To take the input from the user Scanner class is required which is present in util package

To use them we need to create the object of the Scanner class and with the object we can call methods of different data types.

Scanner sc = new Scanner (System.in)

Above line enables user to type something on the console

int a = sc.nextInt();

above line take the input and give it to java program

1dimmensional array

Eg: Array\_1\_Dimensional\_Eg1

// go through the code

Eg: Array\_1\_Dimensional\_Eg2

// go through the code

Eg: Array\_1\_Dimensional\_Eg3

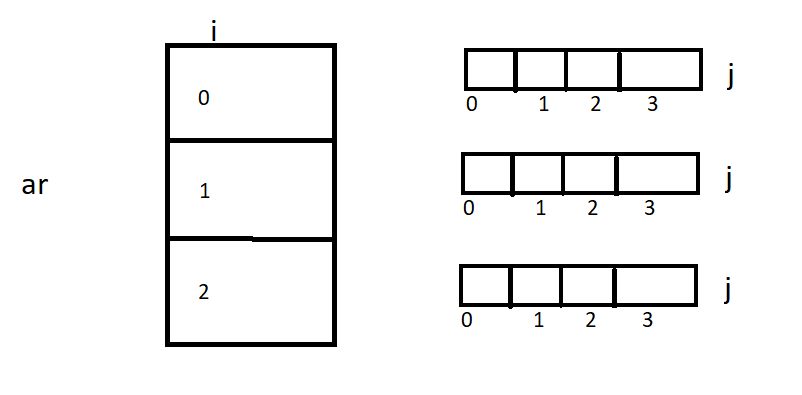
// go through the code

Eg: Array\_1\_Dimensional\_Eg4

// go through the code

2d regular array

Eg: Array\_2\_Dimensional\_Eg2



array.length property gives the size of the i which is 3 here.

ar[i].length gives the length of j

Eg: ar[0] length is 4

ar[1] length is 4

ar[2] length is 4

program working

the outer for loop value i = 0 and enters inner loop, the inner loop runs until a[i].length which is j length i.e is 4 and assigning is data is also done

a[i][j]

a[0][0] = 25;

a[0][1] = 24;

a[0][2] = 74;

a[0][3] = 78;

like this the inner and outer loop works for i = 1,2.

2dimensional jagged array

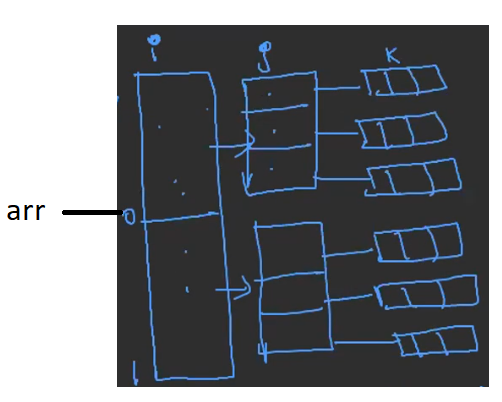
Eg: Array\_2\_D\_Jagged

In jagged array if there is irregular data leave that as empty.

There are different no of students in the class so it is left empty while array declaration, and later the irregular data is declared.

3dimensional array

Eg: Array\_3\_Dimensional\_Eg1



arr.length gives length of i which is 2

arr[i].length gives length of j which is 3

arr[i][j].length gives length of k which is 3

so totally a[i][j][k] = a[2][3][3]

3dimensional jagged array

Eg: Array\_3\_D\_jagged

3d jagged array

|  |  |  |
| --- | --- | --- |
| Collage | class | Students |
| 0 | 0 | 4 |
|  | 1 | 2 |
| 1 | 0 | 3 |
|  | 1 | 1 |
|  | 2 | 5 |
|  | 3 | 2 |
| 2 | 0 | 3 |
|  | 1 | 4 |
|  | 2 | 3 |

Int [][][] arr = new int[3][][]

arr[0] = new int[2][]

arr[1] = new int[4][]

arr[2] = new int[3][]

arr[0][0] = new int[4]

arr[0][1] = new int[2]

arr[1][0] = new int[3]

arr[1][1] = new int[1]

arr[1][2] = new int[5]

arr[1][3] = new int[2]

arr[2][0] = new int[3]

arr[2][1] = new int[4]

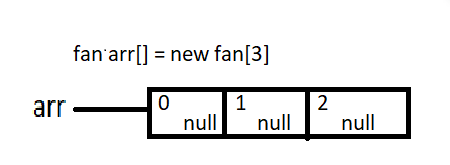
arr[2][2] = new int[3]

Array is not only of primitive type it can be of classes, collections but of similar type data, example is class means only class type of data, if primitive type int means only int type data etc.

