1 What is structure? How to declare a Structure? Explain with Example

- Structure is a collection of logically related data items of different data types grouped together under a single name.
- Structure is a user defined data type.
- Structure helps to organize complex data in a more meaningful way.

Syntax of Structure:

```
struct   structure_name
{
     data_type member1;
     data_type member2;

};
```

- **struct** is a keyword.
- **structure_name** is a tag name of a structure.
- member1, member2 are members of structure.

Example:

```
#include<stdio.h>
#include<conio.h>
struct book
{
     char title[100];
     char author[50];
     int pages;
     float price;
};
void main()
{
     struct book book1;
     printf("enter title, author name, pages and price of
book");
     scanf(“%s”,book1.title);
     scanf(“%s”, book1.author);
     scanf("%d",&book1.pages);
     scanf("%f",&book1.price);
     printf("\n detail of the book");
     printf(“%s”,book1.title);
     printf(“%s”,book1.author);
     printf("%d",book1.pages);
     printf("%f",book1.price);
     getch();
}
```

- book is structure whose members are title, author, pages and price.
- book1 is a structure variable.

2 How do we declare and access structure variables?

Declaration of structure:

- A structure variable declaration is similar to the declaration of variables of any other data type. It includes the following elements:
  1) The keyword **struct.**
  2) The structure tag name.
  3) List of variable names separated by commas.
  4) A terminating semicolon.

Example:

```
struct book
{
     char title[100];
     char author[50];
     int pages;
     float price;
} book1;
struct  book book2;
```

We can declare structure variable in two ways:

1) Just after the structure body like book1.
2) With struct keyword and structure tag name like book2.

Accessing structure members:

The following syntax is used to access the member of structure.

structure_variable.member_name

- structure_variable is a variable of structure and member_name is the name of variable which is a member of a structure.
- The "."(dot) operator or 'period operator' connects the member name to structure name.
- Example:

  book1.price represents price of book1.
- We can assign values to the member of the structure variable book1 as below,

```
strcpy(book1.title,"ANSI C");
strcpy(book1.author,"Balagurusamy");
book1.pages=250;
book1.price=120.50;
```

- We can also use scanf function to assign value through a keyboard.

```
scanf("%s",book1.title);
     scanf("%d",&book1.pages);
```

3 What is Union?

- Union is user defined data type just like structure.
- Each member in structure is assigned its own unique storage area where as in Union, all the members share common storage area.
- All members share the common area so only one member can be active at a time.
- Unions are used when all the members are not assigned value at the same time.

Example:

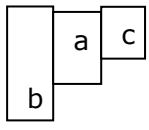
```
union book
{
      char title[100];
      char author[50];
      int  pages;
      float  price;
};
```

4 Difference between Structure and Array

Array	Structure
o An array behave like a built-in datatype all we have to do is to declare an array variable and use it.	o First we have to design and declare a data structure before the variables of that type are declared and use.
o An array is a collection of related data element of same type.	o Structure can have elements of different type.
o An array is derived datatype.	o Structure is programmer defined.

5 Difference between Structure and Union

Structure	Union
o Each member is assigned its own unique storage area.	o All members share the same storage area.
o Total memory required by all members is allocated.	o Maximum memory required by the member is allocated.
o All members are active at a time.	o Only one member is active a time.
o All members can be initialized.	o Only the first member can be initialized.
o Requires more memory.	o Requires less memory.

Example:
`struct SS`
`{`
`  int a;`
`  float b;`
`  char c;`
`};`
1 byte for c
2 bytes for a
4 bytes for b
	Example:
	`union UU`
	`{`
	`  int a;`
	`  float b;`
	`  char c;`
	`};`
	4 bytes for c,b,a
	 

o Total bytes = 1 + 2 + 4 = 7 bytes.	o 4 bytes are there between a,b and c because largest memory occupies by float which is 4 bytes.

6 Explain nested structure with example.

- A structure that contains another structure as a member variable is known as nested structure or structure within a structure.
- Structure which is part of other structure must be declared before the structure in which it is used.

