Structure :

Structure is user defined data type which is used to store heterogeneous data under unique name. Keyword 'struct' is used to declare structure.

The variables which are declared inside the structure are called as 'members of structure'.

Syntax:

```
struct structure_nm
{
          <data-type> element 1;
          <data-type> element 2;
          - - - - - - - - - - -
          - - - - - - - - - - -
          <data-type> element n;
}struct_var;
```

Example :

```
struct emp_info
{
          char emp_id[10];
          char nm[100];
          float sal;
}emp;
```

Note :

1. Structure is always terminated with semicolon (;).

2. Structure name as emp_info can be later used to declare structure variables of its type in a program.

* Instances of Structure :

Instances of structure can be created in two ways as,

Instance 1:

```
struct emp_info
{
          char emp_id[10];
          char nm[100];
          float sal;
}emp;
```

Instance 2:

```
struct emp_info
{
          char emp_id[10];
          char nm[100];
          float sal;
};
struct emp_info emp;
```

In above example, emp_info is a simple structure which consists of stucture members as Employee ID(emp_id), Employee Name(nm), Employee Salary(sal).

* Aceessing Structure Members :

Structure members can be accessed using member operator '**.**' . It is also called as '**dot operator**' or '**period operator**'.

```
structure_var.member;
```

Program :

```
/*  Program to demonstrate structure.*/

#include <stdio.h>
#include <conio.h>

struct comp_info
{
          char nm[100];
          char addr[100];
}info;

void main()
{
          clrscr();
          printf("\n Enter Company Name : ");
          gets(info.nm);
          printf("\n Enter Address : ");
          gets(info.addr);
          printf("\n\n Company Name : %s",info.nm);
          printf("\n\n Address : %s",info.addr);
          getch();
}
```

Output :

```
 Enter Company Name : TechnoExam, Technowell Web Solutions
 Enter Address : Sangli, Maharashtra, INDIA

 Company Name : TechnoExam, Technowell Web Solutions
 Address : Sangli, Maharashtra, INDIA_
```

Array in Structures :

Sometimes, it is necessary to use structure members with array.

Program :

```
/*  Program to demonstrate array in structures.*/

#include <stdio.h>
#include <conio.h>

struct result
{
          int rno, mrks[5];
          char nm;
}res;

void main()
{
          int i,total;
          clrscr();
          total = 0;
          printf("\n\t Enter Roll Number : ");
          scanf("%d",&res.rno);
          printf("\n\t Enter Marks of 3 Subjects : ");
          for(i=0;i<3;i++)
          {
                    scanf("%d",&res.mrks[i]);
                    total = total + res.mrks[i];
          }
          printf("\n\n\t Roll Number : %d",res.rno);
          printf("\n\n\t Marks are :");
          for(i=0;i<3;i++)
          {
                    printf(" %d",res.mrks[i]);
          }
          printf("\n\n\t Total is : %d",total);
          getch();
}
```

Output :

```
Enter Roll Number : 1

Enter Marks of 3 Subjects : 63 66 68

Roll Number : 1

Marks are : 63 66 68

Total is : 197_
```

Structure With Array :

We can create structures with array for ease of operations in case of getting multiple same fields.

Program :

```
/*  Program to demonstrate Structure With Array. */

#include <stdio.h>
#include <conio.h>

struct emp_info
{
          int emp_id;
          char nm[50];
}emp[2];

void main()
{
          int i;
          clrscr();
          for(i=0;i<2;i++)
          {
                    printf("\n\n\t Enter Employee ID : ");
                    scanf("%d",&emp[i].emp_id);
                    printf("\n\n\t Employee Name : ");
                    scanf("%s",emp[i].nm);
          }
          for(i=0;i<2;i++)
          {
             printf("\n\t Employee ID : %d",emp[i].emp_id);
             printf("\n\t Employee Name : %s",emp[i].nm);
          }
          getch();
}
```

Output :

```
        Enter Employee ID : 1

        Employee Name : ABC

        Enter Employee ID : 2

        Employee Name : XYZ

        Employee ID : 1
        Employee Name : ABC
        Employee ID : 2
        Employee Name : XYZ_
```

Structures within Structures (Nested Structures) :

Structures can be used as structures within structures. It is also called as 'nesting of structures'.

