

# Structures

- ◆ A structure is a collection of multiple variables with a single name.
- ◆ Variables in structure may be of same data type or may be of different types.
- ◆ Variables in structure are called **data items**, the data items in a structure are called **Structure Elements, members or Fields**.
- ◆ Structure is used to create user defined data type in C .
- ◆ Simple variable can store only one value at a time, however structure variable can store multiple values at the same time.

## Difference between Array and Structure

- ◆ The difference between array and structure is that array consists of elements of same type, however structure elements may consists of different types.

## DECLARING A STRUCTURE

- ❖ A Structure is declared by using the keyword **struct** followed by the structure name.
- ❖ The structure members are defined with their type inside the opening and closing braces { }
- ❖ The closing brace is ended with semicolon.
- ❖ **This declaration only tells the compiler about the details of structure, compiler does not allocate any memory.**

### Syntax:

```
struct structureName {  
    dataType member1;  
    dataType member2;  
    ...  
};
```

### Example:

```
struct Student {  
    int rno,marks;  
    float avg;  
    char grade;  
};
```

In above example structure contains four member variables; this declaration has created a new data type **Student**.

A variable of type **Student** can store four values at one time.

## DEFINING A STRUCTURE VARIABLE

- The structure variable can be **defined** after the declaration of a structure.
- The process of defining a structure variable is same as defining a variable of basic types such as int and char.
- **The definition tells the complier to allocate memory space for the variable.**
- **Compiler allocates memory according to the elements of the structure.**

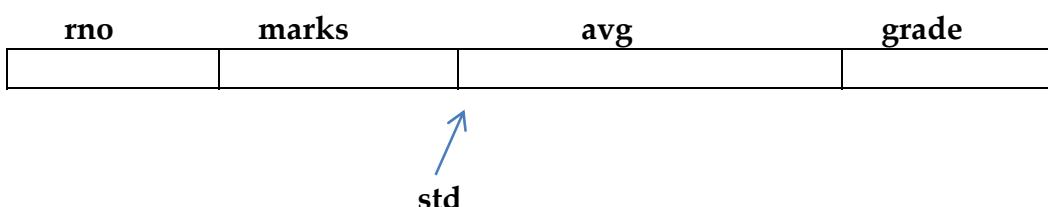
### Syntax:

```
structureName identifier;
```

### Example:

```
Student std;
```

### Memory representation:



How many bytes the variable **std** will occupy?

Ans:  $2+2+4+1 = 9$  bytes      (answer may vary for different types of compliers)

### **ACCESSING MEMBERS OF STRUCTURE VARIABLE**

- Any member of a structure variable can be accessed by using **dot operator**.
- Name of **structure variable** is written on left side, and **member variable** is written on the right side of dot.
- the dot operator is also known as **member access operator**

Syntax:

```
structureVar.MemberVariable;
```

Example:

```
Student std;
std.rno = 10;
std.marks = 62;
std.avg = 71.32;
std.grade = 'A';
```