Example:

```
#include<stdio.h>
#include<conio.h>
struct  address
{
     char add1[50];
     char add2[50];
     char city[25];
};
struct employee
{
     char name[100];
     struct address a;
     int salary;
};
void main()
{
     struct employee e;
     printf("enter name,address,city,salary");
     scanf(“%s”,e.name);
     scanf(“%s”,e.a.add1);
     scanf(“%s”,e.a.add2);
     scanf(“%s”,e.a.city);
     scanf("%d",&e.salary);
     printf("detail of employaee:");
     printf("%s”,e.name);
     printf("%s”,e.a.add1);
     printf("%s”,e.a.add2);
     printf("%s”,e.a.city);
     printf("%d",e.salary);
     getch();
}
```

7 Explain Array of Structure with Example?

- Array of structure mean collection of structures.
- Array storing different type of structure of variables.
- As we have an array of basic data types, same way we can have an array variable of structure.

Following example shows how an array of structure can be used,

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

```
{
     char name[100];
     int rollno;
     float cpi;
};
void main()
{
     struct result  r[66];
     int  i;
     printf("enter detail of student :");

     for(i=0;i<66;i++)
     {
          printf("\nenter name,roll no,cpi");
          scanf("%s",r[i].name);
          scanf("%d%f",&r[i].rollno,&r[i].cpi);
     }
     printf("\n detail of student:\n");
     for(i=0;i<66;i++)
     {
          printf("%s",r[i].name);
          printf("\t%d",r[i].rollno);
          printf("\t%f\n",r[i].cpi);
     }
     getch();
}
```

8 Explain Pointers to Structure with example.

- We can define pointers to structure in very similar way as you define pointer to any other variable.

Syntax:

```
struct  structure_name
{
     Member 1;
     Member 2;
     .
     .
};
 struct   structure_name   *structure_pointer;
```

Example:

```
struct  books
{
char  title[100];
 };
struct   books  *struct_pointer,book1;
```

- Now we can store the address of a structure variable in the above defined pointer variable.
- To find the address of a structure variable, place the '&' operator before the structure 's name

as follow:

- struct_pointer = &book1;
- To access the member of a structure using a pointer to that structure you must use the -> operator as follow:

  structure_pointer->title;

```
#include<stdio.h>
#include<conio.h>
struct  name
{
     int  a;
     float  b;
};
void  main()
{
     struct   name   *ptr, p;
     ptr = &p;                /* Referencing pointer to memory
     address of   p   */
     printf("Enter Integer:  ");
     scanf("%d",&(*ptr).a);
     printf("Enter number:  ");
     scanf("%f",&(*ptr).b);
     printf("Displaying:  ");
     printf("%d \n %f",(*ptr).a,(*ptr).b);
     getch();
}
```

- Structure pointer member can also be accessed using -> operator.

  (*ptr).a is same as **ptr-> a.**
  (*ptr).b is same as **ptr-> b.**

9 What is Typedef?

- **Typedef** is a keyword in the C language, it is used to define own identifiers that can be used in place of type specifiers such as int,float, and double.
- The names you define using typedef are not new data types, but synonyms for the data types or combinations of data types they represent.
- The name space for a typedef name is the same as other identifier.
- A typedef can be used to simplify the declaration for a structure.

  Example:-

```
typedef struct
{
     Char firstName[20];
     Char lastName[20];
     int no;
}student;
```

- Now we can use student directly to define variables of student type without using struct keyword. Following is the example:-

student student_a;

- It is also possible to use type definitions with structures. The name of the type definition of a structure is usually in uppercase letters.

Example:

```
#include<stdio.h>
typedef struct telephone
{
     char *name;
     int number;
}TELEPHONE;
int main()
{
     TELEPHONE indx;
     index.name="xyz";
     index.number=12345;
     printf("Name : %s\n", index.name);
     printf("Telephone number: %d\n",index.number);
     return 0;
}
```

10 What is Enumeration?

- Enumeration type allows programmer to define their own data type. Keyword **enum** is used to defined enumerated data type.
- **enum** type_name{ value1, value2, . . . , valueN };
- Here**, type_name** is the name of enumerated data type or tag. And value1,value2,….,valueN are values of type type_name.
- By default, value1 will be equal to 0, value2 will be 1 and so on but, the programmer can change the default value as below:

```
enum  suit
{
     club=0;
     diamonds=10;
     hearts=20;
         spades=3;
};
```

Declaration of enumerated variable:

- Above code defines the type of the data but, no any variable is created. Variable of type **enum** can be created as:

```
enum Boolean
{
     false;
     true;
};
enum  Boolean check;
```

- Here, a variable check is declared which is of type enum boolean.

Example:

```
#include<stdio.h>
enum  week
{
  sunday, moday, tuesday, wednesday, thursday, friday,
saturday
};
int main()
{
          enum week today;
          today=wednesday;
          printf("%d day",today+1);
          return 0;
}
```

Output:

4 day