UNIT NO _____ (STRUCTURE & UNION)

Array is the collection of same type of data element but in real life we need to group together different type of data (elements), which are related to each other logically. For this purpose in C language we use the concept of structure.

For example - ear, eyes, hair, mouth are different thing but they relate to one person.

Imp

It is a collection of different type of data element which are related to each other logically. It is the collection of heterogineous data which are covered under a single name is called structure.

Syntax → struct structure name
{
data type member 1;
datatype member 2;
};

Example → struct st
{
int roll;
char name[10];
};

struct is a keyword which tells the compiler that structure is being define & member 1, member 2 are declare inside the curly brasis & should be semicolumn at the end of curly brasis.

Members of the structure can be any data type like integer, character, pointer, float.
Here, structure template doesn't reserve or cover any space in the memory for the members They take the space when the actual variable of the structure declare template can be define globaly & locally.

Declaring Structure Variable :-

The structure variable can be declare by two ways.
i) With structure definition
ii) Using the structure tag

i) **With Structure Definition :-** Here, actual variable (objects) are declared at the time of defining the structure.

Syntax →
```
struct structure
{
data type member1;
data type member2;
} structure variable1, v2;
```

Example →
```
struct std
{
int roll;
char name[10];
} st1, st2, st3[4];
```

ii) **Using Structure Variable Tag :-** Here, variable can also be declared using structure tag.

Syntax → struct structure
{
data type member1;
data type member2;
};
struct structure name variable name;

Example → struct std
{
int roll;
char name[10];
};
struct std s1, s2;

Initialization Of Structure Variable :-

The initialisation of structure variable (gives the value to the member of the object) is same as simple variable.

```
struct st
{
int roll;
char name[10];
} st1 = {1, "Arif"};
struct st st2 = {2, "Vijay"};
struct st st3;
st3.roll = 3;
st3.name = "Abu";
```

WAP to display the value of structure member.

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

```
struct stu
{
int roll;
char name[30];
} st1, st2;
void main()
{
st1.roll=1;
st2.name="Anil";
scanf("%d%s", &st2.roll, &st2.name);
printf("%d%s", st1.roll, st2.roll, st1.name,
                                st2.name);
getch();
}
```

Assignment of structure variable :-

We can assign the value of one structure variable to another structure variable variable using the assignment ~~opp~~ operator.

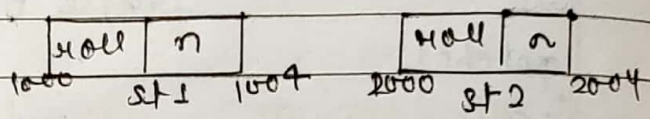
```
#include<stdio.h>
#include<conio.h>
struct stu
{
int roll;
char name[30];
} st1, st2;
void main()
{
st1.roll=1;
st1.name="Anil";
```

st2 = st1;
printf ("%d %d %s %s", st1.roll, st2.roll, st1.name, st2.name);
getch();
}

Storage Of Structure In Memory :-

Member of structures are store in sequential memory location or continuous memory location.

```
struct student
{
int roll;
char name[2];
} st1, st2;
```



Array with structure :- The structure can work with array by two way.

1) **Array within Structure :-** We can use the structure member as an memory array or array can be use as an member of structure

Example

```
struct st
{
int roll;
char name[10];
} st1;
```

ii) **Array Of Structure :-** Here, array variable can be declare as array of structure.

Example

```
struct st
{
int roll;
char name [10];
};
struct st stu[4];
```

When we declared the variables using array then it takes the space in memory location continuously. And when we declared the members of structures using array then it takes the different space in memory location.

Nested Structure (structure in structure) :-

When we place a structure within the another structure it is called nested structed.

Syntax →

```
structure tag1
{
member 1;
member 2;
struct tag2
};
};
```

Example →

```
struct employee
{
char name [10];
```

```
struct time
{
int jath date;
char month[4];
}t1;
};
```

Union :-

Union is a derived data type like Structure and it also contain members of different data type.

This Syntax used for Union, definition, declaration of Union variable and accessing member is similar to Structure but here we used the keyword Union in the place of Struct.

The main difference in Union and Structure is the Main memory is allocated for the members in Structure each members has on memory location where is the member is share memory location.

When a variable of type Union is declared compiler allocate sufficient memory to whole the largest member in Union and since all the members share the largest space one by one.

Syntax :- Same as Struct Syntax :-

```
Union [Union - name]
{
    data type member 1;
    data type member 2;
}
```

Difference between Union & Structure :-

	Union	Structure
1.	It is slow	It is fast
2.	It gives the memory to it's largest member	It gives the memory to it's all members.
3.	It save the memory	Don't save the memory.
4.	For declare the Union, we use the keyword union.	For declare structure to use the keyword struct

5.	Union Union_example
	{
	int integer;
	float decimal;
	char name[20];
	};
		struct Struct_example
		{
		int integer;
		float decimal;
		char name[20];
		};

Date Page 27 Feb 20

Structure - Programs

Que. WAP in a structure to declare a structure name student with member Roll no in integer & name in character, declare 5 object of this structure, initialise the value of this object & print them.

```
# include <stdio.h>
# include <conio.h>
  Struct student —— Structure template 0 byte
  {
    int roll; —— 2 byte
    char name [5]; —— 5 byte
  } st.1, st.2;
  Void main ()
  {
Struct student st3, st4, st5;
                                  Total 35 byte
    st1.roll = 1;
    st1.name = "Anil";
st2 = { 2, "Ashish" }
Scanf ("%d %d %d %s %s %s", &st3.roll,
       &st4.roll, &st5.roll, &st3.name,
       st4.name, &st5.name);
  }
  {
Printf ("%d %s", st.1 roll, st.1 name);
    getch ();
  }
```

Que. WAP to Declare a Structure name St. its member are Id in integer & marks in float declare two object of this Structure initialise the value to this object & print them.

```
#include <Stdio.h>
#include <Conio.h>
  Struct Student
{
    int  id ;
    float marks;

} St.1 , St.2 ;
  Void main ()
{
```

```
St1.id = 10;
St1.float = 2.5;

St 2 = { 3.5 ,  1.5. }
*Scanf (" %d %f %d %f ", & St.1 id, &
   St.1 marks, & St.2 id , & St.2 marks );

}
{x
Printf ("%d %f , St.1 id , St.1 marks);
  printf("%d %f, st.2 id, st.2 marks);
  getch ();
   }
```