#### ◆ Program:

```
/* write a program that declares a structure to store Roll no ,  
and grade of a student . the program should define a structure  
and then input the values and then display these values */  
  
#include<stdio.h>  
#include<conio.h>  
struct student  
{  
    int rno;  
    int marks;  
    float avg;  
    char g;  
};  
int main()  
{  
    student s; //declaring a variable s of type student  
    printf("Enter your Roll No:");  
    scanf("%d", &s.rno);  
    printf("Enter your Marks:");  
    scanf("%d", &s.marks);  
    printf("Enter your average Marks:");  
    scanf("%f", &s.avg);  
    printf("Enter your Grade:");  
    s.g=getche();  
  
    printf("\n-----\n");  
    printf("Your Roll Is :%d\n", s.rno);  
    printf("Your Marks are :%d\n", s.marks);  
    printf("Your Average Marks are :%f\n", s.avg);  
    printf("Your Grade is :%c\n", s.g);  
}
```

```

/* write a program that declares a structure to store (DOB) day,
it then input these values and then display DOB in dd/mm/yy for */

#include<stdio.h>
#include<conio.h>
struct birth
{
    int d;
    int m;
    int y;
};
int main()
{
    birth b; //declaring a variable b of type birth
    printf("Enter day of Your DOB:");
    scanf("%d",&b.d);
    printf("Enter Month of Your DOB:");
    scanf("%d",&b.m);
    printf("Enter year of your DOB:");
    scanf("%d",&b.y);

    printf("\n-----\n");
    printf("Your DOB in dd/mm/yy Is %d/%d/%d",b.d,b.m,b.y);
}

/* write a program that daclares a structure to store bookid,pa
then input data for two books and display the most costly book
*/
#include<stdio.h>
struct book
{
    int bid;
    int pages;
    int price;
};
int main()
{
    book b1,b2;
    printf("Enter data for FIRST BOOK\n");
    printf("Enter book-id of First BOOK:=");
    scanf("%d",&b1.bid);
    printf("Enter pages of First BOOK:=");
    scanf("%d",&b1.pages);
    printf("Enter price of First BOOK:=");
    scanf("%d",&b1.price);
    printf("-----\n");
    printf("Enter data for BOOK-2\n");
    printf("Enter book-id of 2nd BOOK:=");
    scanf("%d",&b2.bid);
    printf("Enter pages of 2nd BOOK:=");
    scanf("%d",&b2.pages);
    printf("Enter price of 2nd BOOK:=");
    scanf("%d",&b2.price);

    printf("-----\n");
    printf("DATA of Most Costly Book is \n");
    if(b1.price > b2.price)
    {
        printf("Book-id = %d\n",b1.bid);
        printf("Book pages = %d\n",b1.pages);
        printf("Book price = %d\n",b1.price);
    }
    else
    {
        printf("Book-id = %d\n",b2.bid);
        printf("Book pages = %d\n",b2.pages);
        printf("Book price = %d\n",b2.price);
    }
}

```

## Initializing Structure variables

- ◆ The process of assigning values to member variables of structure is called initialization of structure variables.
- ◆ The values are written in the same sequence in which they are specified in structure declaration.
- ◆ Each value is separated by comma.

Syntax:

```
structureName identifier = { value1, value2 , . . . . .};
```

Example:

```
Student std ={ 10,62,71.32,'A'};
```

```
/* write a program that daclares a structure to store  
employeeid and salary of an employee .  
It define and initilize a structure variable and display it
```

```
#include<stdio.h>  
struct Emp  
{  
    int eid;  
    int sal;  
};  
int main()  
{  
    Emp e={20,25000};  
  
    printf("Employee-id = %d\n",e.eid);  
    printf("Employee Salary = %d\n",e.sal);  
}
```

```
*****
```

```
/* write a program that daclares a structure Phone  
that store a national code,areacode and number separately  
Create Two variables of structure Phone.  
initilize one variable and input values for other  
then display the value of both variables#include<stdio.h>  
struct Phone  
{  
    int ncode;  
    int acode;  
    int num;  
};  
int main()  
{  
    Phone p1,p2={92,42,9232345};  
    printf("Enter National Code:=");  
    scanf("%d",&p1.ncode);  
    printf("Enter Area Code:=");  
    scanf("%d",&p1.acode);  
    printf("Enter Phone Number:=");  
    scanf("%d",&p1.num);  
  
    printf("-----\n");  
    printf("Phone Number 1: +%d-%d-%d\n",p1.ncode,p1.acode,p1.num);  
    printf("Phone Number 2: +%d-%d-%d\n",p2.ncode,p2.acode,p2.num);  
}
```

## Assigning One Structure Variable to Other

- ◆ A structure variable can be assigned to another structure variable by using assignment operator as follows:
- ◆ 

```
Student std = {1,786,89.52,'A'};  
Student std1 = std;
```

Example:

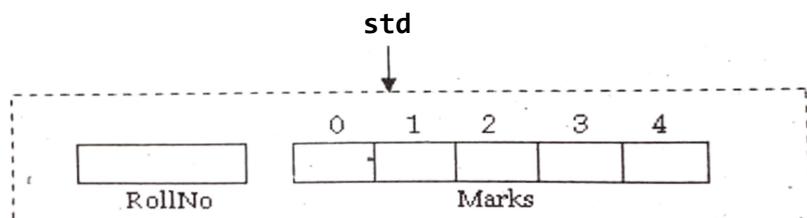
```
/* Write a program that declares a structure to store marks  
and grade of a student. It defines two structure variables.  
It inputs the values in one variable and assigns that variable  
to the second variable.  
It finally displays the values of both variables. */  
  
#include<stdio.h>  
#include<conio.h>  
struct Marks  
{  
    int m;  
    char g;  
};  
int main()  
{  
    Marks a,b;  
    printf("Enter Marks:");  
    scanf("%d",&a.m);  
    printf("Enter Grade :");  
    a.g=getche();  
    b=a;  
    printf("\n First record is as follows: \n");  
    printf(" Marks :%d\n",a.m);  
    printf(" Grade:%c\n",a.g);  
  
    printf("Second record is as follows: \n");  
    printf(" Marks :%d\n",b.m);  
    printf(" Grade:%c\n",b.g);  
}
```

## Array as Member of Structure

- ◆ A structure may consist of different types of data.
- ◆ The member variables can be simple data types such as int, char or float
- ◆ The member variables can also be complex like arrays.

Example:

```
struct Student  
{  
    int RollNo;  
    int Marks[5];  
}std;
```



The above example declares a structure `Student` with two members. The first member is a simple variable of type `int`. The second member is an array of integers. It can be used to store the marks of five subjects.

```
std.RollNo = 10;  
std.Marks[0] = 89;  
std.Marks[1] = 93;  
std.Marks[2] = 83;  
std.Marks[3] = 79;  
std.Marks[4] = 95;
```

We can also initialize a Structure with **Array as Member** like this:

```
Student std = {10,{89,93,83,79,95}};
```

*/\*Write a program that declares a structure to store roll no and marks of five subjects. It defines a structure variable, inputs, the values and displays roll no, marks and average marks \*/*

```
#include <stdio.h>

struct Test
{
    int rno;
    int m[5];
};

int main()
{
    Test r;
    int t = 0;
    float avg = 0.0;
    printf("Enter roll no: ");
    scanf("%d",&r.rno);
    for(int i=0;i<5;i++)
    {
        printf("Enter marks: ");
        scanf("%d",&r.m[i]);
        t = t + r.m[i];
    }
    avg = t / 5.0;
    printf("Roll No: %d\n",r.rno);
    printf("Total marks: %d\n",t);
    printf("Average: %f",avg);
}
```

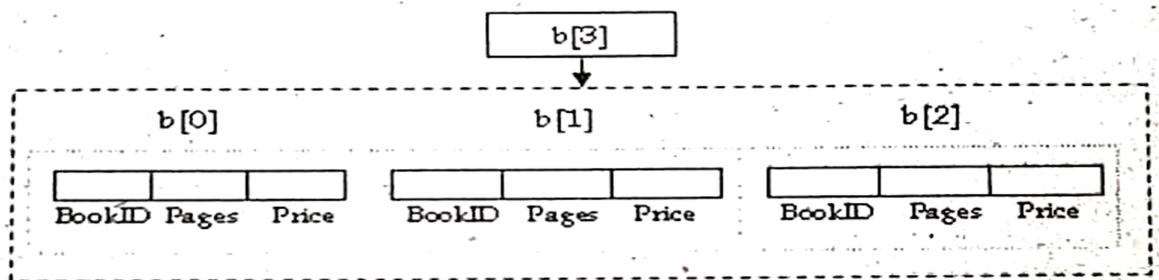
## Array of Structures

---

- ⌚ An array is a collection of same type of data.
- ⌚ An array can be of simple data type such as int, char or float etc.
- ⌚ Similarly, an array can be of user-defined type such as a structure.
- ⌚ An array of structure is a type of array in which each element contains a complete structure.
- ⌚ It can be used to store many records.

