18/11/22

① Array :- Array is an object which Contains Elements of a Similar data type. Additionally the Element of a array are Stored in Contigous memory location.

(i) Why array ?

(1) Convinient / Traditional way to store data is to Create Variable. not good approach to store large volume of data.

(2) large volume of data Stored in Single Variable (Array)

② What is array ?

(i) array is indexed based data structure to store large volume of data using single Variable name.

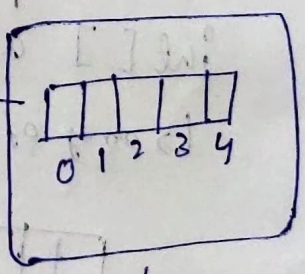
(ii) Array can Store homogenius type data.

(iii) Array in java treated as object.

↳ memory stored in heap area.

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

↳ Type of data



heap area

↳ a is array of integer.

↳ [] ⇒ array ⇒ Collection of data.

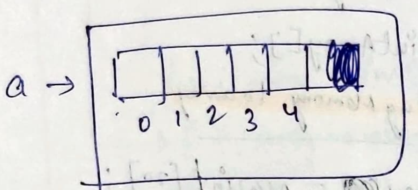
↳ ~~integer~~ type of data.

(iv) Homogenius Data:-

array is Collection of Similar data (or) Homogenous Data.

→ array of integer

int [ ] a = new int[5];

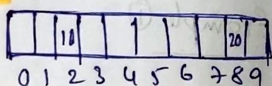


Heap area.

Eg① 10 - Students marks
↓ int

→ a is array of integer. a is reference Variable. it is refering array of integer.

int [ ] a = new int[10];



a[2] = 10;
a[8] = 20;

// to access data in array.

S.o.P (a[2]); → Print data in 2
↳ 10

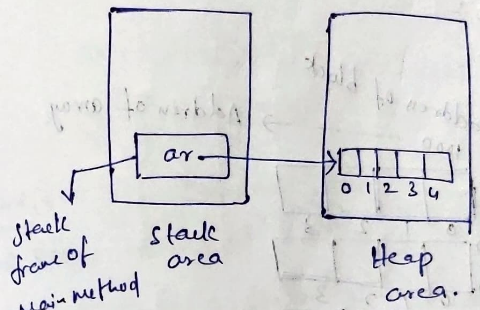
1D - Array

① Case-1 :- Create an array to store Marks of 5 Students

→. Students = 5.

int[ ] ar = new int[5];

→ Memory Map.



Case ② :- 2D - array.

Create an array to store Marks of 3 classes Each with 4 Students.

Classes	Students
0	4
1	4
2	4

→ data is regular

→ In array's we have Regular / Jagged array.

→ 2D - regular Array.

→ we have Create 2-D regular array for above data.

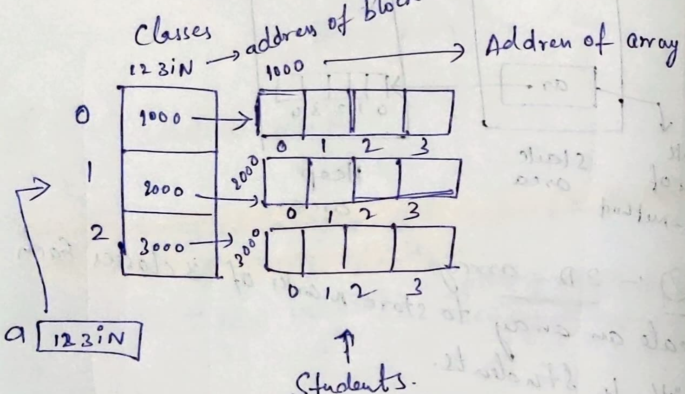
⇒ 2D Regular Array.

2 dimentional array of integer.

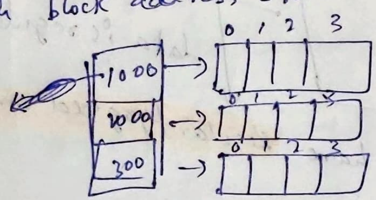
int [ ] [ ] ar = new int[3][4];

↓ Classes ↓ students.

⇒ Memory Map :-

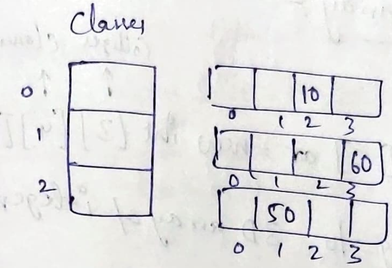


→ Each student block address stored in classes.



