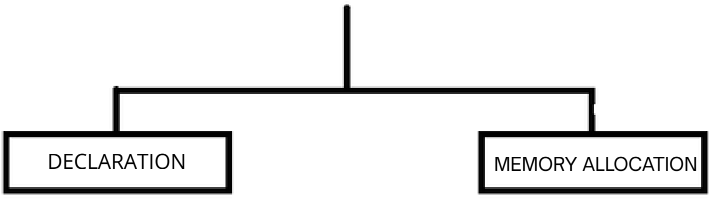
Shraddhesh-08

ARRAY

ARRAY

How to Use Arrays in Java

If we want to use arrays in java we have two steps.



1.Declaration of array

- WHEN WE ONLY DECLARE AN ARRAY, IT JUST CREATES A REFERENCE (ADDRESS) FOR IT, BUT NO MEMORY IS ALLOCATED YET.

Syntax

```
Type 1:
        datatype variablename [];

 Type 2:
        datatype [] variablename;
```

SHRIDDHESH PEDNEKAR

Null

null means “no value” or “no reference to any object.” It shows that the variable does not currently point to any object in memory.

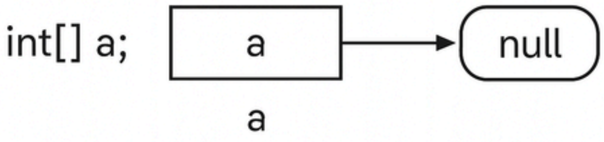
EXAMPLE

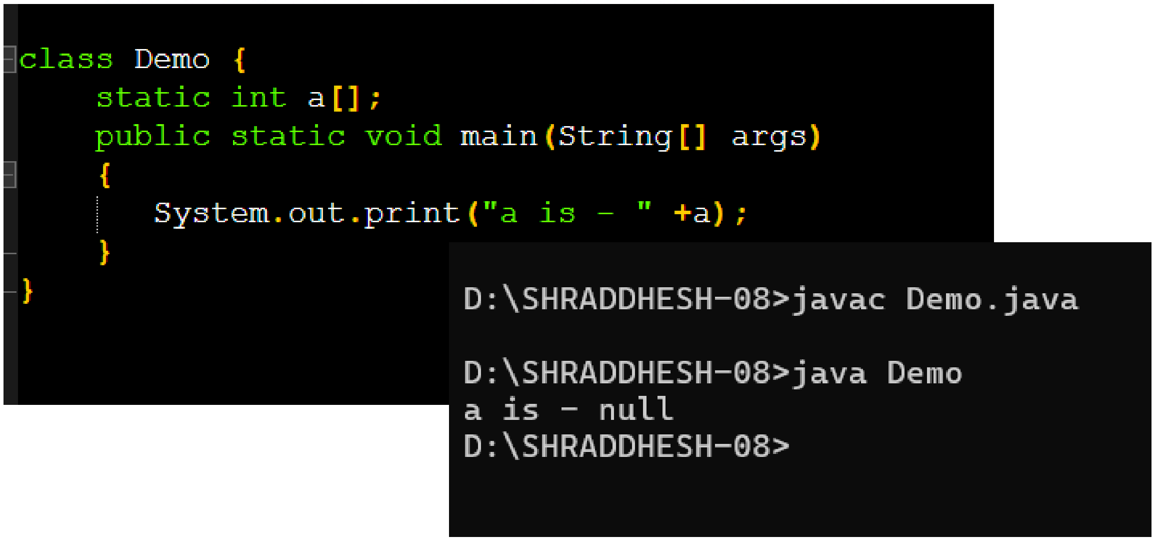
```
int[] a;  // declaration only
System.out.println(a);  // Error: variable a not initialized
```

SHRIDDHESH PEDNEKAR

Here:

- a is declared but not initialized.
- It doesn’t point to any memory location (no array created).
- So its value is null.





