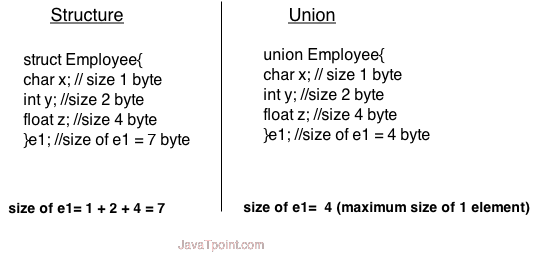
# C Union

Like structure, **Union in c language** is a user defined datatype that is used to hold different type of elements.

But it doesn't occupy sum of all members size. It occupies the memory of largest member only. It shares memory of largest member.



### **Advantage of union over structure**

It **occupies less memory** because it occupies the memory of largest member only.

### **Disadvantage of union over structure**

It can **store data in one member only**.

## **Defining union**

The **union** keyword is used to define union. Let's see the syntax to define union in c.

1. **union** union\_name
2. {
3. data\_type member1;
4. data\_type member2;
5. .
6. .
7. data\_type memeberN;
8. };

Let's see the example to define union for employee in c.

1. **union** employee
2. {   **int** id;
3. **char** name[50];
4. **float** salary;
5. };

#### C Union example

Let's see a simple example of union in C language.

1. #include <stdio.h>
2. #include <string.h>
3. **union** employee
4. {   **int** id;
5. **char** name[50];
6. }e1;  //declaring e1 variable for union
7. **int** main( )
8. {
9. //store first employee information
10. e1.id=101;
11. strcpy(e1.name, "Sonoo Jaiswal");//copying string into char array
12. //printing first employee information
13. printf( "employee 1 id : %d\n", e1.id);
14. printf( "employee 1 name : %s\n", e1.name);
15. **return** 0;
16. }

Output:

employee 1 id : 1869508435

employee 1 name : Sonoo Jaiswal

As you can see, id gets garbage value because name has large memory size. So only name will have actual value.