Structures

As we saw Arrays contains elements of same data types. But if we need something which contains elements of different data types. So structures basically helps us in this.

It is a collection of different data types which referred under one name

Like information about a student composed of many different datatypes. like Name, age, Gender, Roll No, class

Structure Definition or structure defining and Variables.

We use keyword Struct followed by its name and body enclosed in curly braces Body of the structure contains the definition of its members and each member must have name.

→ declaration ends up by ; and after that structure variables are declared.

```
Struct <name> {
    member1
    member2
    :
    membern;
};  Structure Variable name
```

Eg:

```
Struct Student
{   Char Name[20];
    int age;
    int roll;
    Char class[10];
};  Stud1, Stud2
```

Accessing Structure Elements

There are two ways of accessing structure elements

(i) . dot operator or structure member operator :-

It accesses a structure member via structure variable

eg

```
Stud1.age;
```

(ii) Structure pointer operator(->) or Arrow operator :

Suppose a structure member declared as

```
int *gender      // 1 for male and 2 for female
```

and structure variable *Stud3

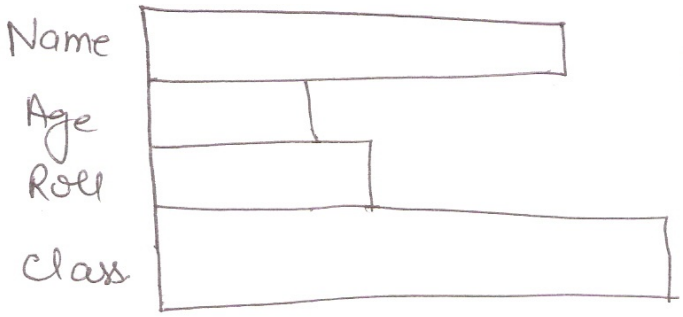
So

*Stud3 a pointer which has been declared to point to struct student and the address of structure *gender has been assigned to Stud3. So print it we will use

```
Stud3 -> gender
```

Memory Allocation of Structure

As soon as we declared structure name and the body elements with its variable space allocated to the memory



Memory Allocation

```
/* Simple and Complete program for Structure */
#include<Stdio.h>
#include<Conio.h>
 Void main()
{
   Struct Student
      { Char name[20];
        int age;
        int roll, Sub1, Sub2, Sub3, Sub4;
        int total;
      };
     Student Stud; // Variable declaration.
     Clrscr();
     printf("Enter the record of the Student :");
     printf("Name : ");
     Scanf("%s\n", Stud.name);
      printf("Age : ");
      Scanf("%d\n", Stud.age);
       printf("Roll NO. ");
       Scanf("%d \n", Stud.roll);
       printf("Enter marks of Subject 1,2,3,4 : ");
     Scanf("%d %d %d %d \n", Stud.Sub1, Stud.Sub2, Stud.Sub3,
                Stud.Sub4);
     printf("Grade of Student: %s is !", Stud.name);
       If Stud.total = Stud.Sub1 + Stud.Sub2 + Stud.Sub3 + Stud.Sub4
          if( Stud.total < 240)
              Printf("The goode D");
              else
                if( stud.total < 320)
                  Printf("\ B");
                  else
                  printf("A");
```

Arrays of Structure :-

An array of Structure can be declared as an ordinary array And in same manner we will access the member of Structure as we do by ordinary variable.

```
/* program to print 10 roll No. of */
  void main ()
  { Struct student
          { int roll;
          }; stud[10];
    printf("Enter 10 roll No ");
    for( int i=0; i<10; i++)
     { scanf("%d", stud[i].roll);
     }
    printf("The ten Roll No. are "); for(i=0; i<10; i++)
    printf("%d \n", stud[i].roll);
    getch();
  }
```

Output

```
Enter 10 ROLL NO.
 2
 8
19
18
 7
14
 5
 6
29
10
The ten ROLL NO. are
 2
 8
19
18
 7
14
 5
 6
29
10
```

Explanation

→ In this we declared Structure Variable Stud[10] of Student type

→ We access the values same as we access with ordinary Variables.