Syntax:

```
struct structure_nm
{
          <data-type> element 1;
          <data-type> element 2;
          - - - - - - - - - - -
          - - - - - - - - - - -
          <data-type> element n;

          struct structure_nm
          {
                    <data-type> element 1;
                    <data-type> element 2;
                    - - - - - - - - - - -
                    - - - - - - - - - - -
                    <data-type> element n;
          }inner_struct_var;
}outer_struct_var;
```

Example :

```
struct stud_Res
{
          int rno;
          char nm[50];
          char std[10];

          struct stud_subj
          {
                    char subjnm[30];
                    int marks;
          }subj;
}result;
```

In above example, the structure stud_Res consists of stud_subj which itself is a structure with two members. Structure stud_Res is called as 'outer structure' while stud_subj is called as 'inner structure.' The members which are inside the inner structure can be accessed as follow :

```
result.subj.subjnm
result.subj.marks
```

Program :

```
/*  Program to demonstrate nested structures.

Creation Date : 23 Nov 2010 04:04:01 AM

Author : www.technoexam.com [Technowell, Sangli] */

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

```
struct stud_Res
{
          int rno;
          char std[10];
          struct stud_Marks
          {
                    char subj_nm[30];
                    int subj_mark;
          }marks;
}result;

void main()
{
          clrscr();
          printf("\n\t Enter Roll Number : ");
          scanf("%d",&result.rno);
          printf("\n\t Enter Standard : ");
          scanf("%s",result.std);
          printf("\n\t Enter Subject Code : ");
          scanf("%s",result.marks.subj_nm);
          printf("\n\t Enter Marks : ");
          scanf("%d",&result.marks.subj_mark);
          printf("\n\n\t Roll Number : %d",result.rno);
          printf("\n\n\t Standard : %s",result.std);
          printf("\nSubject Code : %s",result.marks.subj_nm);
          printf("\n\n\t Marks : %d",result.marks.subj_mark);
          getch();
}
```

Output :

```
Enter Roll Number : 1

Enter Standard : MCA(Sci)-I

Enter Subject Code : SUB001

Enter Marks : 63


Roll Number : 1

Standard : MCA(Sci)-I
       Subject Code : SUB001

Marks : 63_
```

Union :

Union is user defined data type used to stored data under unique variable name at single memory location.

Union is similar to that of stucture. Syntax of union is similar to stucture. 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.

To declare union data type, 'union' keyword is used.

Union holds value for one data type which requires larger storage among their members.

Syntax:

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

Example:

```
union techno
{
        int comp_id;
        char nm;
        float sal;
}tch;
```

In above example, it declares tch variable of type union. The union contains three members as data type of int, char, float. We can use only one of them at a time.

* Memory Allocation :

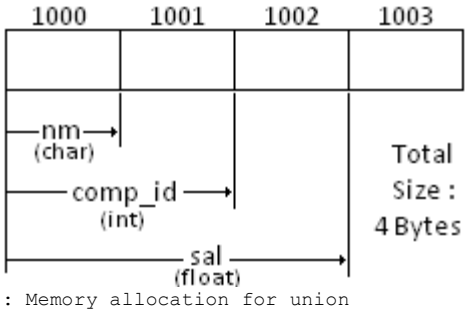

Fig : Memory allocation for union

To access union members, we can use the following syntax.

```
tch.comp_id
tch.nm
tch.sal
```

Program :

```
/*  Program to demonstrate union.

Creation Date : 10 Nov 2010 09:24:09 PM

Author :www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>

union techno
{
          int id;
          char nm[50];
}tch;


void main()
{
          clrscr();
          printf("\n\t Enter developer id : ");
          scanf("%d", &tch.id);
          printf("\n\n\t Enter developer name : ");
          scanf("%s", tch.nm);
          printf("\n\n Developer ID : %d", tch.id);//Garbage
          printf("\n\n Developed By : %s", tch.nm);
          getch();
}
```

Output :

```
        Enter developer id : 101

        Enter developer name : technowell

Developer ID : 25972

Developed By : technowell_
```