**Example:**

```
struct Book
{
    int bid;
    int pages;
    int price
};
Book b[3];
```



We can declare an array of structures and initialize it as follow:  
`Book b[3] = {{1,230,125,50},{2,480,185,75},{3,360,145,50}};`

```
/* Write a program that declares a structure to store id, pages and price of a book.  
It defines an array of structures to store the records of five books.  
It inputs the records of five books and displays the record of most costly book.  
*/
```

```
#include<stdio.h>  
struct book  
{  
    int bid;  
    int pages;  
    int price;  
};  
int main()  
{  
    book b[5];  
    int i,max,m;  
    for(i=0;i<5;i++)  
    {  
        printf("Enter book-id :");  
        scanf("%d",&b[i].bid);  
        printf("Enter pages :");  
        scanf("%d",&b[i].pages);  
        printf("Enter price:");  
        scanf("%d",&b[i].price);  
    }  
    max=b[0].price;  
    m=0;  
    for(i=1;i<5;i++)  
    if(b[i].price > max)  
    {  
        max=b[i].price;  
        m=i;  
    }  
    printf("Most costly Book is \n");  
    printf("Book-id = %d\n",b[m].bid);  
    printf("Book pages = %d\n",b[m].pages);  
    printf("Book price = %d\n",b[m].price);  
}
```

```
*****
```

```
/* Write a program that declares a structure Teacher to store Tid, Tname and salary.  
The program defines an array of Teacher structure to store the records of five  
teachers. It inputs the records of five teachers. The program inputs a Tid from user  
and search the teacher and show it's record. */
```

```
#include<stdio.h>  
struct Teacher  
{  
    int Tid;  
    char Tname[50];  
    int sal;  
};  
int main()  
{  
    Teacher t[5];  
    int id,i,n=-1;  
    for(i=0;i<5;i++)  
    {  
        printf("Enter Teacher-id :");  
        scanf("%d",&t[i].Tid);  
        printf("Enter Teacher Name :");  
        scanf("%s",&t[i].Tname);  
        printf("Enter Salary:");  
        scanf("%d",&t[i].sal);  
    }  
    printf("Enter Teacher-id to Search :");  
    scanf("%d",&id);  
    for(i=1;i<5;i++)  
    if(t[i].Tid == id)  
        n=i;  
  
    if (n== -1)  
        printf("Record Not Found");  
    else  
    {  
        printf("Teacher record is as follow \n");  
        printf("Teacher-id = %d\n",t[n].Tid);  
        printf("Teacher Name = %s\n",t[n].Tname);  
        printf("Teacher Salary = %d\n",t[n].sal);  
    }  
}
```

### output

```
Enter Teacher-id :1  
Enter Teacher Name :Faisal  
Enter Salary:15000  
Enter Teacher-id :2  
Enter Teacher Name :Afzal  
Enter Salary:20000  
Enter Teacher-id :3  
Enter Teacher Name :Haroon  
Enter Salary:10000  
Enter Teacher-id :4  
Enter Teacher Name :Akmal  
Enter Salary:17000  
Enter Teacher-id :5  
Enter Teacher Name :Akhtar  
Enter Salary:16000  
Enter Teacher-id to Search :4  
Teacher record is as follow  
Teacher-id = 4  
Teacher Name = Akmal  
Teacher Salary = 17000
```

# Nested Structure

- ◆ A structure within a structure is known as nested structure.
- ◆ A nested structure is created when the member of a structure is itself a structure.

