CHAPTER NO: 7	STRUCTURE AND UNION
1	What is structure? Explain its syntax with suitable example.
2	Explain how the members of structure can be accessed?
3	How the size of structure can be determined in 'C' program?
4	Difference between (.) dot and (->)arrow operator.
5	What is nested structure? Explain with example.
6	Explain pointer to structure and use of arrow operator with example.
7	What is an array of pointer to structure?
8	What is union?
9	Difference between union and structure.
10	Explain array of structure with example.

Q-1 WHAT IS STRUCTURE? EXPLAIN ITS SYNTAX WITH SUITABLE EXAMPLE.

ANS:

- Structure is a collection of elements of either similar or dissimilar data
- type that share a common name.
- A various elements of a structure are known as structure members.
- Structure is a user defined data type.

Syntax:

Keyword **structure tag name**

```
struct structure-tag
{
        datatype variable1;
        datatype variable2;
        datatype variable3;
};
```

Terminating semicolon

Example :

```
struct book_bank
{
        char title[20];
        char author[15];
        int pages;
        float price;
};
```

- Here struct is a keyword that declares a structure that contains four members title, author, pages and price.
- This four members are known as structure elements (structure members).
- book_bank is the name of structure.

Q-2 EXPLAIN HOW THE MEMBERS OF STRUCTURE CAN BE ACCESSED?

ANS:

- The members of the structure cannot be accessed directly as other variables, as they are associated with a structure.
- The structure elements or member variables associated with the structure are accessed using the structure member operator (.) also called the dot operator which is used between the structure name and the member name.
- The members of the structure are accessed by two methods:
    1. Accessing the values using member operator (.)
    2. Accessing the values using scanf statement.

1. Accessing the values using member operator (.):-

Syntax:

```
Structure-variable . member-name=value;
```

dot operator

Example:

```
Emp01.salary=10000;
```

2. Accessing the values using scanf statement:

Syntax:-

```
scanf(“control string”,&structure-variable . member-name);
```

Example:-

```
scanf(“%f”,&emp01.salary);
```

Q-3HOW THE SIZE OF STRUCTURE CAN BE DEFINED IN ‘C’ PROGRAM?

ANS:

- The size of structure in C language depends upon the elements of the structure.
- The size of structure is the sum of length of its elements.

Example:

```
#include<stdio.h>
#include<conio.h>
struct student
{
        int roll;
        char name[10];
};
void main()
{
        printf(“\n size of structure=%d”,sizeof(struct student));
}
```

Output:

```
        size of structure=12
```

Q-4DIFFERENCE BETWEEN DOT (.) AND ARROW (->) OPERATOR.

ANS:

Dot (.) operator	Arrow(->) operator
1) Dot (.) operator is used to access the members of the sturcutre.	**1) Arrow (->) operator is used to access the members of the structure using pointer.**

Q-5 WHAT IS NESTED STRUCTURE? EXPLAIN WITH EXAMPLE.

ANS:

- A structure can also include another structure as its member in its definition.
- A structure defined within another structure is known as a nested structure.

Example :

```
struct salary
{
        char name;
        char department;
        int basic_pay;
        int dearness_allowance;
        int house_rent_allowance;
        int city_allowance;
}employee;
```

- We can group all the items related to allowance together and declare them under a substructure as follow:

```
struct salary
{
        char name;
        char department;
                struct
                {
                        int dearness;
                        int house_rent;
                        int city;
                }allowance;
}employee;
```

- The salary structure contains a member named allowance, which itself is a structure with three members.
- The members contained in the inner structure namely dearness, house_rent and city can be followed:

```
employee.allowance.dearness
employee.allowance.house_rent
employee.allowance.city
```

Q-6 EXPLAIN POINTER TO STRUCTURE AND ARROW OPERATOR WITH EXAMPLE.

ANS:

- Structure is the data structure which consists of group of elements that may or may not have same data type.
- firstly defining the structure

```
struct student
{
        int roll;
        char name[20];
        char lname[20];
};
struct student s1;
```

- Defining the pointer to structure

```
struct student *p;
```

- It is already known that pointer must be initialized before it is used.
- pointer to the structure can also be allocated memory dynamically.

```
p=(struct student *)malloc(sizeof(struct student));
```

Example:

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

```
struct student
{
        int roll;
        char name[20];
        char lname[20];
}
struct student s1;

void main()
{
        struct student *p;
        p=&s1;
        clrscr();
        s1.roll=1;
        strcpy(s1.name,”karishma”);
        strcpy(s1.lname,”kotecha”);
        printf(“\n rollno=%d”,p->roll);
        printf(“\n name=%s”,p->name);
        printf(“\n lname=%s”,p->lname);
        getch();
}
```

Q-7 WHAT IS AN ARRAY OF POINTER TO STRUCTURE?

ANS:

Q-8 WHAT IS UNION?

ANS:

- Union is a collection of elements of either similar or dissimilar data type that share a common name.
- Union is also user defined data type like structure.
- Union occupy less memory than structure because each members of union shard the common storage space (memory)
- So Union can handle only one member at a time.
- union keyword is used to declared the union.

Syntax:

keyword union tag name

```
union union-tag
{
        datatype variable1;
        datatype variable2;
        datatype variable3;
};
```

terminating semicolon

Example:-

```
union bar
{
        Char c;
        Float x;
}b1;
```

Q-9 DIFFERENCE BETWEEN UNION AND STRUCTURE.

ANS:

UNION	STRUCTURE
1) Union keyword is used to declare union.	1) structure keyword is used to declare structure.
1) All the members in union shares common storage space	2) Each member in structure has its own storage space.
2) Only one member can be accessed at a time.	3) All the members of structure can be accessed at any time.
3) Total required size is the largest one from memory occupied by all members.	4) Total required size is the addition of memory occupied by individual members.

4) Example :
union student
{
int rollno;
char name[20];
char city[20];
float weight;
} ;
Total size required= max(2,20,20,4)=20 bytes
	5) Example :
	structure student
	{
	int rollno;
	char name[20];
	char city[20];
	float weight;
	} ;
	Total size required= 2+20+20+4=46 bytes

5) It requires less memory than structure.	6) It requires more memory than union.

Q-10 EXPLAIN ARRAY OF STRUCTURE WITH EXAMPLE.

ANS:

- Array of structure means creating the array of the type struct.
  Example: struct student s[5];

- If we want to access the members of the structure, it can be done in following way:-

s[0].rollno	s[1].rollno
s[0].marks[0]	s[1].marks[0]
s[0].marks[1]	s[1].marks[1]
.	.
.	.
.	.
.	.
s[0].marks[4]	s[1].marks[4]