

Union

- how to create unions
- access its members
- learn the similarities between unions and structures
- learn the differences between unions and structures.

Just like Structures, the union is a user-defined data type. All the members in union share the same memory location. The union is a data type that allows different data belong to different data types to be stored in the same memory locations. One of the advantages of using a union is that it provides an efficient way of reusing the memory location, as only one of its members can be accessed at a time. A union is used in the same way we declare and use a structure.

Defining union

We use the union keyword to define the union. The syntax to define union in C is given below.

```
union union_name
{
    datatype member1;
    datatype member2;
};
```

Following is the example of Union in C

```
union books
{
    int pages;
    float price;
    char title[20];
}b1;
```

This declares a variable b1 of type union books. This union contains three members each with a different data type, price belongs to float data type, pages belong to integer data type and title belongs to character datatype. However, we can use only one of them at a time. This is because, only one location is allocated for all the union variables, irrespective of their size.

How to access the members of a union

We use "." operator to access the members of a union.

Example: Accessing Union Members

```
#include <stdio.h>
#include <string.h>

union Book {
    int pages;
    float price;
    char title[20];
};

int main( ) {
    union Book b1;
    b1.pages = 100;
    printf( "Pages: %d\n", b1.pages);
    b1.price = 250.5;
    printf( "Price : %.1f\n", b1.price);
    strcpy( b1.title, "C Programming");
    printf( "Title : %s\n", b1.title);

    return 0;
}
```

What are the similarities between Structure and Union

1. Structure and union are user-defined data types used to store data of different types.
2. The members of structure and union can be objects of any type, including other structures and unions or arrays.
3. A union or a structure can be passed by value to functions and returned by value by functions.
4. '.' operator is used for accessing union and structure members.

What are the differences between Structure and Union

1. The keyword union is used to define a union and a keyword struct is used to define the structure
2. Each member within a structure is assigned a unique storage area of location whereas memory allocated is shared by individual members of the union.
3. Individual members can be accessed at a time in structure whereas only one member can be accessed at a time in union
4. Altering the value of the member will not affect other members of the structure, whereas altering the value of any member will affect other member's values.
5. Several members of a structure can be initialized at once, whereas only one member can be initialized in union