Eg: Array\_Object\_Type

Memory map

For String and object the default value is null.



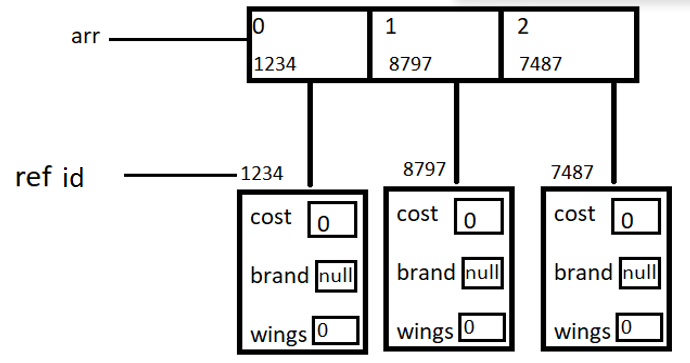
When object is created default values are assigned, here we are creating object of class type . the default value of Object type is “null”.

Creating object of fan and storing them in array locations

arr[0] = new fan()

arr[1] = new fan()

arr[2] = new fan()



When object is created for fan at array locations 0,1,2

All the variables are in fan are created and default values are assigned, and reference id is given to the 0,1,2 locations.

And later the default values are overridden when with assigned values

Disadvantages of array

1. It can store only homogenous type of data

Eg: int arr[] = new int[3]

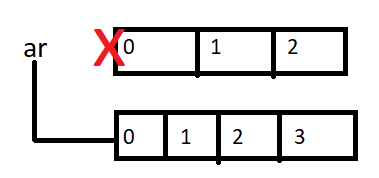
Here we declared int type of data we cannot assign char, float, double, long etc type of data.

suppose if type of array is class it should only be that class type.

1. Memory of array is fixed it cannot grow or shrink

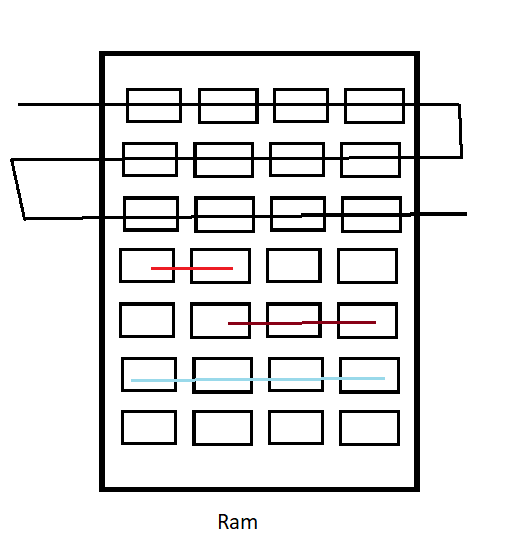
Eg: int arr[] = new int[3]

If we try to increase the array size to 4 it will not increase its size instead a new array is created.



1. Array demands contiguous memory location

int arr[] = new int[3]



Int size is 4 bytes and array size is 3 so 4\*3 = 12 bytes

12 bytes of continuous memory is needed to store the data in ram

In the above diagram the ram consists of some other data stored in bytes (different colored lines), the array needs only the contiguous memory location on 12 bytes. since the first 12 locations are free array is stored there (black line)

Array cannot be stored in between free spaces of storage if there is no continuous storage of its size.

Note: To overcome the disadvantages of array we have array list.

We can use methods in arraylist like sort (arraylist.sort()) , but we cannot do that in the array

Different syntax for declaring an array

Int[] a = new int[size declaration] -> recommended

Int []a = new int [size declaration]

Int a[] = new int [size declaration]

A class is said to be utility class if all the methods are static.