**Day-01**

**10-03-2025**

**=================**

**Arrays:**

**=====**

How variables internally work?

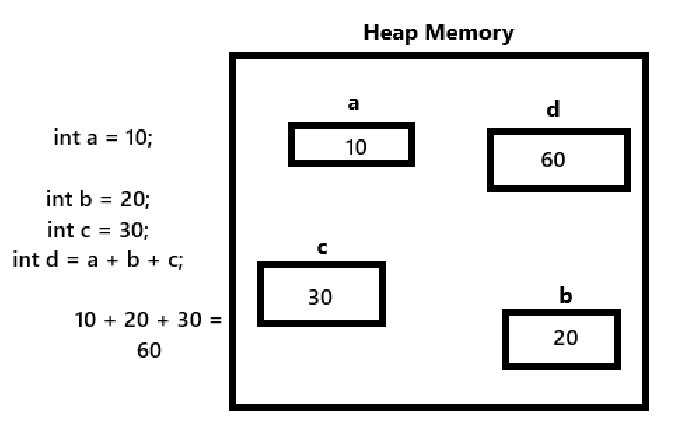
----------------------------------

->

int a= 10;

when we can initialize the variable,

the JVM can create the memory block for variable with the size based on the datatype then store the assigned value into that memory block.



Why Arrays?

---------------

-> When we have more number of variables in the java program it can increase the complexity or damage the processing time.

More variables can take more processing time.

-> To overcome this drawback we can use arrays.

-> When we want to store more than one value with same type into the single variable, we can use "arrays".

Array Definition:

-----------------

Array is one of the reference datatype.

It is a collection of similar (same datatype) data items.

Array Declaration:

---------------------

Syntax:

datatype array-name[] = new datatype[size];

or

datatype[] array-name = new datatype[size];

How Array can internally work?

-----------------------------------

After the declaration of an array, the JVM can create the memory in the heap with the size of (number-of-elements X size of datatype).

Ex: int[] a = new int[5];

in this case, the JVM can create 20-bytes of memory in the heap area.

and the total 20-bytes can divide into 5-blocks.

Each block with 4-bytes of size (int size).

And each block can be represented with an index value.

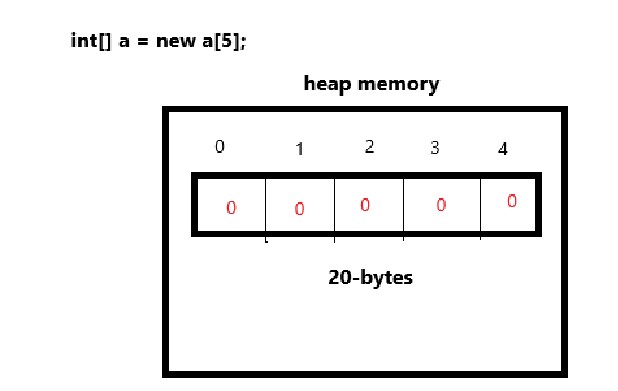
-> When we have not assigned any values for the array which we have declared then, the JVM can store default values according to the specified datatype.

Ex: if the array is integer type:

default value = 0

String type ==> null

boolean type ==> false etc.



-> To access individual array elements, we can use an index.

Syntax:

array-name[index];

public class ArrayDeclaration {

public static void main(String[] args) {

int a[] = new int[7];

char[] b = new char[5];

String[] c = new String[4];

System.out.print(a[0]);

System.out.print(a[1]);

System.out.print(a[2]);

System.out.print(a[3]);

System.out.print(a[4]);

System.out.print(a[5]);

System.out.print(a[6]);

System.out.println();

System.out.print(b[0]);

System.out.print(b[1]);

System.out.print(b[2]);

System.out.print(b[3]);

System.out.print(b[4]);

System.out.println();

System.out.println(c[0]);

}

}