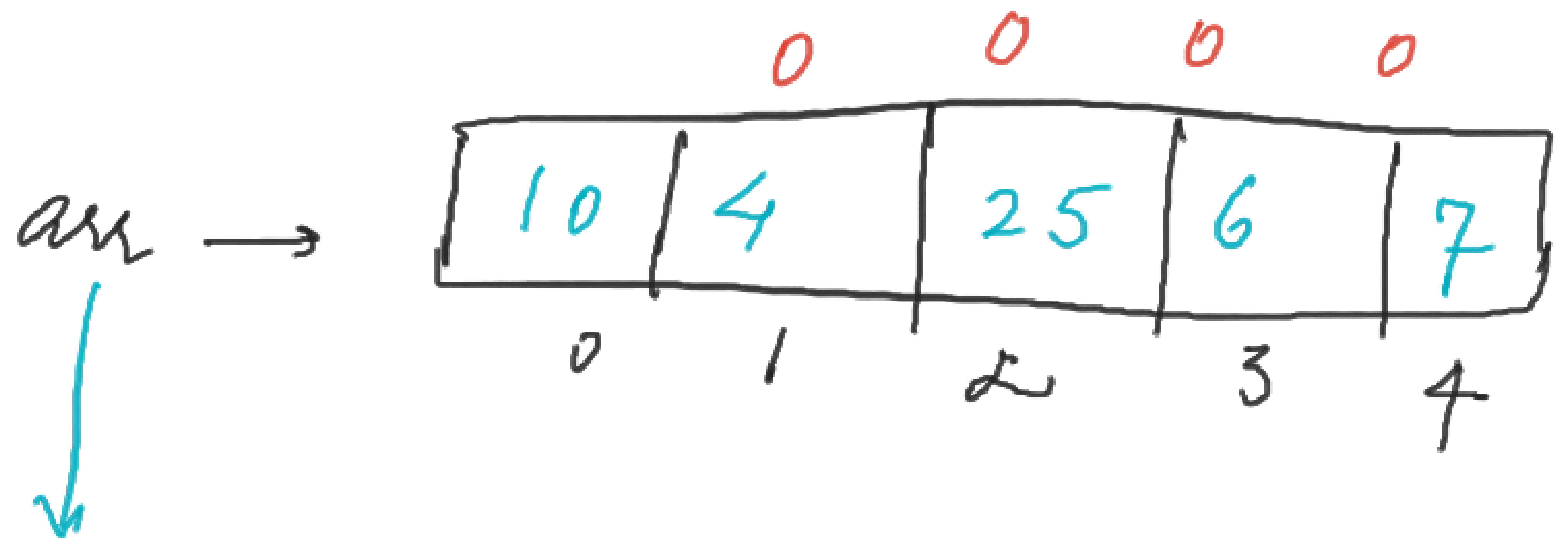


Cupboard

HeapM|m



Primitive →	Wrapper Class
byte →	Byte →
short →	Short
int →	Integer
long →	Long
float –	Float
double –	Double
char →	Character
boolean →	Boolean



Value Assignment in an Array →

arr[0] = 10;   arr[4] = 7;

arr Name[index No] = value;

St1: Array's size specify -

```
int[] arr = new int[size];
```

St2: value Assign / Store into Array.

$arr \left[\begin{matrix}index\\No.\end{matrix}\right] = 40; \xrightarrow{} value$

