**Store Information and Display it Using Structure**

#include <stdio.h>

struct student

{

char name[50];

int roll;

float marks;

} s;

main()

{

printf("Enter information:\n");

printf("Enter name: ");

fgets(s.name, sizeof(s.name), stdin);

printf("Enter roll number: ");

scanf("%d", &s.roll);

printf("Enter marks: ");

scanf("%f", &s.marks);

printf("Displaying Information:\n");

printf("Name: ");

printf("%s", s.name);

printf("Roll number: %d\n", s.roll);

printf("Marks: %.1f\n", s.marks);

}

**Output**

Enter information:

Enter name: Jack

Enter roll number: 23

Enter marks: 34.5

Displaying Information:

Name: Jack

Roll number: 23

Marks: 34.5

In this program, a structure student is created. The structure has three members: name (string), roll (integer) and marks (float).

Then, a structure variable s is created to store information and display it on the screen.

# C Program to Add Two Distances (in inch-feet system) using Structures

In this example, you will learn to take two distances (in the inch-feet system), add them and display the result on the screen.

To understand this example, you should have the knowledge of the following [C programming](https://www.programiz.com/c-programming) topics:

* [C struct](https://www.programiz.com/c-programming/c-structures)

If you do not know, 12 inches is 1 foot.

## Program to add two distances in the inch-feet system

#include <stdio.h>

struct Distance

{

int feet;

float inch;

} d1, d2, result;

main()

{

// take first distance input

printf("Enter 1st distance\n");

printf("Enter feet: ");

scanf("%d", &d1.feet);

printf("Enter inch: ");

scanf("%f", &d1.inch);

// take second distance input

printf("\nEnter 2nd distance\n");

printf("Enter feet: ");

scanf("%d", &d2.feet);

printf("Enter inch: ");

scanf("%f", &d2.inch);

// adding distances

result.feet = d1.feet + d2.feet;

result.inch = d1.inch + d2.inch;

// convert inches to feet if greater than 12

while (result.inch >= 12.0) {

result.inch = result.inch - 12.0;

++result.feet;

}

printf("\nSum of distances = %d\'-%.1f\"", result.feet, result.inch);

}

**Output**

Enter 1st distance

Enter feet: 23

Enter inch: 8.6

Enter 2nd distance

Enter feet: 34

Enter inch: 2.4

Sum of distances = 57'-11.0"

In this program, a structure Distance is defined. The structure has two members:

* **feet** - an integer
* **inch** - a float

Two variables d1 and d2 of type struct Distance are created. These variables store distances in the feet and inches.

Then, the sum of these two distances are computed and stored in the result variable. Finally, result is printed on the screen.

# C Program to Store Information of Students Using Structure

In this example, you will learn to store the information of 5 students by using an array of structures.

To understand this example, you should have the knowledge of the following C programming topics:

* C Arrays
* C struct

## Store Information in Structure and Display it

#include <stdio.h>

struct student

{

char firstName[50];

int roll;

float marks;

} s[5];

main()

{

int i;

printf("Enter information of students:\n");

// storing information

for (i = 0; i < 5; ++i) {

s[i].roll = i + 1;

printf("\nFor roll number%d,\n", s[i].roll);

printf("Enter first name: ");

scanf("%s", s[i].firstName);

printf("Enter marks: ");

scanf("%f", &s[i].marks);

}

printf("Displaying Information:\n\n");

// displaying information

for (i = 0; i < 5; ++i) {

printf("\nRoll number: %d\n", i + 1);

printf("First name: ");

puts(s[i].firstName);

printf("Marks: %.1f", s[i].marks);

printf("\n");

}

}

**Output**

Enter information of students:

For roll number1,

Enter name: Tom

Enter marks: 98

For roll number2,

Enter name: Jerry

Enter marks: 89

Displaying Information:

Roll number: 1

Name: Tom

Marks: 98

In this program, a structure student is created. The structure has three members: name (string), roll (integer) and marks (float).

Then, we created an array of structures s having 5 elements to store information of 5 students.

Using a for loop, the program takes the information of 5 students from the user and stores it in the array of structure. Then using another for loop, the information entered by the user is displayed on the screen.

**Demonstrate the Dynamic Memory Allocation for Structure**

#include <stdio.h>

#include <stdlib.h>

struct course

{

int marks;

char subject[30];

};

main()

{

struct course \*ptr;

int noOfRecords;

printf("Enter the number of records: ");

scanf("%d", &noOfRecords);

// Memory allocation for noOfRecords structures

ptr = (struct course \*)malloc(noOfRecords \* sizeof(struct course));

for (int i = 0; i < noOfRecords; ++i) {

printf("Enter subject and marks:\n");

scanf("%s %d", (ptr + i)->subject, &(ptr + i)->marks);

}

printf("Displaying Information:\n");

for (int i = 0; i < noOfRecords; ++i) {

printf("%s\t%d\n", (ptr + i)->subject, (ptr + i)->marks);

}

free(ptr);

}

**Output**

Enter the number of records: 2

Enter subject and marks:

Science 82

Enter subject and marks:

DSA 73

Displaying Information:

Science 82

DSA 73