Structures in C

Introduction

The array can be used to represent a group of data items that belong to the same type,such as int or float.however,we cannot use an array is we want to represent a collection of data items of different type using a single name.

c programming supports a constructed data type known as structure,a mechanism for packing data of different type.a structure is a convenient tool for handling a group of logically related data items.

Defining a Structure

Syntax

```
struct tag_name
{
        data_type member1;
        data_type member2;
        data_type member-n;
};
```

consider a emp database consisting of emp name,gender,and age,we can define a structure to hold this information as follows.

```
struct emp_data
{
        char name[30];
        char gender[5];
        int  age;

};
```

the keyword **struct** declares a structure to hold the details of three data fields,namely name,gender,and age.these fields are called structure members.

Declaring Variables

A structure variable declaration is similar to the declaration of variabl of any other data type.

Syntax

```
struct tag_name
{
                ---------
                ---------
                ---------
} var1,var2,var3,varn;
```

the above syntax var is structure variable name,you can define any name of variable,the following simple example of structure variable.

```
struct emp_data
{
        char name[30];
        char gender[5];
        int  age;

} emp1,emp2,emp2;
```

the above example is simple structure variable name emp1,emp2, and emp3 is structure variables.

Structure Initialization

the structure variable can be initialized at compile time,the following general example of structure initializing.

```
main()
{
        struct emp_data
        {
                int weight;
                float height;
        } emp1 = {50,165.50};

        ------
        ------
}
```

the above example 50 is emp1.weight and 165.50 is emp1.height.there is a one-to-one correspondence between the members and their initializing values.

Rules for Initializing Structures

there are a few rules to keep in mind while initializing structure variable at compile-time.

1. We cannot initialize individual members inside the structure template.

2. The order of values enclosed in braces must match the order of members in the structure definition.

3. it is permitted to have a partial initialization.we can initialize only the first few members and leave the remainig blank.the uninitialized members should be only at the end of the list.

4. The uninitialized members will be assigned default values as follows:

zero for integer and floating point numbers.

'\0' for characters and string.

Example

```
#include <stdio.h>
struct student
{
    char name[50];
    int roll;
    float marks;
};

void main()
{
    struct student s;
    clrscr();
    printf("Enter Students Info:\n\n");
    printf("Enter Student Name:");
    scanf("%s",s.name);
    printf("Enter roll number: ");
    scanf("%d",&s.roll);
    printf("Enter marks: ");
    scanf("%f",&s.marks);
    printf("\nThe Students Information\n");
    printf("Name: %s\n",s.name);
    printf("Roll Number: %d\n",s.roll);
    printf("Marks: %.2f\n",s.marks);
    getch();
}
```

Output

```
Enter Student Info:

Enter Student Name:Johan
Enter roll number:01
Enter marks:78

The Students Information

Name: Johan
Roll Number: 01
Marks: 78
```