#DSA

Lecture No. (8)

★ Arrays & ArrayList ★.

* Array is something collection of data types.

* In Array you can store only one type of data type value
e.g. → int, (only int) , String. etc.

* SYNTAX of Array:-

// datatype [] Variable_name = new datatype[size];

// store 5 Roll nums:

e.g →

```
int[] rollno = new int[5]
```

OR

```
int[] Rollno.2 = {23, 12, 45, 32, 15};
```

↓ Internal Java syntax

Declaration of array. getting declared in stack Memory.

Initialisation. Actual in heap Memory.

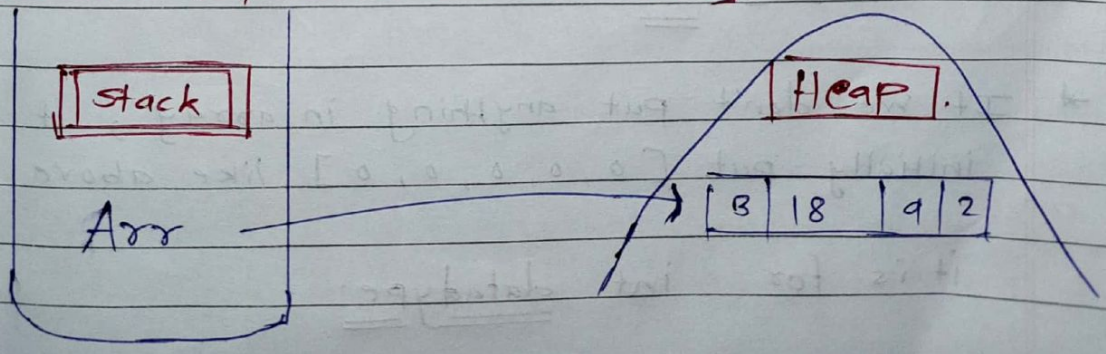
* Dynamic Memory Allocation :→

Compile time | Runtime.

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

- int[] → Datatype
- arr → referance variable.
- new int[5] → Creating objects in heap memory.
- New is used to creat an object. (creat size (5) object in heap. Mem).

* Diagramatic Repoesentation of Array :-



// 8 Memory allocations in Java (continous or not ?

⇒ It may be continous or not, it totally depends on JVM.

* Reason Reasons :-

① Objects are stored in heap Memory.

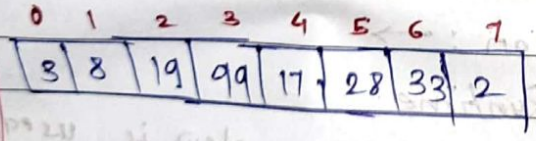
② According to JSL (Java language Specification) it is Mentioned that heap objects are not continous.

③ Dynamic Memory Allocations.

∴ Hence, May not be continous ⇒ Depends on JVM

* Index of Array :-

index starts from 0.



arr[0] = 3
arr[1] = 8
arr[2] = 19
arr[3] = 99
arr[4] = 17
arr[5] = 28
arr[6] = 33
arr[7] = 2

print(arr[0]) = 3 (it will print 3)

You can also change the values like,

arr[5] = 20.

* If we don't put anything in array, it will initially put [0, 0, 0, 0, 0] like above array.

it is for int datatype.

all other, classes, obj. n all stored in heap

& For string :- "null" its not any datatype object. or any.

→ Its a literal

→ It is a by default value of special variable. (referance)

* Take input using for loop: Syntax ↓

```
// input using for loop:
for (int i = 0 ; i < arr.length; i++ ) {
        arr[i] = in.NextInt();      ==> input.
    }
& For printing same input ;
  for ( int i = 0 ; i < arr.length; i++ ) {
        System.out.Print ( arr[i] )
  }.
```

* Input using for each loop (enhanced for loop).

```
Syntax:- for ( int num : arr ) {
            System.out.Print ( num + " " ) ;
         }.
```

+ Input using tostring Method (Best & Easiest one).

```
Syntax → Take input in in.nextInt ;
         System.out.Println (Arrays.tostring (arr)) ;
```

output ==> [1, 2, 3, 4, 5].

* For String datatype:- Same:-

```
String [] str = new string[4] ;
for { int i = 0 ; , i < str.length; i++) {
```

Same output.

8th sept 8:30

In Java → ① Strings are immutable.
② Arrays are mutable.

classmate Date Page

2D Array :-

Array of Arrays

Syntax : `int[][] arr = new int[3][];`

3 → no. of rows, [] → no. of columns

OR

```
int[][] arr2D = {
                {1, 2, 3},      0th index
                {4, 5, 6},      1st index
                {7, 8, 9}       2nd index
                };
```

3 x 3

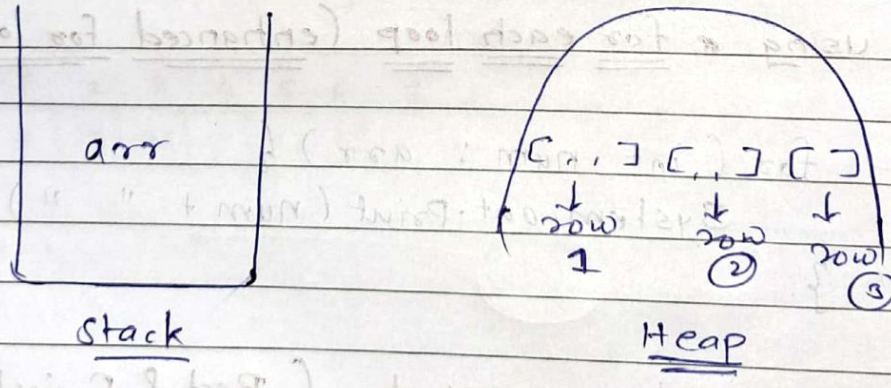
1 2 3

4 5 6

7 8 9

- input number of rows compulsory.
- number of columns are not mandatory.

Representation :-



- index value :- `arr[0] = {1, 2, 3}.` &
`arr[0][1] = 2.`

imp

```
arr[2][0] = 4
arr[2][1] = 5
arr[2][2] = 6.
arr[3] = {7, 8, 9}
```

* Arraylist :- (Java. Util. Package).

→ If you don't know about the size of Array or how many collections are there then "Arraylist" is helpful.

→ Syntax :→

```
ArrayList<Integer> list = new ArrayList<>(10);
```

↓

Java. Util. Package.

→ So many things you can add in it. (more & more).

* Internal Working of Arraylist:-

→ Basically, length is fixed. of Arraylist.
→ when it will full, it will create new Arraylist & add an old to it.
→ old Arraylist is deleted.