

Structures

A **structure in C** is a collection of variables, possibly different types, under a single name. Each member is allocated its own memory space, and the structure's size is the sum of the sizes of all its members.

Syntax:

```
struct name {  
    member1 definition;  
    member2 definition;  
    ...  
    memberN definition;  
};
```

Unions

A **union in C** is similar to a structure but with a key difference: all members share the same memory location. This means only one member can store a value at any given time. The size of the union's largest member determines the union size.

Syntax:

```
union name {  
    member1 definition;  
    member2 definition;  
    ...  
    memberN definition;  
};
```

Similarities Between Structure and Union

Structures and unions are also similar in some aspects listed below:

- Both are user-defined data types used to store data of different types as a single unit.
- Their members can be objects of any type, including other structures, unions, or arrays. A member can also consist of a bit field.
- Both structures and unions support only assignment = and sizeof operators. The two structures or unions in the assignment must have the same members and member types.
- A structure or a union can be passed by value to functions and returned by value by functions. The argument must have the same type as the function parameter. A structure or union is passed by a value like a scalar variable as a corresponding parameter.
- '·' operator or selection operator, which has one of the highest precedence, is used for accessing member variables inside both the user-defined datatypes.

The table below lists the primary differences between the C structures and unions:

Parameter	Structure
Definition	A structure is a user-defined data type that groups different data types into a single entity.
Keyword	The keyword <i>*struct*</i> is used to define a structure.
Size	The size is the sum of the sizes of all members, with padding if necessary.
Memory Allocation	Each member within a structure is allocated a unique storage area of location.
Data Overlap	No data overlap as members are independent.
Accessing Members	Individual members can be accessed at a time.