St3: Get Array's Index Value :

```
Syso(arr[index]); →
```

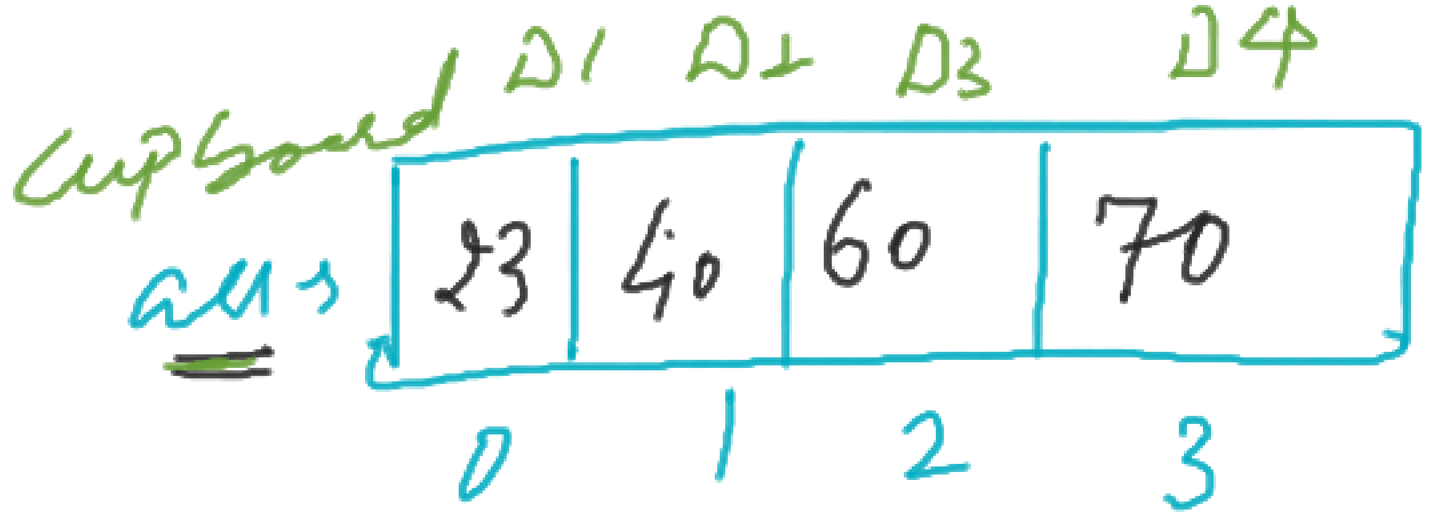
2nd

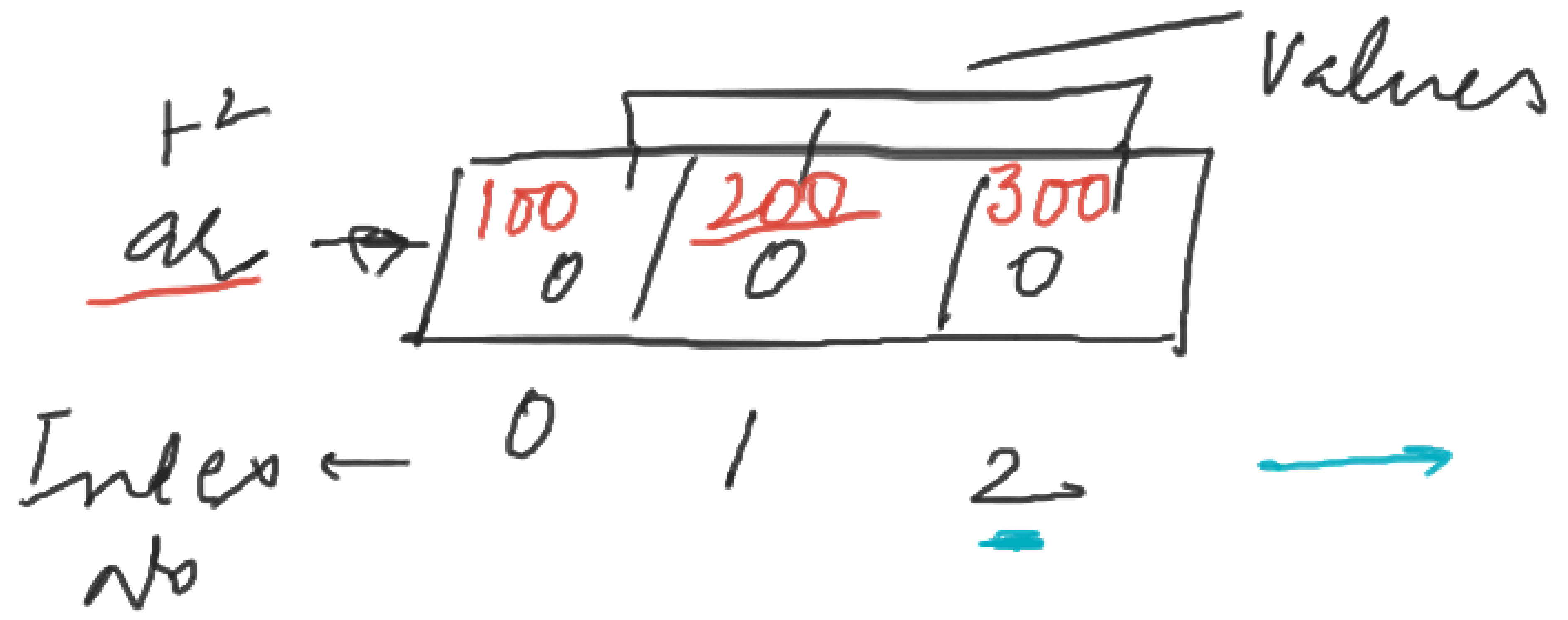
```
int[] arr = {23, 40,60,70};

Syso(arr[1]); → 40
```

Ans

$a = \underbrace{23,40,60,70}$





ar[0] = 100;
ar[1] = 200;
ar[2] = 300;

Loop

→ Total length → (arr.length) // 3

Last Index =
(arr.length - 1) ;

1.# Array → Store $\underbrace{2, 4, 6, 8, 10, 12 \ldots \ldots 20}$] . 10

2.# find Sum of Array Elmns :

→ {2, 4, 1, 2, 6} → Sum ⇒ 15

3.# find the Largest Elm of an Array .

$[41, 43, 60, \boxed{90}, 2]$

↳ Ans .