Structure Pointer in C

Last Updated : 23 Dec, 2024

A structure pointer is a pointer variable that stores the address of a structure. It allows the programmer to manipulate the structure and its members directly by referencing their memory location rather than passing the structure itself. In this article let's take a look at structure pointer in C.

Let's take a look at an example:

```c
#include <stdio.h>

struct A {
    int var;
};

int main() {
    struct A a = {30};

    // Creating a pointer to the structure
    struct A *ptr;

    // Assigning the address of person1 to the pointer
    ptr = &a;

    // Accessing structure members using the pointer
    printf("%d", ptr->var);

    return 0;
}
```

Explanation: In this example, ptr is a pointer to the structure A. It stores the address of the structure a, and the structure's member var is accessed using the pointer with the -> operator. This allows efficient access to the structure's members without directly using the structure variable.

Syntax of Structure Pointer

The syntax of structure pointer is similar to any other pointer to variable:

> *struct struct_name* ***ptr_name;***

Here, **struct_name** is the name of the structure, and **ptr_name** is the name of the pointer variable.

Accessing Member using Structure Pointers

There are two ways to access the members of the structure with the help of a structure pointer:

1. Differencing and Using (.) Dot Operator.
2. Using (->) Arrow operator.

Differencing and Using (.) Dot Operator

First method is to first dereference the structure pointer to get to the structure and then use the dot operator to access the member. Below is the program to access the structure members using the structure pointer with the **help of the dot operator.**

```
#include <stdio.h>
#include <string.h>

struct Student {
    int roll_no;
    char name[30];
    char branch[40];
    int batch;
```

```c
    struct Student s1 = {27, "Geek", "CSE", 2019};

    // Pointer to s1
    struct Student* ptr = &s1;

    // Accessing using dot operator
    printf("%d\n", (*ptr).roll_no);
    printf("%s\n", (*ptr).name);
    printf("%s\n", (*ptr).branch);
    printf("%d", (*ptr).batch);

    return 0;
}
```

Output

```
27
Geek
CSE
2019
```

Using (->) Arrow Operator

C language provides an array operator (->) that can be used to directly access the structure member without using two separate operators. Below is the program to access the structure members using the structure pointer with the **help of the Arrow operator.**

```c
#include <stdio.h>
#include <string.h>

struct Student {
    int roll_no;
    char name[30];
    char branch[40];
    int batch;
};
```

```
    // Pointer to s1
    struct Student* ptr = &s1;

    // Accessing using dot operator
    printf("%d\n", ptr->roll_no);
    printf("%s\n", ptr->name);
    printf("%s\n", ptr->branch);
    printf("%d", ptr->batch);

    return 0;
}
```

Output

```
27
Geek
CSE
2019
```

Explanation: In this code, a **struct Person** is defined with name and age as members. A pointer ptr is used to store the address of person1. The arrow operator (**->**) is used to access and modify the members of the structure via the pointer, updating the **name** and **age** of person1, and printing the updated values.

Structure Pointer

Visit Course

Comment More info

Campus Training Program

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:
K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305

Company

About Us

Legal

Privacy Policy

Careers

Contact Us

Corporate Solution

Campus Training Program

Explore

POTD

Job-A-Thon

Connect

Community

Videos

Blogs

Nation Skill Up

Tutorials

Programming Languages

DSA

Web Technology

AI, ML & Data Science

DevOps

CS Core Subjects

Interview Preparation

GATE

School Subjects

Software and Tools

Courses

IBM Certification

DSA and Placements

Web Development

Data Science

Programming Languages

DevOps & Cloud

GATE

Trending Technologies

Offline Centers

Noida

Bengaluru

Pune

Hyderabad

Patna

Preparation Corner

Aptitude

Puzzles

GfG 160

DSA 360

System Design