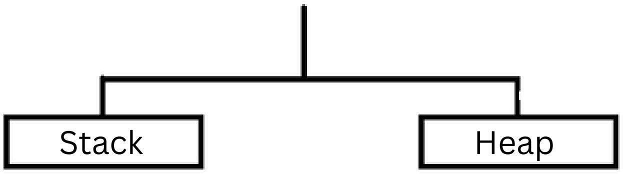
```
class Demo {
    static int a[];
    public static void main(String[] args)
    {
       System.out.print("a is - " +a);
    }
}
```

```
D:\SHRADDHESH-08>javac Demo.java

D:\SHRADDHESH-08>java Demo
a is - null
D:\SHRADDHESH-08>
```

2.Memory allocation of array :

Memory allocation means reserving space in RAM to store data (like numbers, characters, objects, etc.) during program execution.

memory allocation happens in two parts:



Stack

Heap

1. Stack Memory

- Stores local variables, method calls, and references (like array names).
- Memory is created automatically when a method is called.
- Memory is released automatically when the method ends

Example:

```
int[] a; // 'a' stored in stack (it is just a reference)
```

SHRIDDHESH PEDNEKAR

Here, only the reference a is created in the stack memory.
It doesn’t yet point to any actual array data (so value = null).

2. Heap Memory

- Used to store objects and arrays (actual data).
- When you use new keyword, memory is allocated in the heap.

Example:

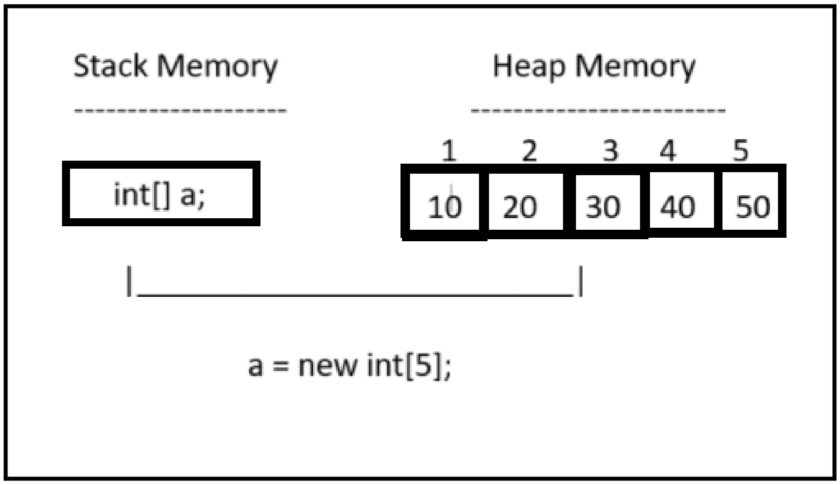
```
a = new int[5];
```

SHRIDDHESH PEDNEKAR

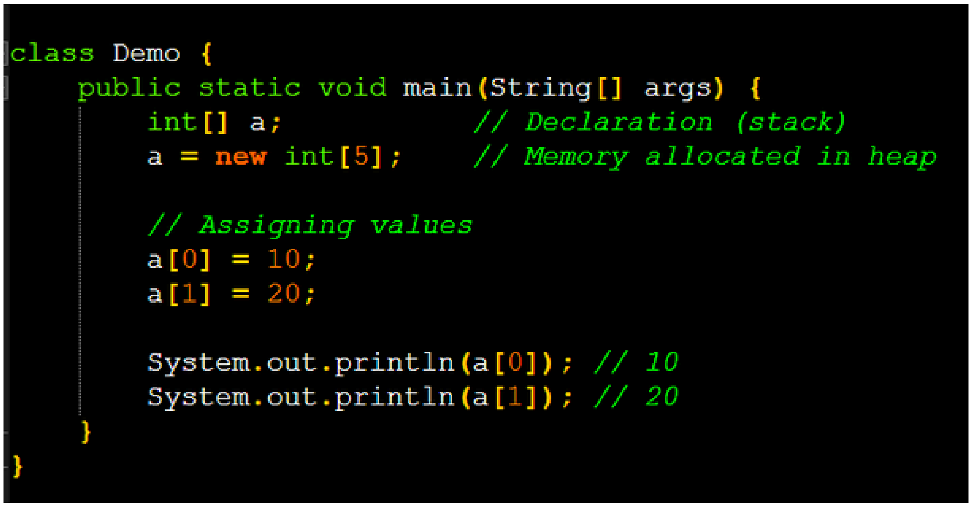
Now:

- A block of 5 integer spaces is created in heap memory.
- The reference a (in stack) now points to that block in heap.

