Logging in From a Linux System or Localhost
4 days 40 min ago

Auto response
4 days 6 hours ago

Re: final issue.
4 days 22 hours ago

Re: Rocky, thank you for this
4 days 23 hours ago

English | Deutsch | Site Map/RSS Feeds | Advertise

You are here: **Home** » **Learning C/C++ Step-By-Step** » **Learning C/C++ Step-By-Step - Page 11**

Learning C/C++ Step-By-Step - Page 11

Submitted by ganesh35 (Contact Author) (Forums) on Wed, 2009-01-07 18:36. ::

0 Tweet

11. Step-by-Step C/C++ --- C Programming - Unions

Unions

1. Introduction

```
/* 81_union.c */
#include <stdio.h>
struct s_emp
{
            int eno;
            char name[20];
            float sal;
};
union u_emp
{
            int eno;
            char name[20];
            float sal;
};
int main()
{
            struct s_emp se;
            union u_emp ue;
            printf("\nSize of Employee structure    :  %d", sizeof(se));
            printf("\nSize of Employee Union      :  %d", sizeof(ue));
            return 0;
}
```

Output: Size of Employee Structure : 26
Size of Employee Union : 20

2. About Union

When a large number of variables are requested to use in a program. They were occupies a large amount of memory. Unions provide an easiest way to save memory by using replacement technique. It uses same memory location for all type of variables.

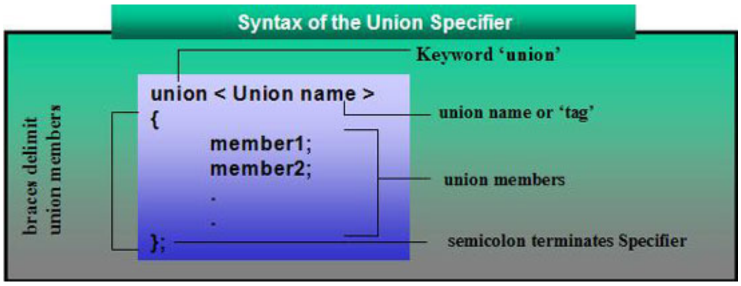
A union is a data type in C, which allows the overlay of more than one variable in the same memory area.

Characteristics of Unions:

1. Union stores values of different types in a single location in memory.
2. A union may contain one of many different types of values but only one is stored at a time.
3. The union only holds a value for one data type. If a new assignment is made the previous value has no validity.
4. Any number of union members can be present. But union type variable takes the largest memory occupied by its members.

3. Declaration of a Union

Union is a data type through which objects of different types and sizes can be stored at different times. Definition of a Union is same as a Structure. The only change in the declaration is the substitution of the keyword union for the keyword struct.



Eg:

```
union ddate
{
    int day;
    int month;
    int year;
};
```

```
union student
{
    int sno;
    char name[20];
    int marks;
    float avg;
};
```

4. Defining a Union Variable

Defining a Union variable is the same as structure and that for defining a built-in data type such as int.

```
int a; /* Valid */
union date d; /* Valid in both C and C++ */
```

Calculation of Union size

Every data type in C/C++ has a specified size, i.e int has 2 bytes of size, float has 4 bytes of size and so on. Here is the way to find the size of a Union variable.
sizeof :- This function is used to find the size of a given variable.

```
printf("%d", sizeof(int)); /* 2 */
printf("%d", sizeof(float)); /* 4 */
printf("%d", sizeof(union emp)); /* Displays the size of the emp union */
```

5. Difference between Structures and Unions

Here is the difference between Structures and Unions

Structure	Union
	1. It can hold different types (variables) in different locations.
1. It can hold different types (variables) in a single location.	
2. It may contain more than one type (variable) but only one is stored at a time.	2. It may contain more than one type (variable) all are stored in memory at a time.
3. Any number of union members can be present. But union type variable takes the largest memory occupied by its member.	3. It requires memory of the size of all its members.
4. On its process only one member can be accessed at any given time.	4. On its process all the members can be access at any time.
5. The scope of union is the function and the scope of its members is also same as the union itself. (They can be accessed directly in the program).	5. The scope of Structure is the function only. Structure members are unable to access directly in the program.

6. Operations on Unions

A union is also similar to structure it can perform all the operations like structures. Operations on Union are listed below.

- A union variable can be assigned to another union variable.
- A union Variable can be passed to a function as a parameter
- The address of the union variable can be extracted by using the address-of operator (&).
- A function can accept and return a union or a pointer to a union.

```
/* 82_union.c */
#include <stdio.h>
union u_emp
{
       int eno;
       char name[20];
       float sal;
};
int main()
{
       union u_emp ue;
       printf("Enter Employee Number  : "); scanf("%d", &ue.eno);
       printf("Enter Employee Name   : "); scanf("%s", ue.name);
       printf("Enter Employee Salary  : "); scanf("%f", &ue.sal);
       printf("\n\nEmployee Details are as follows...\n");
       printf("%d %s %f ", ue.eno, ue.name, ue.sal);
       return 0;
}
```

What is the output?
Only ue.sal is correct. What about rest of variables.

At any instant only one of the union variables will have a meaningful value. Only that member, who is last written, can be read. At this point, other variables will contain garbage. It is the responsibility of the programmer to keep track of the active variable (i.e. variable which was last accessed).

Here is the best way to accept and display records of an employee.

```
/* 83_emp.c */
#include <stdio.h>
union u_emp
{
       int eno;
       char name[20];
       float sal;
};
int main()
{
       union u_emp ue;
       printf("\nEnter Employee Number  : "); scanf("%d", &ue.eno);
       printf("\n%d", ue.eno);
       printf("\nEnter Employee Name : "); scanf("%s", ue.name);
       printf("\n%s", ue.name);
       printf("\nEnter Employee Salary  : "); scanf("%f", &ue.sal);
       printf("\n%f",ue.sal);
       return 0;
}
```

7. Scope of a Union

The scope of union is different than structure. A structure variable can be accessed by the its functions only. Where as a union and its members can be accessed by its function.

```
/* 84_scope.c */
#include <stdio.h>
int main()
{
    union
    {
        int i;
        char c;
        float f;
    };
    i = 10; c = ‘a’; f = 4.5; /* Union members */
    printf("The value of c is : %c", c);
    return 0;
}
```

previous up next
Learning C/C++ Step-By-Step - Page 10 Learning C/C++ Step-By-Step - Page 12

0 Tweet

add comment | view as pdf | print: this | all page(s)

Related Tutorials

- Beginner's Guide To c++
- An Explanation of Pointers (C++)

⚠ *Please do not use the comment function to ask for help! If you need help, please use our forum.*
Comments will be published after administrator approval.

Mistake in difference
Submitted by Snehal (not registered) on Mon, 2009-06-15 08:00.

In the difference between union and structure ,the names are misplaced ...means in union , structure points are placed and viceversa... Make it correct

reply | view as pdf

Howtos | Mini-Howtos | Forums | News | Search | Contribute | Subscription
Site Map/RSS Feeds | Advertise | Contact | Disclaimer | Imprint