→ Address of class block stored in ar

ar [123iN]



ar[0][2] = 10;

ar[2][1] = 50;

ar[1][3] = 60;

Case ③ :- 3D Regular Array

⇒ store Marks of 3 Colleges 4 classes having. Each Class 3 students.

→ 3 dimensional.

Colleges	Class	Students
0	0 1 2 3	→ 3 → 3 → 3 → 3
1	0 1 2 3	→ 3 → 3 → 3 → 3
2	0 1 2 3	→ 3 → 3 → 3 → 3

→ All Colleges have same no:of Classes and Students

→ it is Called 3D → regular array.

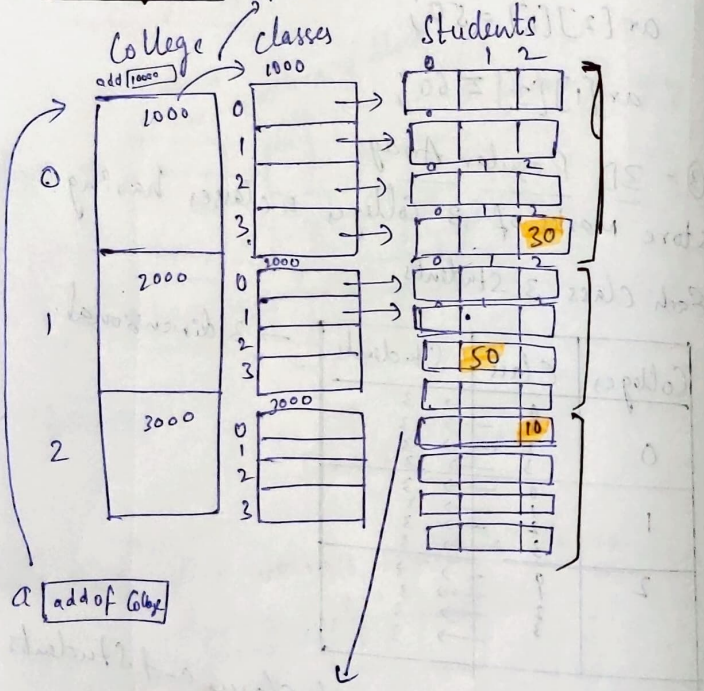
① 3D → Regular array :-

Colleges Classes Students

```
int [ ][ ][ ] ar = new int [3] [4] [3];
```

→ ar is reffering to 3D array of integer.

② Memory Map :-




```
ar [2] [0] [2] = 10;
ar [0] [3] [2] = 30;
ar [1] [2] [1] = 50;
```

Case ④ :- 2D Jagged array :-

→ 3 classes with different number of Students

→ irregular data so it is called 2D-Jagged array.

Class	Students
0	5
1	3
2	4

→ 2D-Jagged array

```
int [ ][ ] ar = new int [3][ ];
```

↳ we should not leave Empty.

↳ not sure leave it

array of 3rd class ↔ ar[0] = new int [5];

array of 1st class → ar[1] = new int [3];

array of 2nd class → ar[2] = new int [4];

Case ⑤ :- 3D Jagged array

College	classes	students
0	0, 1 } 2	4, 2
1	0, 1, 2, 3 } 4	3, 1, 5, 2
2	0, 1, 2 } 3	3, 4.

~~college~~ :

→ 3D array of integer.

int[][][] ar = new int [3][ ][ ]; → colleges

↓ Colleges

→ Classes.

Classes {
ar [0] = new int[2] [ ] ;
ar [1] = new int [4][ ] ;
ar [2] = new int [3] [ ] ;
}

→ Zeroth college and zeroth class students

Students {
ar[0][0] = new int[4];
ar[0][1] = new int[2];

ar[1][0] = new int[3];
ar[1][1] = new int [1];
ar[1][2] = new int[5];
ar[1][3] = new int[2];

ar[2][0] = new int [3]
ar[2][1] = new int [4];
ar[2][2] = new int [2];
}

Case ⑤ :- College and Class are same by different no. of Students in class.

int [ ][ ][ ] ar = new int [3][4][ ];

Coll	Class	Stu
0	0	4
	1	2
	2	3
	3	1
1		
3		

ar [0][0] = new int[4];
ar [0][1] = new int[2];
ar [0][1] = new int [3];
ar [0][3] = new int [1];

⇒ for accessing Elements in 3D array -

ar [0][1][0] → Indexes

→ 1D, 2D, 3D ⇒ Regular
2D, 3D ⇒ Jagged.
4D ⇒

→ Multi dimensional array.