## Example

```
struct A
{
    int n;
    float b;
};

struct B
{
    char c;
    A x;
};
```

## Accessing Members of Nested Structure

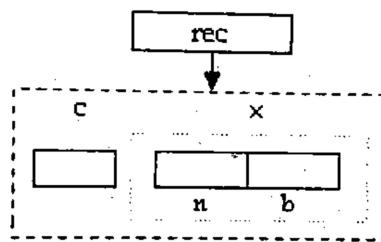
The member variable of nested structure can be accessed using multiple dot operators.

The first dot operator refers the member variable of outer structure. The second dot operator refers the inner structure and so on.

### Example:

```
B rec;
```

The above line creates a **structure variable rec of type B**. It is a nested structure that contains another structure variable x as its member.



The values can be stored in above nested structure like this:

```
rec.c = 'A';
rec.x.n = 10;
rec.x.b = 50.43
```

it can be initialized like this

```
B rec1 = { 'B' , {40,45.67} };
```

## Review Questions

### Q.1. Define a structure.

A structure is a collection of multiple variables with single name. It may contain similar or different data types. The data items in a structure are called structure elements, members or fields. The structures are used to join simple variables together to form complex variables.

### Q.2.. What is the main difference between array and structure?

The difference between an array and a structure is that array consists of a set of variables of same data type. However, a structure may consist of different data types.

### Q.3. What is difference between declaring and defining a structure? T

The declaration tells the compiler about the details of the structure. The compiler does not allocate any memory. The definition tells the compiler to allocate memory space for the variable. The compiler allocates memory according to the elements of structure.

### Q.4. How are Members of structure accessed?

Any member of a structure variable can be accessed by using **dot operator**. The name of the structure variable is written on the left side of the dot. The name of member variable is written on right side of the dot. The dot operator is also known as member access operator.

### Q.5. What is a nested structure?

A structure within a structure is known as nested structure. A nested structure is created when the member of a structure is itself a structure.

```
/* Write a program that uses two structures Result and Record. The  
Result structure stores marks and grade, Record structure stores roll  
number and a Result type. The program declares a variable of type  
Record, inputs roll number, marks and grade. It finally displays these  
values on the screen. */
```

```
#include<stdio.h>  
#include<conio.h>  
struct Result  
{  
    int marks;  
    char grade;  
};  
struct Record  
{  
    int rno;  
    Result r;  
};  
int main()  
{  
    Record rec;  
    printf("Enter Roll No :");  
    scanf("%d",&rec.rno);  
    printf("Enter Marks :");  
    scanf("%d",&rec.r.marks);  
    printf("Enter Grade:");  
    rec.r.grade=getche();  
  
    printf("\nStudent record is as follow : \n");  
    printf("Student Roll no = %d\n",rec.rno);  
    printf("Marks = %d\n",rec.r.marks);  
    printf("Grade = %c\n",rec.r.grade);  
}
```

```
/* Write a program that uses two structures date and phonebook. The  
date structure stores day, month and year. Phonebook structure stores  
name, city, telephone and a date type. The program declares a variable  
of type phonebook, inputs values and displays the values. */
```

```
#include<stdio.h>  
#include<conio.h>  
struct date  
{  
    int day;  
    int month;  
    int year;  
};  
struct phonebook  
{  
    char name[40];  
    char city[40];  
    int tel;  
    date birthday;  
};  
  
int main()  
{  
    struct phonebook pb;  
    printf("Enter Name :");  
    scanf("%s",&pb.name);  
    printf("Enter City :");  
    scanf("%s",&pb.city);  
    printf("Enter Phone No:");  
    scanf("%d",&pb.tel);  
    printf("Enter Date of Birth (dd-mm-yy):");  
    scanf("%d%d%d",&pb.birthday.day,&pb.birthday.month,&pb.birthday.year);  
  
    printf("Record is as follow : \n");  
    printf("Name is = %s\n",pb.name);  
    printf("City is = %s\n",pb.city);  
    printf("Phone is = %d\n",pb.tel);  
    printf("Birth date is =  
          %d-%d-%d\n",pb.birthday.day,pb.birthday.month,pb.birthday.year);  
}
```