Structure in C

In the C programming language, a structure (or `struct`) is a user-defined data type that allows grouping variables of different data types under a single name. This is particularly useful when you need to represent complex data with different types of attributes, like a person’s profile that has both numerical and string data.

Structures help organize data more efficiently, making the program more readable and easier to maintain.

# Syntax of Structure in C:

c

struct structure\_name {

data\_type member1;

data\_type member2;

// More members

};

- `struct`: Keyword used to declare a structure.

- `structure\_name`: The name you assign to the structure.

- `member1`, `member2`, etc.: Variables (called "members") that can be of different types.

Example of a Structure in C

Let’s say you want to represent information about a student. This information includes:

- Name (string)

- Roll number (integer)

- Marks (float)

Here’s how you can use a structure to store this data in C:

c

***#include <stdio.h>***

***#include <string.h>***

***// Define the structure***

***struct Student {***

***char name[50];***

***int roll\_number;***

***float marks;***

***};***

***int main() {***

***// Declare a structure variable***

***struct Student student1;***

***// Assign values to members of the structure***

***strcpy(student1.name, "Alice");***

***student1.roll\_number = 101;***

***student1.marks = 95.5;***

***// Access and print structure members***

***printf("Name: %s\n", student1.name);***

***printf("Roll Number: %d\n", student1.roll\_number);***

***printf("Marks: %.2f\n", student1.marks);***

***return 0;***

***}***

Breakdown of the Example:

1. Structure Definition:

- A `struct Student` is defined, containing three members: `name` (a string), `roll\_number` (an integer), and `marks` (a float).

2. Structure Declaration:

- `struct Student student1;` declares a variable `student1` of type `struct Student`.

3. Assigning Values:

- Values are assigned to the structure members using the `.` (dot) operator, for example, `student1.roll\_number = 101;`.

4. Accessing and Displaying:

- The members of the structure can be accessed and printed using the dot operator, as shown in `student1.name`, `student1.roll\_number`, and `student1.marks`.

Output:

Name: Alice

Roll Number: 101

Marks: 95.50

Advantages of Using Structures in C:

- Organizes complex data: Structures allow you to group different types of data, which makes it easier to manage.

- Easy to handle: Once defined, structures can be used like built-in types for creating arrays, pointers, or functions that operate on them.

Thus, structures provide a way to model complex data more effectively, offering better organization and readability in programs.