

Structure Pointers and Nested Structure



Overview

- C allows programmers to create user-defined data types by grouping data of different types together using struct keywords, such data types are called structures. Like any other data type in C, variables of user-defined structure occupy addresses in a memory block, and pointers can be used to point them. A pointer pointing to a structure is called structure pointer. Structures and pointers in C together help in accessing structure members efficiently. Structure pointer declaration is similar to declaring a structure variable using the struct keyword followed by the type of structure it will point to. A structure pointer can only hold the address of the structure used at the time of its declaration. Structures and pointers in C together make accessing structure value and passing to functions easier.

Structure Pointers

- Different components of a structure are called members for example, in the above case, student name and roll number are members of the structure. Like every other data type, structure variables are stored in memory, and we can use pointers to store their addresses. Structure pointer points to the address of the structure variable in the memory block to which it points. This pointer can be used to access and change the value of structure members. This way, structures and pointers in C can be used to create and access user-defined data types conveniently.

Syntax to Define a Structure

```
struct structure_name {  
    data_type member_variable_1;  
    data_type member_variable_2;  
    // ... data_type member_variable_N;  
};
```

- C struct keyword is used to create a new data type, followed by the structure name. We define different members of the structure inside parenthesis. Once a structure is defined, its name structure_name can be used to declare variables as

```
struct structure_name structure_variable;
```

Declare a Struct Pointer

- Structure pointer in C is declared using the keyword struct followed by structure name to which the pointer will point to followed by pointer name. A structure pointer can only hold the address of a variable of the same structure type used in its declaration.

```
struct structure_name *structure_pointer;
```

- This way structures and pointers in C are used together to create a pointer pointing to the structure.

Initialising of a Structure Pointer

- After a structure pointer is declared, we need to initialize it to a variable before using it. To initialize a variable, we need to provide the address of the structure variable using the & operator.

```
structure_pointer = &structure_variable;
```

- Also, the structure pointer can be initialized during the time of declaration.

```
struct structure_type *structure_pointer = &structure_variable;
```


Accessing Structure Member using a Pointer

- There are two ways to access the values of structure members using pointers –
 1. Using asterisk (*) and dot (.) operator with the structure pointer.
 2. Using membership or arrow (->) operator.

Nested Structure

- A **nested structure** in C is a structure within structure. One structure can be declared inside another structure in the same way structure members are declared inside a structure.
- The member of a nested structure can be accessed using the following syntax:
Variable name of Outer_Structure.Variable name of Nested_Structure.data member to access

Examples of Nested Structure

- Consider there are two structures Employee (dependent structure) and another structure called Organisation(Outer structure).
- The structure Organisation has the data members like organisation_name,organisation_number.
- The Employee structure is nested inside the structure Organisation and it has the data members like employee_id, name, salary.
- For accessing the members of Organisation and Employee following syntax will be used:
 - org.emp.employee_id;
 - org.emp.name;
 - org.emp.salary;
 - org.organisation_name;
 - org.organisation_number;
- Here, org is the structure variable of the outer structure Organisation and emp is the structure variable of the inner structure Employee.

Different Ways of Nesting Structures

- **By separate nested structure:** In this method, the two structures are created, but the dependent structure(Employee) should be used inside the main structure(Organisation) as a member.
- **By Embedded nested structure:** Using this method, allows to declare structure inside a structure and it requires fewer lines of code.

Drawbacks of nested Structure

- **Independent existence not possible:** It is important to note that structure Employee doesn't exist on its own. One can't declare structure variable of type struct Employee anywhere else in the program.
- **Cannot be used in multiple data structures:** The nested structure cannot be used in multiple structures due to the limitation of declaring structure variables within the main structure. So, the most recommended way is to use a separate structure and it can be used in multiple data structures

Passing Nested Structure to Function

- A nested structure can be passed into the function in two ways:
 - Pass the nested structure variable at once.
 - Pass the nested structure members as an argument into the function.

Accessing Nested Structure

- **Normal Variable** - Outer and inner structure variables are declared as normal variables and the data members of the outer structure are accessed using a single dot(.) and the data members of the inner structure are accessed using the two dots
- **Pointer Variable** - One normal variable and one pointer variable of the structure are declared to explain the difference between the two. In the case of the pointer variable, a combination of dot(.) and arrow(->) will be used to access the data members.

THANK YOU

