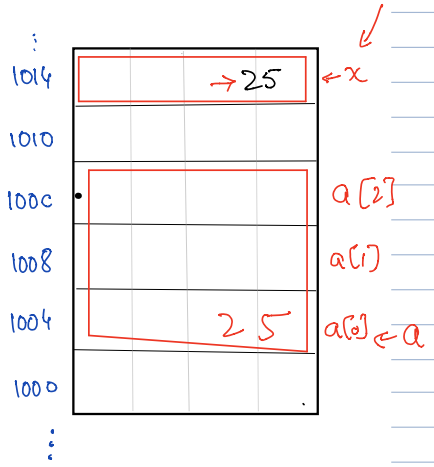
COLLECTING RELATED DATA

```
int x;
x = 25;
```

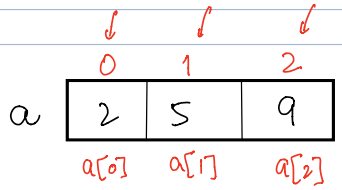


Arrays — same data type for all elements

- Homogeneous
- Contiguous

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

(int: data type; a: name; 3: size / # of elements)




```
a[0] = 25;
for (int i = 2; i >= 0; i--) {
    cout << a[i] << endl;
}
```

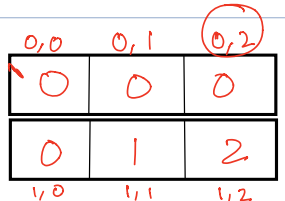
"i--"

→ i ⇒ error

```
a[0][2] = 27;
```

2-D Arrays:

```
int a[2][3];
for ( int i = 0; i < 2; i++) {
```

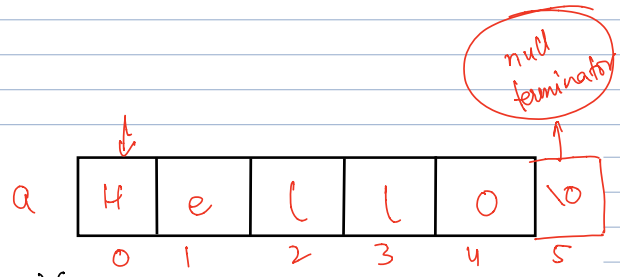


i 0
j 01

```
for ( int j=0;  j< 3 ;  j++) {   → columns
    a[i][j] =  i * j ;           //cout << a[i][j];
}
}
```

Character Arrays: string

```
char a[] = "Hello";
```




```
for (int i= 0;  i< 6 ; i++) {
    cout << a[i] << endl;
}
cout << a ;
```

Hello

H
e
l
l
o
.

→ Homogeneous

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string a = "Hello";
    a[2]
```

char a[];

Structs:

"A student has a name and a roll number."

struct / type — string — int

Objective: create a structure to store student information.

#include

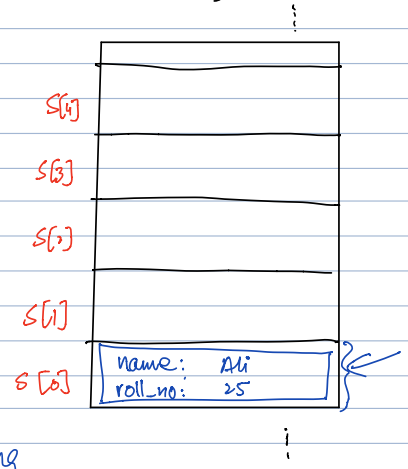
s1	
name:	"Ali"
roll_no:	25

```
struct student {
    string name;
    int roll_no;
};
int main() {
    student s1;
    s1.name = "Ali";
    s1.roll_no = 25;
    cout << s1.name;
}
```

s1 → "enter the box"

```
int x[5];
```

type — name — size / # of elements.




```
student s[5];
```

type — name — # elements.

```
s[0].name = "Ali";
```

assign string.

s[0] → student; name → enter

```
s[0].roll_no = 25;
```