**Structure**

In **C programming**, a **structure** is a user-defined data type that groups related variables of different data types into a single unit. Structures are particularly useful when you need to work with related data as a single entity.

**Syntax for Declaring a Structure**

struct StructureName {

dataType member1;

dataType member2;

// More members...

};

Here:

* struct is the keyword.
* StructureName is the name of the structure.
* member1, member2, etc., are the variables inside the structure, called **members**.

**Example of a Structure in C**

Let’s create a structure to store information about a student (name, roll number, and marks).

#include <stdio.h>

#include <string.h> // For using strcpy()

// Define the structure

struct Student {

char name[50];

int rollNo;

float marks;

};

int main() {

// Declare a variable of type struct Student

struct Student student1;

// Assign values to members of student1

strcpy(student1.name, "Alice"); // Assign a string

student1.rollNo = 101; // Assign an integer

student1.marks = 95.5; // Assign a float

// Print the values

printf("Student Information:\n");

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

printf("Roll Number: %d\n", student1.rollNo);

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

return 0;

}

**Output**

Student Information:

Name: Alice

Roll Number: 101

Marks: 95.50

**Key Points About Structures**

1. **Grouped Data:** Structures allow you to group variables of different types.
2. **Accessing Members:** Use the dot operator (.) to access members of a structure.
3. **Multiple Instances:** You can create multiple variables of the same structure type.
4. **Memory Efficiency:** Structures organize data efficiently, but each member is stored in separate memory locations.