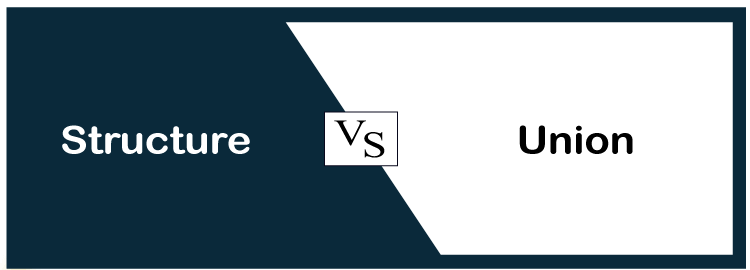
Difference between Structure and Union

**Structure** and **union** both are *user-defined* data types in the  C/C++ programming language. In this section, we will see what the Structure and Union are; and the differences between them.



What is a structure ?

[**Structure (struct)**](https://www.javatpoint.com/structure-in-c)  is a user-defined data type in a programming language that stores different data types' values together. The **struct** keyword is used to define a structure data type in a program. The struct data type stores one or more than one data element of different kinds in a variable.

Suppose that you want to store the data of employee in your C/C++ project, where you have to store the following different parameters:

* Id
* Name
* Department
* Email Address

One way to store 4 different data by creating 4 different arrays for each parameter, such as id[], name[], department[], and email[]. Using array id[i] represents the id of the ith employee. Similarly, name[i] represents the name of ith employee (name). Array element department[i] and email[i] represent the ith employee's department and email address

+The advantage of using a separate array for each parameter is simple if there are only a few parameters. The disadvantage of such logic is that it is quite difficult to manage employee data. Think about a scenario in which you have to deal with 100 or even more parameters associated with one employee. It isn't easy to manage 100 or even more arrays. In such a condition, a **struct** comes in.

Syntax of struct

1. **struct** [structure\_name]
2. {
3. type member\_1;
4. type member\_2;
5. . . .
6. type member\_n;
7. };

Example

1. **struct** employee
2. {
3. **int** id;
4. **char** name[50];
5. string department;
6. string email;
7. };

What is a Union?

**In "c," programming** union is *a user-defined data type* that is used to store the different data type's values. However, in the union, one member will occupy the memory at once. In other words, we can say that the size of the union is equal to the size of its largest data member size. Union offers an effective way to use the same memory location several times by each data member. The **union** keyword is used to define and create a union data type.

Syntax of Union

1. **union** [**union** name]
2. {
3. type member\_1;
4. type member\_2;
5. . . .
6. type member\_n;
7. };

Example

1. **union** employee
2. {
3. string name;
4. string department;
5. **int** phone;
6. string email;
7. };