Course Name: O Level (2nd Sem B1, B2 and B3 Batch) **Subject:** C Language
Topic: Union in C **Date:** 08-June-2020

What is Union?

Union is user defined data type used to stored data under unique variable name at single memory location. Syntax of union is similar to structure. But the major **difference between structure and union is 'storage.'** In structures, each member has its own storage location, whereas all the members of union use the same location. Union contains many members of different types, it can handle only one member at a time.

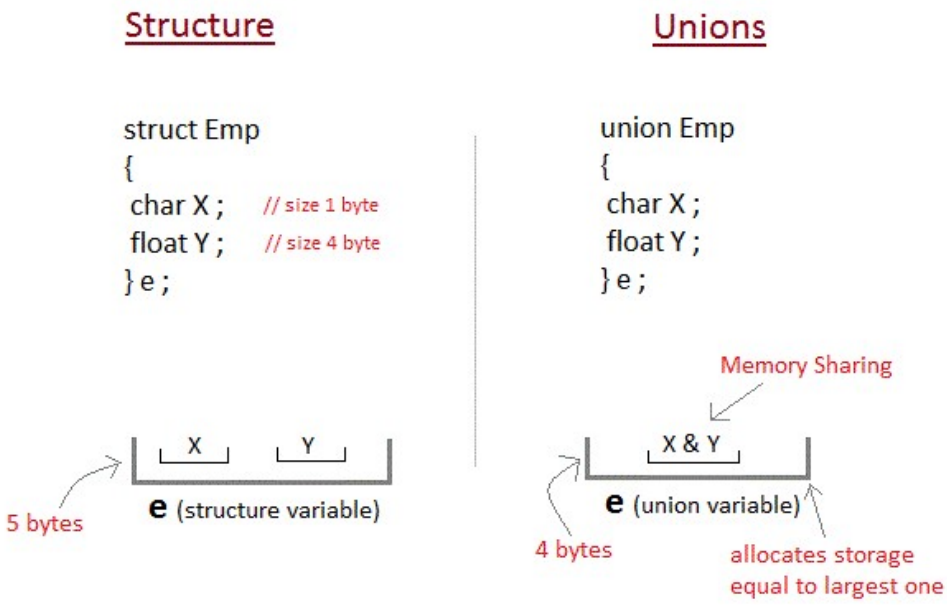
Defining a Union

Syntax:

```
union union_name
{
  <data-type> element 1;
  <data-type> element 2;
   <data-type> element 3;
}union_variable;
```

Members of union with different data types

Example :



Accessing Union Members

- To access any member of a union, we use the **member access operator (.)**.
- The member access operator is coded as a period between the union variable name and the union member that we wish to access.
- You would use the keyword **union** to define variables of union type.

The following example shows how to use unions in a program:

```
#include <stdio.h>
#include <conio.h>
union student
{
  int roll;
  float per;
  char name[20];
};
```

```
void main( )
{
  union student data;
  clrscr();
  data.roll = 10;
  data.per = 40.5;
  data.name = "C Programming";
  printf( "data.roll : %d\n", data.roll);
  printf( "data.per : %f\n", data.per);
  printf( "data.name : %s\n", data.name);
  getch();
}
```

When the above code is compiled and executed, it produces the following result

data.roll : 1917853763
data.per: 41223605803277948604527599943 68.000000
data.anem: C Programming

Here, we can see that the values of **roll** and **per** members of union got corrupted because the final value assigned to the variable has occupied the memory location and this is the reason that the value of **name** member is getting printed very well.

Now let's look into the same example once again where we will use one variable at a time which is the main purpose of having unions

```
#include <stdio.h>
#include <conio.h>
union student
{
  int roll;
  float per;
  char name[20];
};
void main( )
{
 union student data;
 clrscr();
 data.roll = 10;
 printf( "data.roll : %d\n", data.roll);
 data.per = 40.5;
 printf( "data.per : %f\n", data.per);
 data.str = "C Programming";
 printf( "data.name : %s\n", data.name);
 getch();
}
```

When the above code is compiled and executed, it produces the following result

data.roll : 10
data.per : 40.500000
data.name : C Programming

Here, all the members are getting printed very well because one member is being used at a time

Advantage of union over structure

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

Disadvantage of union over structure

- Only the last entered data can be stored in the union. It overwrites the data previously stored in the union.

Similarities between Structure and Union

- Both are user-defined data types used to store data of different types as a single unit.
- Their members can be objects of any type, including other structures and unions or arrays. A member can also consist of a bit field.
- Both structures and unions support only assignment = and sizeof operators. The two structures or unions in the assignment must have the same members and member types.
- A structure or a union can be passed by value to functions and returned by value by functions. The argument must have the same type as the function parameter. A structure or union is passed by value just like a scalar variable as a corresponding parameter.
- **'.'** operator is used for accessing members.

Differences between Structure and Union

	Structure	Union
Keyword	The keyword **struct** is used to define a structure	The keyword **union** is used to define a union
Size	When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is **equal to the sum of sizes of its members**.	When a variable is associated with a union, the compiler allocates the memory by considering the size of the largest memory. So, size of union is **equal to the sum of sizes of largest members**.
Memory	Each member within a structure is assigned unique storage area of location.	Memory allocated shared by individual member of union.
Value Altering	Altering the value of a member will not affect other members of the structure.	Altering the value of any of the member will alter other member values.
Accessing Members	Individual member can be accessed at a time.	Only one member can be accessed at a time.
Initialization of Members	Several members of a structure can initialize at once.	Only one member of a union can be initialized

Assignment:

1. What is union?
2. Write some similarities between structure and union.
3. Difference between structure and union.