

Structure

Structures are usually used when we want to store dissimilar data together. For example, we want to store data about a book. Book has its title, author name, number of pages and price. All of the book attributes belong to different data types. One way to store the data is to construct individual arrays, and another method is to use a structure variable. Structure elements are always stored in contiguous memory locations.

The general form of a structure declaration statement is given below:

```
struct <structure name> {  
    structure element 1;  
    structure element 2;  
    structure element 3;  
} struct_variable;  
//Or  
struct book  b1, b2, b3 ;
```

Before the final semicolon, we specify the structure variables but it is optional. We can also specify the structure variable in main body. The keyword **struct** should be used to define variables of structure type.

Following is the example of declaring struct in C:

```
struct Books{  
    char title[20];  
    char author_name[100];  
    float price;  
    int pages;} book1;
```

How to Access the Structure Elements?

As we use subscript to access individual elements of an array. But in the case of structures, to access any element, we use the **operator (.)**. This dot operator is coded between the structure variable name and the structure member that we wish to access.

Note that before the dot operator there must always be a structure variable and after the dot operator there must always be a structure element.

```

#include <stdio.h>
struct book {
char title[20];
char Author_name[100];
float price;
int pages;
} ;
int main( ) {
struct book book1 = {"C programming", "ABC", 130.00, 550};
printf ( "\n Title = %s", book1.title ) ;
printf ( "\n Name = %s", book1.Author_name ) ;
printf ( "\n Price = %.2f",book1.price ) ;
printf ( "\n Pages = %d", book1.pages ) ;
return 0;}

```

Additional Features of Structures :-

- We can assign the values of a structure variable to another structure variable of the same type using the assignment operator.
- Structure can be nested within another structure.
- We can pass the structure variable to a function. We can pass the individual structure elements or the entire structure variable into the function as an argument.
- We can have a pointer pointing to a struct just like the way we can have a pointer pointing to an int, or a pointer pointing to a char.

Where are structures useful?

Structures can be used for a variety of purposes like:

1. Structures are used to store a large amount of data.
2. They are used to send data to the printer.
3. For placing the cursor at an appropriate position on screen, we can use structure.
4. It can be used in drawing and floppy formatting.
5. We use structures in finding out the list of equipment attached to the computer.