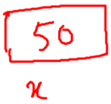
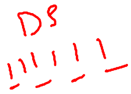
Data Structur

int x = 50;





Linear ,	Non-linear DS
✓ Arrays	Trees
Linked	Graphs
Stack	etc..
Queues	

✗ similar data

Array → list

↳ list of character ⇒ a, b, c, d ✓

list of integers ⇒ 1, 2, 3, 4

list of float ⇒ 1.1 , 2.1 , —

list of double ⇒ 2.11 , 3.11 ———

int v1 = 'a'

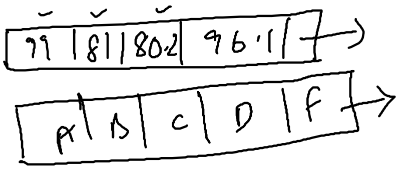
int v2 = 'b'

==

Student

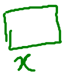
float

grades



How to Declare Array?

int x;



5

datatype array_name[size]

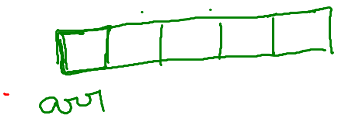
Khanan

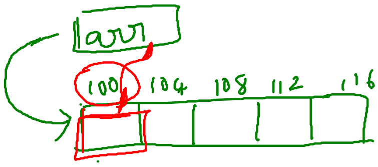
~~x~~

int arr[5];

int 4 bytes

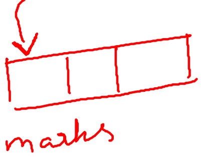
arr, a





continuous memory

int marks[3];



Intitailizing Array

int x = 10;

int x;
x = 9;

int arr[5];

arr[0] = 1;

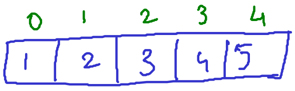
arr[1] = 2

arr[2] = 3

X

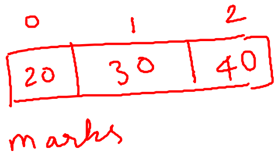
arr[3] = 4

arr[4] = 5



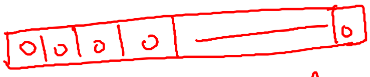
DI

int marks[3] = {20, 30, 40};

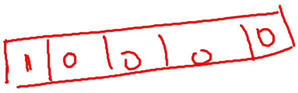


=> int nums[1000] = { . . . . . — } X

int nums[1000] = {0}

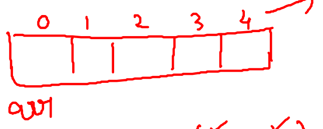


int nums[1000] = {1};



Accessing Array Elements

int arr[5];



Indexing

1 X

0 ✓

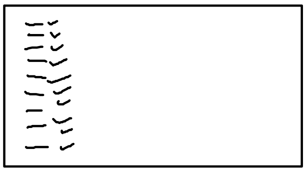
5 (0 ✓, 4 ✓)

(0 — (n-1))

int arr[5] = {1, 2, 3, 4, 5}

arr[3]; // 4

arr[1]; // 2



Array is powerful concept, jo store, organize, access.

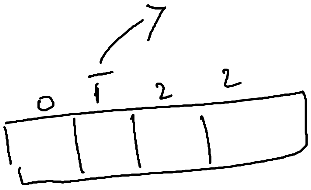
Why use Array?

100

string nit1 - "Jalebi"
ni : "k"

} => Arr
1 var
-

= 100 —
var



Printing of Arrays

```
int arr[5]={10,20,30,40,50};
for(int i=0;i<5;i++){
    cout<<arr[i]<<endl;
}
```

5

i=0 , 0<5 ✓
arr[0]

//

i=1 , 1<5 ✓
arr[1]

i=2 2<5 ✓
arr[2]

i=3 , 3<5 ✓
arr[3]

i=4 , 4<5 ✓
arr[4]

i=5 5<5 ✗

Output
10
20
30
40
50

int arr[5] = {10, 20} 000

int arr[] = {6, 3} ✓ int arr[2]

int arr[10] = {1} 0 1, —— 0 0 0 0}

int arr[]; // error ✓

Homework: Complete Array initialize with 2

int arr[100] = {2, 2, 2, 2} ✗

≡ 2 2 2 2

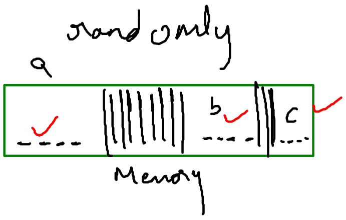
comment

arr[100] = {2} 0 1 0 0 0 0 0 0

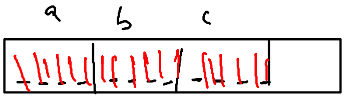
Memory Representatiin of Arrays

contiguous { } linked { }

int a = 5; // 4 bytes
int b = 10; // 4bytes
int c = 15 // 4bytes



int arr[3] = {5, 10, 15}



Types of Array

1) One Dimensional Array (1D)
linear — ||||

int arr[5] = {1,2}

2) Two Dimensional array (2D)
An arr -> matrices

=

1	2	3
4	5	6
7	8	9

int arr[rows][col] = { }

3) Multidimension (3D) ||||||||

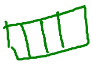
int arr[2][3][4] // 3D Array

4) Dynamic Array -> runtime

5) Array of objects (OOP)

6) Character Array -> characters (string)

Find Length of the array

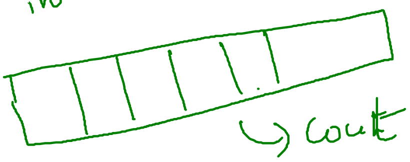


size

int => 4bytes.

int arr[10]

{10 x 4 => 40}



sizeof(a) -> 4
b -> 4
d -> 8
c => 1by

$\frac{40}{4}$ => 10

int arr[10] = {9}

9	0	0	0	0	0	0	0	0

=> 10

Good Practice and Bad Practice

1) int arr[] = {1, 2, 3, 4} ✓ good practice

int arr[4].

2) int arr[10] = {3, 4, 5} ✓ 0, 0 0 0 ✓

3) int arr[1000] ; ✗ arr[0] => garbage

unexpected

5, B

a	b	c	d	\0

0

(\0) -> null charact