Unit-6 Structure and Union

6.1 Concept of Structure

A structure is a collection of different datatypes grouped together under a single name using the struct keyword. It is also known as a user-defined datatype that enables the programmer to store different datatype records in the structure. Furthermore, the collection of data elements inside the structure is termed as the member.

Syntax

```
struct structure_name
{
    datatype variable_name1;
    datatype variable_name2;
    datatype variable_nameN;
};
```

eg

```
struct student
{
    char name[20];
    int age;
    float marks;
};
```

Here, student is the name of the structure. name, age, marks are members of the structure. Each member can have a different datatype.

Structure Program example in C

```
#include<stdio.h>
#include<string.h>
  struct student
  {
     int id;
     Char name[30];
     char gender[10];
      int age;
   };
     int main( )
     {
       struct student st;
        st.id = 1;
        strcpy (st.name, "rajesh");
        strcpy (st.gender, "male");
        st. age = 28;
        printf (" id is %.d\n", st.id);
        printf ("name is %.s\n", st.name);
        printf ("gender is %.s\n", st.gender);
        printf ("age is %.d\n", st. age);
       return 0;
     }
```

output:

```
id is 1
name is rajesh
gender is male
age is 28
```

Write a C program to demonstrate structure and take user input.

```
#include <stdio.h>
struct student
{
    int s_id;
    char s_name[20];
    char s_address[20];
    int s_marks;
};
int main()
{
    struct student st;
    printf("Enter student id:\n");
    scanf("%d", &st.s_id);
    printf("Enter student name:\n");
    scanf("%s", st.s_name);
    printf("Enter student address:\n");
    scanf("%s", st.s_address);
    printf("Enter student marks:\n");
    scanf("%d", &st.s_marks);

    printf("\n Displaying Student info:\n");
    printf("%d is id\n", st.s_id);
    printf("%s is name\n", st.s_name);
    printf("%s is address\n", st.s_address);
    printf("%d is marks\n", st.s_marks);
    return 0;
}
```

output:

```
Enter student id:
1
Enter student name:
rajesh
Enter student address:
sukhad
Enter student marks;
80
Displaying student information:
1 is id
rajesh is name
sukhad is address
80 is marks
```

6.2 Initializing and Accessing members

Initialization

```
struct student s1 = { 1, "Ram", 85.5};
```

eg

```
#include <stdio.h>
struct student
{
    int id;
    char name[20];
    float marks;
};
int main()
{
    struct student s1 = { 1, "Ram", 85.5};
    printf("ID is %d\n", s1.id);
    printf("name is %s\n", s1.name);
    printf("marks is %.1f\n", s1.marks);
}return 0;
```

Example 2

initialize using dot (.) operator, assiging values after declaration.

```
#include <stdio.h>
#include <string.h>
   struct student
   {
     int id;
     char name[30];
     float marks;
   };
     int main()
     {
       struct student s1;
       s1.id = 1;
       strcpy(s1.name, "Ram");
       s1.marks = 85.5;
       printf("Id is %d \n", s1.id);
       printf("name is %s \n", s1.name);
       printf("marks is %.1f \n", s1.marks);
       return 0;
     }
```

output:

```
Id is 1
name is Ram
marks is 85.5
```

Accessing members of the structure is done by using (.) operator.

eg

```
printf("%d", s1.id);
printf("%s", s1.name);
```

6.3 Array of structure

Array of structure is collection of multiple records of same structure.
It is used to store multiple records of the same type and accessed using index like

s[i].id
s[i].name
→ Access using s[i].member

Note:- Used for storing multiple same-type data Loop is commonly used.

Example

```
#include <stdio.h>
struct student
{
    int id;
    char name[20];
};
int main()
{
    struct student s[2];
    for(int i=0; i<2; i++)
    {
        printf("enter id and name:\n");
        scanf("%d %s", &s[i].id, s[i].name);
    }
    for(int i=0; i<2; i++)
    {
        printf("ID: %d Name: %s\n", s[i].
               id, s[i].name);
    }
    return 0;
}
```

output:

```
Enter id and name:
1 Ram
Enter id and name:
2 Hari
```