

CS & IT ENGINEERING

Data Structures

Introduction to Data Structures Lec- 01



By- Pankaj Sharma sir

A stylized illustration of a laptop with an orange base and a blue frame. The screen is white and contains the text 'TOPICS TO BE COVERED' in a dark blue, hand-drawn font.

TOPICS TO BE
COVERED

Three dashed orange arrows originate from the right side of the laptop screen. One arrow points horizontally to the top of the 'Introduction' box. Another arrow points horizontally to the middle of the box. A third arrow points horizontally to the bottom of the box.

Introduction

1 to 1000

Data Structure

$$(625)/100 \Rightarrow (6) \quad 25/10 \Rightarrow 2$$

25 el-
1 \rightarrow 10
2 \rightarrow 10
3

Row wise arrangement
with a grouping of 10

1	2	3	...	10
11	12	13	...	20
...
91	92	93	...	100

7th Page
3rd Row
5th col

1	11	21	...	91
2	12	22	...	92
3	13	23	...	93
...
10	20	30	...	100

Column wise arrangement
with a grouping of 10

7th Page
3rd col
5th row

Row wise arrangement
with a grouping of 20

1	2	3	...	19	20
21	22	23	...	39	40
...
181	182	183	200

No Pattern, Randomly

3	7
---	---

Ex 2

Contact List

→ ~~Sorted~~ Unsorted

~~Search~~ → 9

Ex3:

Dictionary

Sorted

Search

Parrot

Lion

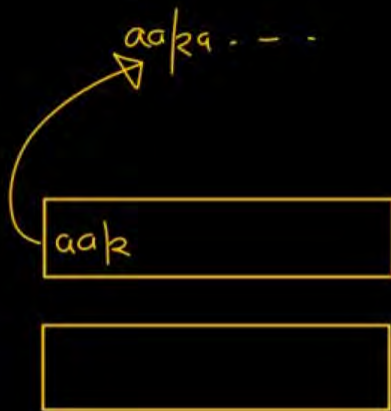
→ Insert

→ Unsorted

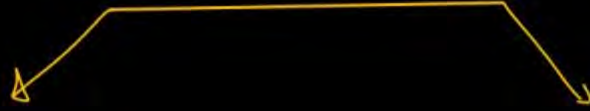
Ex4

e-mail

gmail



Data structure



1) Linear data structure

2) Non Linear data structure

A data structure in which
an element can have
almost 2 neighbours.

More than 2 neighbours ✓
can be possible.

Linear data structure

1) Arrays → Address calculation

2) Linked List → Code

3) Stack }

4) Queue }

Hashing

Non Linear data structure

1) Tree : Binary tree
Binary Search tree
Heap
AVL tree } 60%

2) Graph

C Programming ✓

→ Arrays, pointers, structure

→ 0 knowledge → [C recording] 2x →

Array

int m1, m2, m3;

≡

avg = (m1 + m2 + m3) / 3

≡

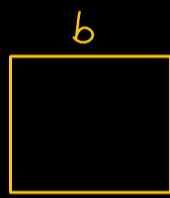
□

int m1, m2, m3, m4, m5, ...
... m50;

int a, b, c;



2016



3096



4196

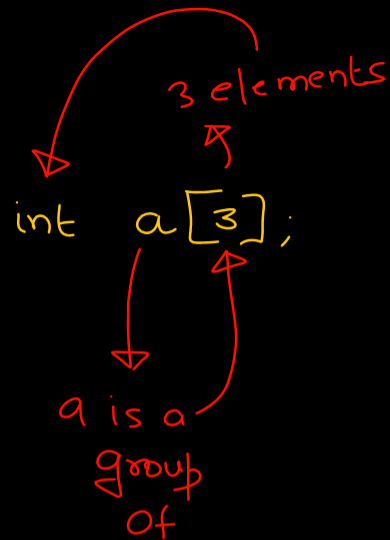


int a[3];

