

Arrays / ArrayList

* SYNTAX

`datatype [] variable name = new datatype [size];`

or directly:-

`datatype [] var-name = { your array };`

1. datatype represents datatypes of the elements in the arrays
2. all datatypes in an array will be the same.
3. `int [] ros;` — declaration of array, ros is getting defined in the stack.
4. `ros = new int [5]` — actually here object is being created in the memory (heap) initialisation.

* Working of an Array

compile time runtime

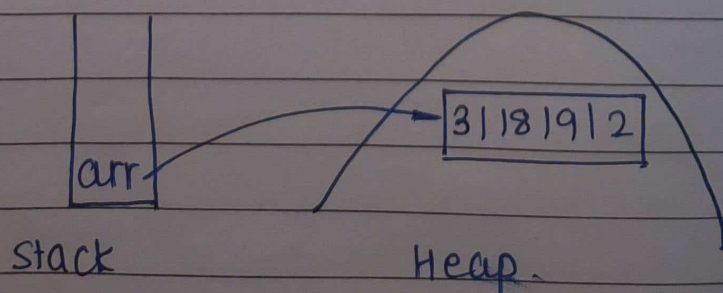
`int [] arr = new int [5]`

datatype reference variable creating the object in heap memory. initialisation at compile time.

declaration at compile time

⇒ Dynamic Memory Allocation:-

at runtime / execution time memory is allocated.



* Internal Representation of Array.

⇒ In C/C++ array is allocated continuous memory locations.

⇒ In Java it totally depends on JVM whether it is continuous or not.

⇒ array objects are in heap ⇒ heap objects are not continuous.

⇒ Dynamic Memory allocation :- Hence array objects in Java may not be continuous depends on JVM.

* Index of an array

3	8	9	10	53	33
0	1	2	3	4	5

* new keyword

⇒ new is used to create an object.

⇒ Initially if we don't provide values in the array, i.e. `int arr = new int[5] ⇒ {0,0,0,0,0}` all zeroes.

by default it has all values to 0.

* `String [] arr = new String [5];`

* Internal Working of Object arrays.

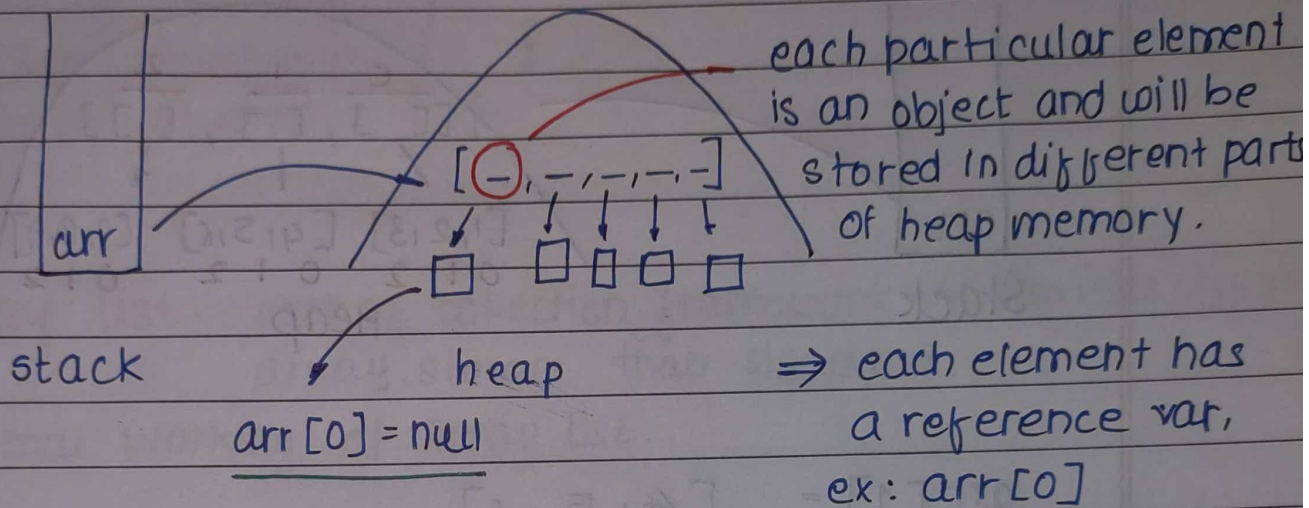
classmate

Date

Page

* primitives (int, char etc) are stored in stack memory

* All other objects are stored in heap memory.



⇒ by default reference variable points to null.

* for each loop → replacing `for (int i=0; i < arr.length; i++)`
└─ for (int num: arr) {
 // body
}
here num represents element of the array.

* To string :- `Arrays.toString (arr)`

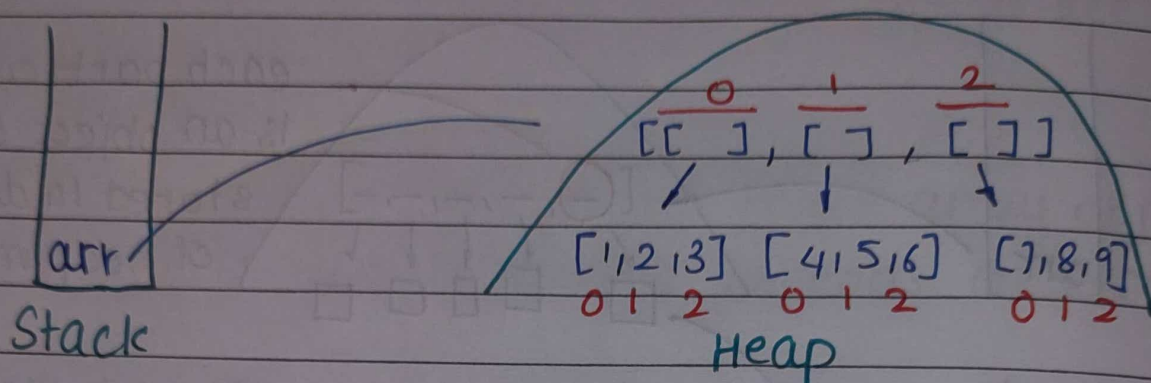
↓
converts to string

* arrays are mutable in JAVA.

* 2D Arrays in JAVA.

⇒ can be assumed to be like a matrix $[2 \times 2]$

* Internal Working of an Array.



`arr[1] = [4, 5, 6]`
 0 1 2

`arr[1][0] = 4`

⇒ column size is not necessary ⇒ size of the individual row does not matter.

⇒ i.e

[1, 2, 3]
[4, 5]
[6, 7, 8, 9, 10]
[5, 10, 11, 12]

this is also accepted.

* Initially we have

0		
1		
2		

classmate

Date _____

Page _____

first for loop \Rightarrow $\left\{ \begin{array}{l} \text{for (row=0 ; row < 3 , row++)} \{ \\ \quad // \text{ now we have every row.} \\ \quad \text{for (col=0 ; col < 2 ; col++)} \{ \\ \quad \quad \text{arr[r][c] = inp}^{\text{a}} \\ \quad \} \\ \} \end{array} \right.$

because 2D
array iterating
through each row
& all columns.

* **Array List.** part of collection framework & provides dynam array. slower than standard.

* **Internal Working of Array List.**

- size ~~is~~ is fixed internally.
- suppose the arraylist fills by some amount:-
 - \Rightarrow it will create a new array list of say double the size.
 - \Rightarrow old elements are copied in the new arraylist.
 - \Rightarrow old ones are deleted.