

- ! Array ! -

An Array is a Collection of Similar kinds of data element stored at Continuous memory location.

- In java, arrays are object, because of this we must Remember few points of array

- Since arrays are treated as object by java, So they also are Always created dynamically.
- To create an array we use the keyword new just like we use it to create an object.
- Since they are dynamically created they live in Heap Area.
- Because they are dynamically they don't have any name so they required a reference to be connected to them and this reference is called as Array Reference.

Syntax:-

Since arrays are object, they created in two Step.

1). create array reference

`<data type> [] <array_ref_Name>;`

2) create the actual array

`<array_ref_Name> = new <data type> [Size];`

↓
it can be Integer
or any Variable.

Syntax → Reference
`int [] arr;`

`arr = new int [5]`

↓ in short (Mamange)

★ `int [] arr = new int [5]`

or

★ `int arr[] = new int [5]`

But we make multiple array then

```
int arr[], brr;
```

v/s \rightarrow arr[] is array but brr is variable.

Bestway \rightarrow ^{int} [] arr, brr
to make

\rightarrow make array for both

Array

Access array :-

Syntax.

$\langle \text{array-ref-name} \rangle [\text{index No.}] = \text{Value};$

Example

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

```
arr [2] = 10;
```

```
arr [1] = 20;
```

```
arr [0] = 40;
```

```
arr [3] = 50;
```

```
arr [4] = 1;
```

```
System.out.println(arr[0]);
```

→ 40.

WAP to create integer array of size 'n' where 'n' should be taken from the User. Then ask the User to input 'n' element in the array and display all the element along with their Sum and average

```
import java.util.Scanner;
```

```
class Main
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        int n, i;
```

```
        Scanner kb = new Scanner(System.in);
```

```
        System.out.println("Enter size");
```

```
        n = kb.nextInt();
```

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

```
        for (i = 0; i <= n - 1; i++)
```

```
            System.out.println(arr[i]);
```

```
            System.out.println(i);
```

```
        }
```

```
    }
```

```
{  
    sop("Entn no.");  
    arr[i] = kb.nextInt();  
}
```

```
int sum = 0
```

```
for (i = 0 ; i <= n-1 ; i++)
```

```
{
```

```
    sop(arr[i]);
```

```
    sum = sum + arr[i];
```

```
}
```

```
sop("Average " + (float) sum/n);
```

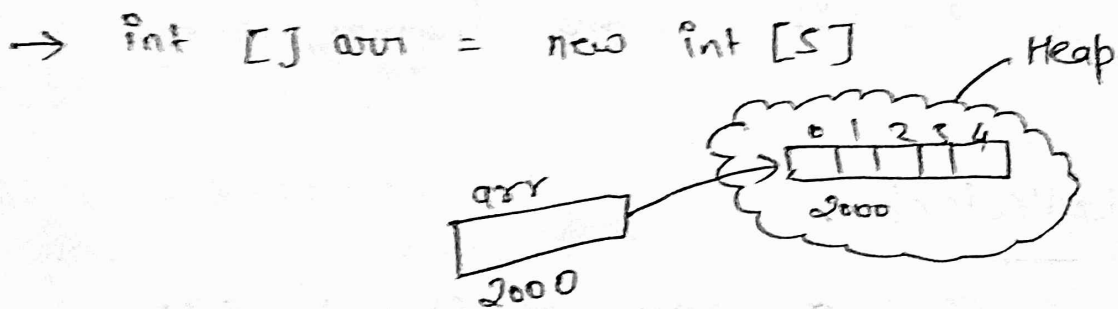
```
}
```

```
}
```

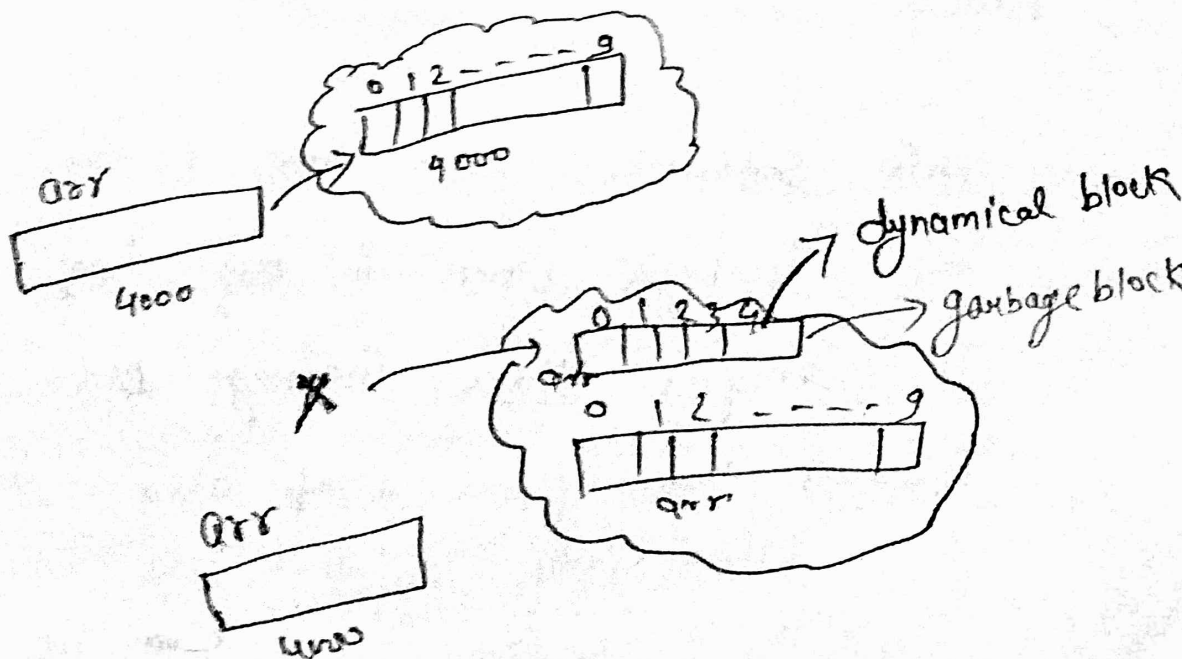
How Java Handles De-allocation of dynamic Block?

→ Before, we can understand how Java handles De-allocation of dynamically blocks, we must first understand 2 very important terminology:-

- ① Garbage Block
- ② Garbage Collector



→ ~~int~~ `arr = new int [10]`



In java , if we do not have any reference pointing to dynamic Block (object, array) then such dynamic block are better known as Garbage Block. In simple word , we can say that if a dynamic block does not has any reference pointing to it, then it become Garbage block.

Garbage Collector:-

In JVM a Special Software called Garbage collector to handle Garbage Block.

This Software run time to time and identify the Garbage Block in our heap area. Once it identifies these Garbage Block , then it Removes them from heap area and Send them to FREE POOL so that the Garbage Block can be utilized.

All this activity done by JVM.

1) `int [] arr = new int [12]`

`arr[0] = 31;`

`arr[1] = 28;`

`arr[2] = 31;`

`⋮`

`arr[11] = 31;`

another method! -

`int [] month = new int [] { 31, 28, 31, ... 31 };`

Simple way →

↓
No Size if use this way.

(or)

`int [] month = { 31, 28, 31, ... 31 }`

↓

New Keyword is optional