Similarities and Difference Between Union and Structure in C Language

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 field.
- A Union or a Structure can easily pass by value to functions and also 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.

Difference Between Structure and Union in C

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] {	The Syntax of declaring a Union in C is: union [union name] {

	type element_1; type element_2; . . } variable_1, variable_2, …;	type element_1; type element_2; . . } variable_1, variable_2, …;
Size	A Structure does not have a shared location for all of 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.
Value Altering	Altering the values of a single member does not affect the other members of a Structure.	When you alter the values of a single member, it affects the values of other 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 multiple values of the various members.	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 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.

Keep learning and stay tuned