Problem Solving With C

UE15CS151

Structures

Structure

A structure is a Derived/ Structured Data Type.A structure is a collection of related variables.It may contain variables, possibly of different types, grouped together under a single name / tag.

- Structures help organize complicated data.
- A structure must be defined prior to a structure variable being declared.
- Structure definitions include a tag, member elements, and a variable definition.

Structure Definition:

```
struct tag
{ Member1
Memebr 2
............
Member n;
};
```

Example **:**

```
structXYZ
{
int a;
float b;
char c;
};
```

In the above definition “XYZ” is the Tag
a,b,c are the structure members and “struct” is a key word

Structure definitions inform the compiler what the structure will look like.
Structure definition does **not** allocate memory

Structure variable Declaration :

structstructure_TagVariable_name;

example: struct XYZ var1;

Or

A Structure variable declaration can be done while defining a structure itself

Example**:**

```
structXYZ
{
int a;
float b;
char c;
} var2;          // Variable declaration done here
```

We are also allowed to create multiple Structure variable in one declaration Statement

Example : struct XYZ var1,var2,var3,var 4......... ;

or

```
structXYZ
{
int a;
float b;
char c;
} var1,var2,var3,var4 ........... ;
```

Memory allocation for the structure variable :

The memory allocated for a structure variable will be the summation of the sizes of the Types of its members

size of a structure varianle = sum of the sizes of all its member

This isn't GENERIC as Padding is added when ever requires

Example :structXYZ

```
{
int a;
float b;
} var1;
```

```
sizeof(var1) = (sizeof(a)+sizeof(b)
             =    4    +    4
             =    8 bytes + Padding
```

The size for the structure variable var1 will be 8 bytes [consideringint and float to be of 4bytes]. A concept of Padding comes into picture when

structurevaraiablevar 1

a	4
b	4

Example 2 :

```
structarry
{
int a[3];
char c[4];
float b;
} arry1;
```

Memory allocation for arry1

member name	Size
a[0]	4
a[1]	4
a[2]	4
c[0]	1
c[1]	1
c[2]	1
c[3]	1
b	4

```
sizeof(arry1) = sizeof(a) + sizeof(c) + sizeof(b)
              =    (4*3) + (1*4) + 4
              =    12 + 4 + 4
              =    20 + Padding (if required)
```

Assigning values to the Structure Members :

One cannot assign a value to the structure member during the structure definition

Example :
```
struct new
{
int a = 100;            // Error
float b = 2.5           //Error
};
```

1)
```
structXYZ
        {
        int a;
        float b;
        char c;
        } ;

struct XYZ  v1 = {100,25.5,’A’};
```

2)
```
structXYZ
        {
        int a;
        float b;
        char c;
        }  v1 = {100,25.5,’A’};
```

3) Using Dot operator
```
structXYZ
        {
        int a;
        float b;
        char c;
        }  v1 ,v2;

v1.a = 100;
v1.b = 25.5;
v1.c = ‘A’;

v2.a = 8976;
v2.b = 2005.5;
v2.c = ‘B’;
```

Accessing the structure members:

In the previous topic we have done assigning of value to a particular structure member.

The retrieval of these assigned values is done by the using the same DOT operator

Example :

```
structXYZ
        {
        int a;
        float b;
        char c;
        }  v1 = {100,25.5,’A’};

printf(“Integer is %d\n”,v1.a);         // accessing ‘a’

printf(“Float is %f\n”,v1.b);           //accessing ‘b’

printf(“Char is %c\n”,v1.c);            //accessing ‘c’

        v1.a = 5000;

printf(“New Integer is %d\n”,v1.a);          // accessing ‘a’
```

Output :

```
Integer is 100
Float is 25.5
Char is A
New Integer is 5000
```

NESTED STRUCTURE:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
	system("clear");
	struct emp_contact_details
	{
		int door_no;
		char street[20];
		char city[20];
		char mob_no[10];
	}ecd;

	struct employee
	{
		int emp_id;
		char name[30];
		double salary;
		struct emp_contact_details ecd;			// Nested structure
	};


	struct employee emp1 = {1234,"ABC",45000,{292,"chruch
street","Bangalore","9986403"}};
	printf("EMPLOYEE DETAILS\n");
	printf("%s\n%s\n", emp1.name,emp1.ecd.mob_no);
}

#include<stdio.h>
#include<stdlib.h>
int main()
{
	system("clear");
	struct employee
	{
		int emp_id;
		char name[30];
		double salary;
		struct emp_contact_details			// Nested structure
		{
			int door_no;
			char street[20];
```

```
                        char city[20];
                        char mob_no[10];
                }ecd;
        };


        struct employee emp1 = {1234,"ABC",45000,{292,"chruch
street","Bangalore","9986403728"}};
        struct employee *ptr;
        ptr = &emp1;
        printf("EMPLOYEE DETAILS\n");
        printf("%s\n%s\n", ptr->name,ptr->ecd.mob_no);
}
```