

C-101 \Rightarrow Structures in C - Part 1

Introduction to Structures

* Why we need variable? \rightarrow To store value.

* When we want to store roll no. of 10 students then we use arrays under a single variable name which contains homogeneous values or data types.

* But when student have different informations like Roll no. (int), name (string), marks (float) for 2 students, then

Student 1

Student 2

Variables {
int rollno1;
char nameS1[];
float marks1;

Variables {
int rollno2;
char nameS2[];
float marks2;

* For 2 students if we take variables
when we start for 60 students we
cant manage more variables.

* But if we go for array; array can store
only same type of values of every students

```
int rollno[60] = { ... };  
char name[60];  
float marks[60];
```

→ Here also we take different arrays
for each type and this is also not
convenient to use.

Is it possible to declare or to take
different data types under a single type?

→ * Yes it is possible using "structure".

Note:-

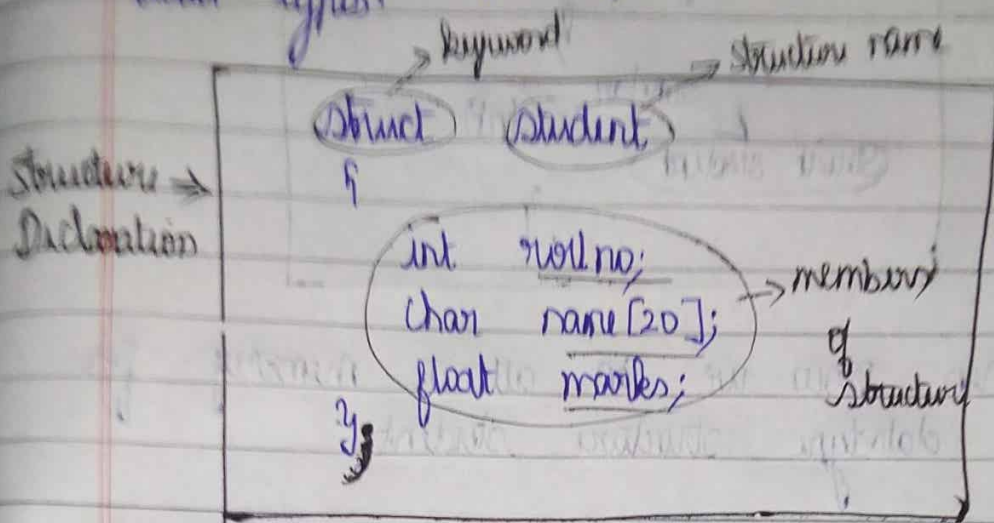
* int, float, double are fundamental data type
or primitive

* array is ~~not~~ ^{derived} ~~defined~~ data type

* Structure is user defined data type.

Structure :-

* Structure is user defined datatype in which we can store different data types under single name with the help of fundamental or primitive data types.



* Using this we can store a student information for many number of students.

Note:-

❌ We cannot initialize a individual structure member.

eg:-

```
struct student {  
    int rollno = 1;   
    char
```

* Because; this is just a template or it will just show compiler how structure is

laid out and memory is not allocated for this.

So, struct student is just a data type like int and it is not variable.

```
g: main()
{
    struct student int a = 1;
}
```

Similar

* Thus now we can allocate memory for our datatype structure student.

```
main()
{
    struct student
    {
        int a;
        int rollno;
        char name[20];
        float marks;
    }
    struct student s1;
}
```

Object of Structure

4 bytes
2 bytes
4 bytes

Total bytes of memory for s1 is allocated.

* For s1 how many bytes of memory is allocated?

→ It is allocated depending on the primitive datatypes defined in structure.

* 's1' is object of a structure.