3 elements

a is a group of

of



a



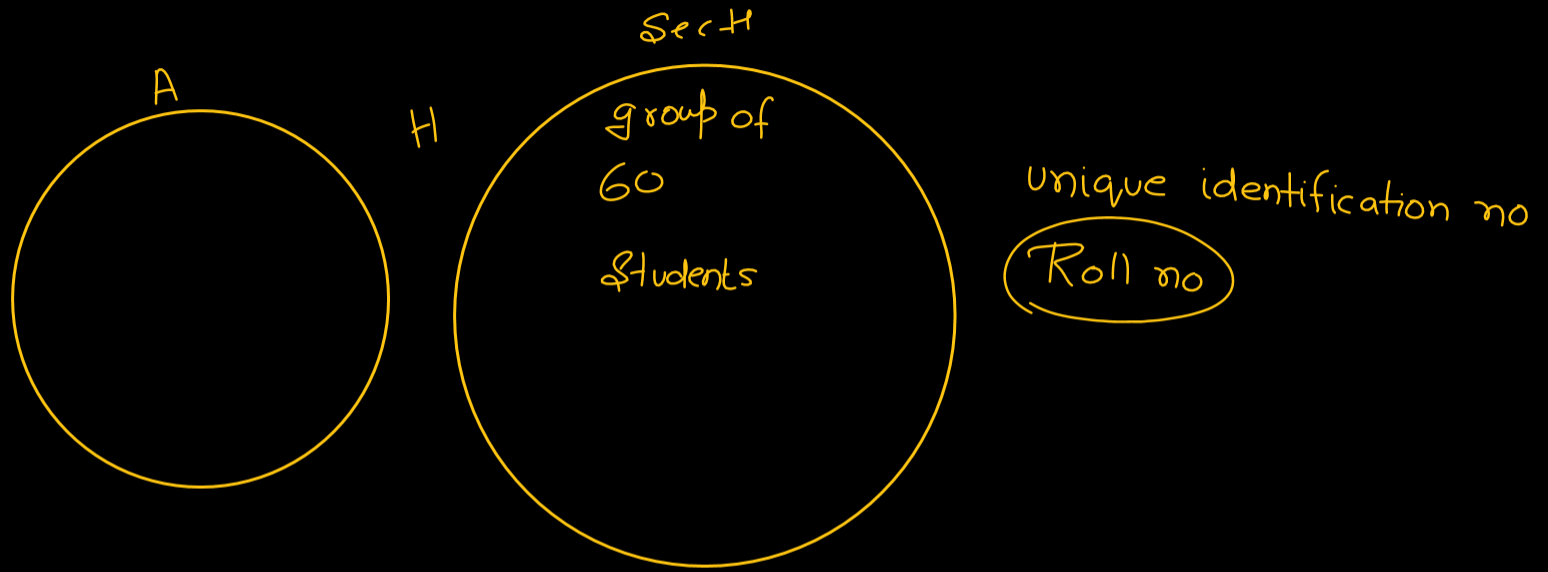
group of elements of same type.

```
int a[10];
```

```
char c[5];
```

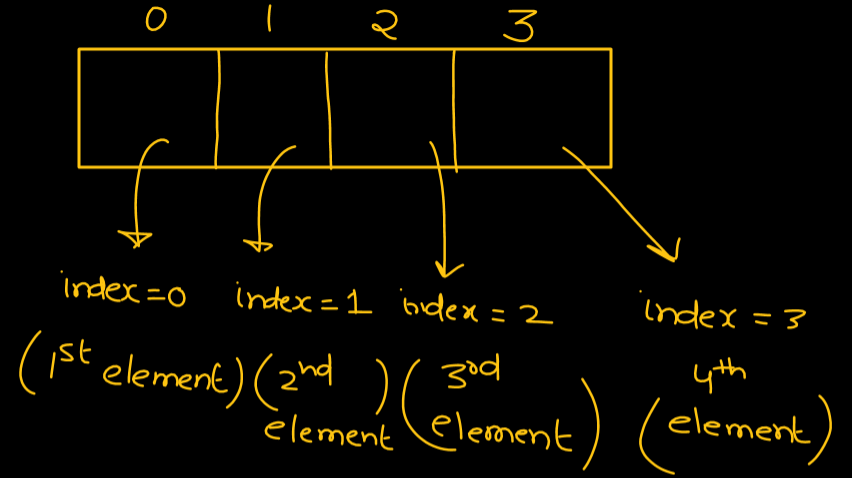
```
float p[20];
```

Collection of homogenous types of data elements



int a[4];

a

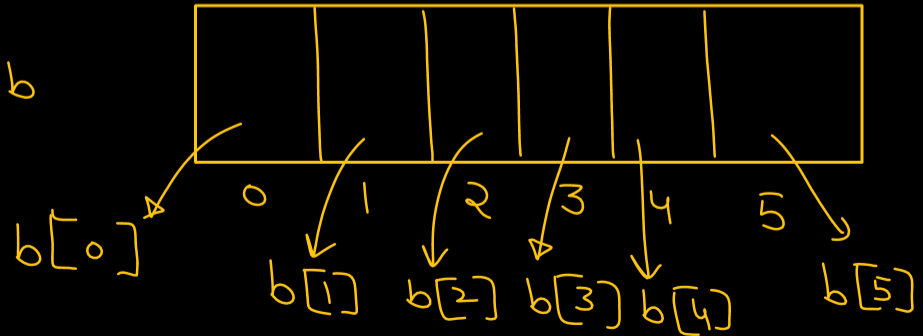


int b[6];

Diagram illustrating the range of indices for array `b`. Arrows point from the expression `b[6]` to the values 0 and 6-1, indicating the valid index range is 0 to 5.

0 6-1
0 to 5

b



```
int a, b, c;
```

$a = 10;$ ✓

$b = 30;$ ✓

$c = 50;$ ✓

a

0	1	2
10	20	100

$a[0]$ $a[1]$ $a[2]$

```
int a[3];
```

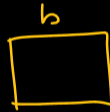
$a[0] = 10;$

$a[1] = 20;$

$a[2] = 100$

```
void main(){
```

```
    int a,b,c;
```



```
    int a[3];
```

