STRUCTURE

Structure is a group of variables of different data type represented by a single name. When a structure is created no memory space is allocated memory is allocated when variable is defined. The elements in a structure are stored in a continues memory locations

Define structure

Struct keyword is used

Syntax:-

```
struct struct_name
{
DataType member1_name;
DataType member2_name;
…………..
};
```

in structure the final brace should follow a semicolon.

declare a variable of structure

syntax: -

```
struct struct_name var_namr;

OR

struct struct_name
{
DataType member1_name;
DataType member2_name;
…………..
}var_name;
```

Data members of Structure

It can be accessed in two ways

Member operator/dot operator(.)
structure pointer operator(->)

Syntax: -

```
var_name.member_name;
```

Nested structure

In this the data entered will move accordingly in a order in which according to the data's entered.

```
#include<stdio.h>

void main()

{

  struct student
```

```
   {
      char name[20];
      int age;
   };
   struct prnt
   {
      char father[20];
      struct student a1;
   };
   struct prnt s1={"abc","xyz",123};
   printf("father_name %s \n",s1.father);
   printf("age %d",s1.a1.age);
}
```

OUTPUT

```
father_name abc
age 123
```

Array of structures

It is defined as the collection of multiple structures variables where each variable contain information about different entities.

```
#include<stdio.h>
void main()
{
   int i,j;
   struct std
   {
      int rll;
      char name[30];
   };
   printf("how many students   ");
   scanf("%d",&i);
   struct std s[i];
   for(j=0;j<i;j++)
   {
       printf("enter name ");
       scanf("%s",&s[j].name);
       printf("roll no ");
       scanf("%d",&s[j].rll);
   }
   printf("\n");
   printf("std information \n");
   for(j=0;j<i;j++)
   {
       printf(" name -> %s    roll no.-> %d \n",s[j].name,s[j].rll);
   }

}
```

OUTPUT

```
how many students   2
enter name abc
roll no 3
enter name ade
roll no 4

std information
 name -> abc    roll no.-> 3
 name -> ade    roll no.-> 4
```

passing of structure to function

```
#include<stdio.h>
   struct std
   {
       int rll;
```

```
        char name[30];
    };
void display(struct std s)
{
        printf(" name -> %s    roll no.-> %d \n",s.name,s.rll);


}
void main()
{
    int i,j;
    printf("how many students   ");
    scanf("%d",&i);
    struct std s[i];
    for(j=0;j<i;j++)
    {
        printf("enter name ");
        scanf("%s",&s[j].name);
        printf("roll no ");
        scanf("%d",&s[j].rll);
    }
    printf("std information \n");
    for(j=0;j<i;j++)
    {
    display(s[j]);
    }
}
```

OUTPUT

```
how many students   2
enter name abc
roll no 3
enter name cde
```

```
roll no 5
std information
 name -> abc    roll no.-> 3
 name -> cde    roll no.-> 5
```

Return structure from function

Here a structure function definition is made to get the information of the students

```
#include<stdio.h>
Struct student
{
Char name[20];
Int rll;
};
Struct student getinfo()
{
Struct student s1;
Printf("enter name");
Scanf("%s",&s1.name);
Printf("enter roll no.");
Scanf("%d",&s1.rll);
Return s1;
}
Int main()
{
Struct student s;
S=getinfo();
Printf("display details \n");
Printf("name %s             roll no. %d",s.name,s.rll);
Return 0;
}
```

Typedef usage

It is used to bring simplicity to a programme by using simplifying the syntax of declaring variables

```
 Typedef struct distance {
Int feet;
Float inch;
}distance;
Int main(){
Distance d1,d2;
}
```

Passing struct by difference

```
#include<stdio.h>
typedef struct multi
{
   int a,b;
   float c;
}var;
void m2(var v1,var v2,var *re)
{
   re->a = v1.a+v1.b;
   re->b = v2.a + v2.b;
}
void main()
{
   var v1,v2,re;
   printf("\n first serires \n");
   printf("enter 1 number");
   scanf("%d",&v1.a);
   printf("enter 2 number");
   scanf("%d",&v1.b);
   printf("\n second serires \n");
   printf("enter 1 number");
   scanf("%d",&v2.a);
```

```
    printf("enter 2 number");
    scanf("%d",&v2.b);
    m2(v1,v2,&re);
    printf("sum of first series  %d",re.a);
    printf("sum of second series  %d",re.b);
}
```

OUTPUT

```
first serires
enter 1 number2
enter 2 number3

 second serires
enter 1 number4
enter 2 number5
sum of first series  5
sum of second series  9
```

here the result when changed in m2 is also reflected by which the addredd is passed to m2 and re passed by reference

uses of structure

- widely used in Database Management
- changing the cursor size
- placing the cursor at an appropriate position in screen
- formatting floppy
- Hiding a file from directory
- Displaying the directory of disks
- Finding out the list of equipment attached to the computer
- Interacting with the mouse
- Clearing the contents of the screen
- Receiving a key from the keyboard

UNION

Union is a user defined data type which contains variables of different data types. In union it share same storage location separate storage location is not provided like that of structure. The size is allocated based on the largest size union member.

