Difference Between Structure and Union in C

What is a Structure in C?

A Structure is a type of data that is user-defined. It is available in the C programming language that allows a user to combine logically related data items of various data types. Structures basically represent a record. All its elements get stored in the contiguous memory locations. A Structure type- the Variables- can store more than one data item from raging data types all under one name.

What Is a Union?

A Union is a type of data that is user-defined. It is just like the structure. The Union combines various objects of different sorts and sizes together. A user can define a Union using many members, but only one of them holds a value at any given time. It provides you with an efficient way of using a single memory location for various purposes. Thus, varying objects can share a similar location.

Functions and Similarities Between Union and Structure in C

- They are both user-defined data types, and they store different sorts of data together as a single unit.
- Both of their members can be any type of object. It may include different structures and unions/ arrays. Its members can also contain a bit of field.
- A Union or a Structure can easily pass by value to functions and return to the value by functions. Every argument must possess the same parameters as that of the function parameter.
- A Union or Structure passes by the value just like any scalar variable in the form of a corresponding parameter.
- You can use the “.” operator for accessing the members.

Parameter	Structure	Union
Keyword	A user can deploy the keyword **struct** to define a Structure.	A user can deploy the keyword **union** to define a Union.
Internal Implementation	The implementation of Structure in C occurs internally- because it contains separate memory locations allotted to every input member.	In the case of a Union, the memory allocation occurs for only one member with the largest size among all the input variables. It shares the same location among all these members/objects.
Accessing Members	A user can access individual members at a given time.	A user can access only one member at a given time.
Syntax	The Syntax of declaring a Structure in C is: struct [structure name] { type element_1; type element_2; . . } variable_1, variable_2;	The Syntax of declaring a Union in C is: union [union name] { type element_1; type element_2; . . } variable_1, variable_2;
Size	A Structure does not have a shared location for all its members. It makes the size of a Structure to be greater than or equal to the sum of the size of its data members.	A Union does not have a separate location for every member in it. It makes its size equal to the size of the largest member among all the data members.
Storage of Value	In the case of a Structure, there is a specific memory location for every input data member. Thus, it can store	In the case of a Union, there is an allocation of only one shared memory for all the input data members. Thus, it stores one

	multiple values of the various members.	value at a time for all of its members.
Initialization	In the case of a Structure, a user can initialize multiple members at the same time.	In the case of a Union, a user can only initiate the first member at a time.