UNION

- Union can be defined as a user-defined data type which is a collection of different variables of different data types in the same memory location.
- The union can also be defined as many members, but only one member can contain a value at a particular point in time.
- Unions provide an efficient way of using the same memory location for multiple-purpose.
- Union is a user-defined data type, but unlike structures, they share the same memory location.

Defining a Union

- To define a union, you must use the union statement in the same way as did while defining a structure.
- The union statement defines a new data type with more than one member for your program. The format of the union statement is as follows:

```
union [union tag] {
  member definition;
  member definition;
  ...
  member definition;
} [one or more union variables];
```

- The union tag is optional and each member definition is a normal variable definition, such as int i; or float f; or any other valid variable definition.
- At the end of the union's definition, before the final semicolon, you can specify one or more union variables but it is optional.
- Here is the way you would define a union type named Data having three members i, f, and str.

```
union Data {
  int i;
  float f;
  char str[20];
```

```
} data;
```

- Now, a variable of Data type can store an integer, a floating-point number, or a string of characters.
- It means a single variable, i.e., same memory location, can be used to store multiple types of data.
- You can use any built-in or user defined data types inside a union based on your requirement.

Example Program 2.9 Illustration of Union

```
#include <stdio.h>
#include <string.h>
union Data {
  int i;
  float f;
  char str[20];
};
void main( ) {
  union Data data;
  data.i = 10;
  printf( "data.i : %d\n", data.i);
  data.f = 220.5;
  printf( "data.f : %f\n", data.f);
 strcpy( data.str, "Charulatha publication");
  printf( "data.str : %s\n", data.str);
}
```

Output

```
data.i : 10
data.f : 220.500000
data.str : Charulatha publication
```

Difference between Structure and Union

Sl.No	Structure	Union
1	The member of a structure occupies its own memory space.	The member of union share same memory space.
2	The keyword struct is used to define a structure	The keyword union is used to define a structure
3	All the members of a structure can be initialized.	Only the first member of a union can be initialized.
4	In structure, each member is stored in a separate memory location. So need more memory space.	In union, all members are stored in the same memory locations. So, need less memory space.