Syntax: -

Union name_of_union

{

Member1;

Member2;

…….membern;

};

Declaring union variable

Union union_name var1,var2;

Union name_of_union

{

Member1;

Member2;

…….membern;

}var_name1,var_name2;

Members can be accessed using

Var_name->member

Difference between union and structure

Union	Structure
Struct keyword is used to define a structure	Union keyword is sued to define a union
Members do not share memory in structure	Members share the memory space in a union
Any member cab be retrieved at any time in a structure	Only one member can be accessed at a time in a union
Several members of a structure can be initialized at once	Only the first member can be initialized

Size of the structure is equal to the sum of size of each number	size of the union is equal to the size of the largest number

Important Questions

1. **Advantages of structure.**
    - Members do not share memory in structure
    - Any member cab be retrieved at any time in a structure
    - Several members of a structure can be initialized at once
    - Size of the structure is equal to the sum of size of each number
2. **Syntax for structure.**
   Syntax:-
```
        struct struct_name
                {
                        DataType member1_name;
                                DataType member2_name;
                                        …………..
                                                };
```
3. **3 ways of declaring a structure**
    - struct struct_name var_name;
    - struct struct_name
```
        {
        DataType member1_name;
        DataType member2_name;
        …………..
        }var_name;
```
    - typedef struct struct_name
```
                {
                DataType member1_name;
                DataType member2_name;
                …………..
                }var_name
                main(){ Var_name var1,var2;}
```

4. **3 ways of initializing a structure**
    - Dot initialized -> var_name.member;
    - Value initialized -> struct struct_name var_name{member details separated by,};It should be in the order of which we declared in structure if the order is wrong complier will put in error
    - Variant of value initialization -> struct struct_name var_name { .member1=data, .member2=data, member=data };
5. **Difference between structure and union.**

Union	Structure

Struct keyword is used to define a structure	Union keyword is sued to define a union
Members do not share memory in structure	Members share the memory space in a union
Any member cab be retrieved at any time in a structure	Only one member can be accessed at a time in a union
Several members of a structure can be initialized at once	Only the first member can be initialized
Size of the structure is equal to the sum of size of each number	size of the union is equal to the size of the largest number

6. **3 ways of declaring a union variable.**

   union union_name var_name1,var_name2;

   Union name_of_union

   {

   Member1;

   Member2;

   …….membern;

   }var_name1,var_name2;

7. **Define a structure employee with employee number, name and experience. Write a C program to print a list of all employees having more than 5 year's experience.**

   Github:- https://github.com/Amarjith-c-k/structure-union/blob/master/struct.c

```
//Structure programme to read employee details display them and to find employee with
more than 5 yrs of experiment
#include<stdio.h>
struct empl
{
   int eno,eyr;
   char name[20];
};
void display(struct empl e,int j)
{
   printf("details of %d employee \n",j+1);
   printf("employee name  %s\n",e.name);
   printf("employee number  %d\n",e.eno);
   printf("employee eyr of experience  %d\n",e.eyr);
   printf("\n\n");
}
void main()
{
int i,j;
printf("how many employees \n");
scanf("%d",&i);
struct empl e[i];
for(j=0;j<i;j++)
{
   printf("enter employee name  ");
   scanf("%s",&e[j].name);
   printf("enter employee number  ");
   scanf("%d",&e[j].eno);
   printf("Enter employee year of experience  ");
   scanf("%d",&e[j].eyr);
   printf("\n\n");
}
for(j=0;j<i;j++)
{
   display(e[j],j);
}
for(j=0;j<i;j++)
{
   if(e[j].eyr>5)
      printf("employee have more tham 5 year of experiment %s \nemployee I_D =
%d\n",e[j].name,e[j].eno);
      continue;
}
}
```

Output

```
how many employees
2
enter employee name  abc
enter employee number  231
Enter employee year of experience  4


enter employee name  xyz
enter employee number  245
Enter employee year of experience  6


details of 1 employee
employee name  abc
employee number  231
employee eyr of experience  4
```

```
details of 2 employee
employee name  xyz
employee number  245
employee eyr of experience  6


employee have more tham 5 year of experiment xyz
employee I_D = 245
```

8. **Write a C program to read the name roll number and marks (5 subjects) of 10 students using union and display the total marks**

   Github :- https://github.com/Amarjith-c-k/structure-union/blob/master/union.c

```c
#include<stdio.h>
union std
{
   int m[5],rol,ttl;
   char name[20];
};
void main()
{
   union std s[10];
   int i,j;
   for(i=0;i<10;i++)
   {
    printf("details of %d \n",i+1);
    printf("enter the name ");
    scanf("%s",&s[i].name);
    printf("enter roll no.  ");
    scanf("%d",&s[i].rol);
    printf("enter the marks of 5 subject \n");
    s[i].ttl=0;
    for(j=1;j<=5;j++)
    {
       scanf("%d",&s[i].m[j]);
       s[i].ttl=s[i].ttl+s[i].m[j];
    }
       printf("total mark of this student %d \n",s[i]);
   }

}
```

OUTPUT

```
details of 1
enter the name a
enter roll no.  1
enter the marks of 5 subject
1
2
3
4
5
total mark of this student 15

details of 2
enter the name b
enter roll no.  2
enter the marks of 5 subject
2
3
4
5
5
total mark of this student 19

details of 3
enter the name c
enter roll no.  3
enter the marks of 5 subject
1
2
4
5
6
total mark of this student 18

details of 4
enter the name d
enter roll no.  4
enter the marks of 5 subject
5
6
7
8
7
total mark of this student 33

details of 5
enter the name e
enter roll no.  5
enter the marks of 5 subject
5
6
7
```

8
9
total mark of this student 35

similarly up-to the 10th subject