Diagram: -

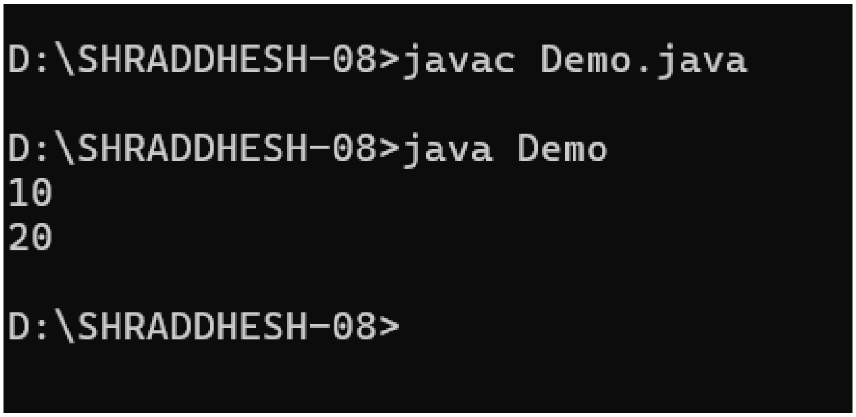


Example



```
class Demo {
    public static void main(String[] args) {
        int[] a;              // Declaration (stack)
        a = new int[5];       // Memory allocated in heap

        // Assigning values
        a[0] = 10;
        a[1] = 20;

        System.out.println(a[0]); // 10
        System.out.println(a[1]); // 20
    }
}
```

Output



```
D:\SHRADDHESH-08>javac Demo.java

D:\SHRADDHESH-08>java Demo
10
20

D:\SHRADDHESH-08>
```

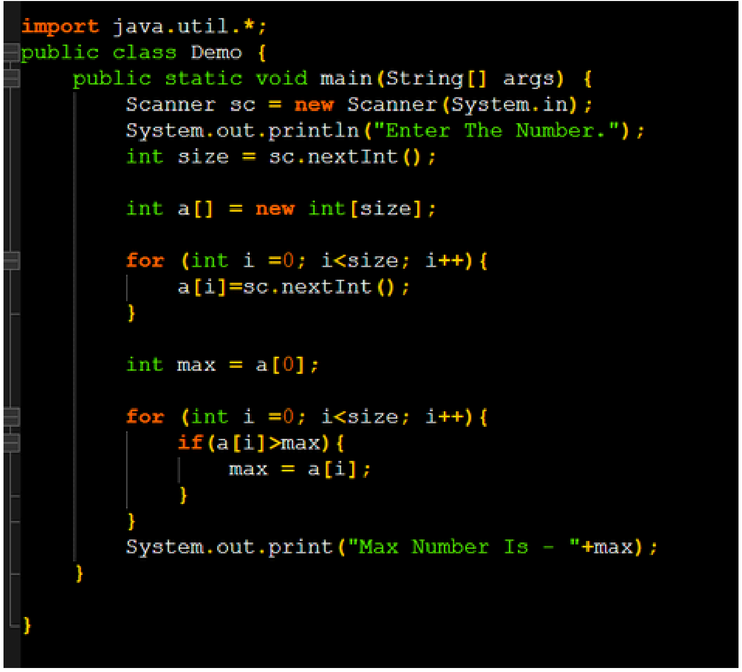
Q. What is the meaning of a[i]?

a[i] represents the element at index i in an array a. In memory terms, its address is calculated

Address of a[i] = Base Address + (i × Size of each element)

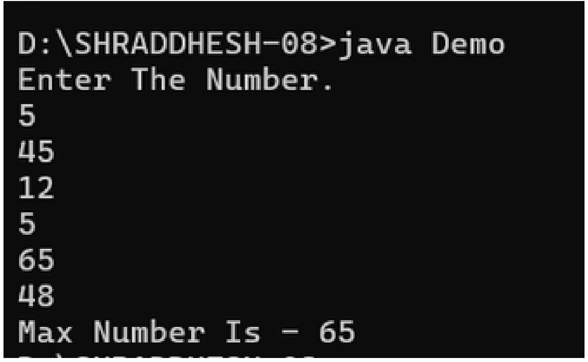
- **Base Address:** The memory location of the first element (a[0]) of the array.
- **Index (i):** The position of the element in the array (starting from 0).
- **Size:** The number of bytes required to store a single element (depends on data type, e.g., int = 4 bytes).

Exmaple -



```java
import java.util.*;
public class Demo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter The Number.");
        int size = sc.nextInt();

        int a[] = new int[size];

        for (int i =0; i<size; i++){
            a[i]=sc.nextInt();
        }

        int max = a[0];

        for (int i =0; i<size; i++){
            if(a[i]>max){
                max = a[i];
            }
        }
        System.out.print("Max Number Is - "+max);
    }

}
```

Output



```
D:\SHRADDHESH-08>java Demo
Enter The Number.
5
45
12
5
65
48
Max Number Is - 65
```