

# UNIONS IN C PROGRAMMING

A union is a variable which may hold members of different sizes and types. The syntax for declaring a union is similar to that of a structure:

```
union number{
    int number;
    double floatnumber;
} anumber;
```

This defines a union called "number" and a variable of it called "anumber". Here, *number* is the name of the union, and acts in the same way as a tag for a structure.

Members of the union can be accessed in the following way:

```
printf("%f",anumber.floatnumber);
```

**Note :** When the C compiler is allocating memory for unions it will always reserve enough room for the largest member. In the above example it is 8 bytes for the double. Whereas, if the above union was a structure, then the C compiler would have reserved 10 bytes which is the total sum of the size of the individual elements.

## Program 7.1



```

#include <stdio.h>

main ()
{
    union id {
        char name[40];
        int number;
    };
    struct {
        int salary;
        union id description;
    } student, faculty;

    printf("%d\n", sizeof(union id));
    student.description.name = "Sam";
    printf("%s %d\n",
student.description.name, student.description.number);
    student.description.number = 12;
    printf("%s %d\n",
student.description.name, student.description.number);

}

```

Notice that in the above program the value 'Sam' is assigned to the union member `student.description.name`, so the first `printf()` statement produces an output 40, which is the size of the union in bytes. The second and third `printf()` statements produce the following outputs respectively :

Sam -33000

# 12

Here, in the first line, the first value is meaningful, but the second one is not; similarly in the second line, the second value is meaningful. This clearly indicates that the union can contain any one value at any given time and not both.

Most C compilers allow an entire union to be assigned to another, provided both of them have the same number of members and are of the same data type. These compilers also permit entire unions to be passed to or from a function.

Source : <http://www.peoi.org/Courses/Coursesen/cprog/frame8.html>