Structures and Arrays

1. Structures Containing Arrays
2. Arrays of Structures

Structures Containing Arrays

- One can define a structure that contains one or more arrays as members.
- The array can be of any C data type (int, char, and so on).
- For example, consider the declaration below.

```
struct data
{
    int x[4];
    char y[10];
};
```

This statement defines a structure of type data that contains a four element integer array member named x and a 10 element character array member named y.

- One can then declare a structure named record of type data as follows:

```
struct data record;
```

- One can access individual elements of arrays that are structure members using a combination of the member operator and array subscripts:

```
record.x[2] = 100;
record.y[1] = 'x';
```

- The character arrays are most frequently used to store strings and the name of an array, without brackets, is a pointer to the array.
- Because this holds true for arrays that are structure members, the expression:

```
record.y
```

is a pointer to the first element of array y[ ] in the structure record.

- Therefore, one could print the contents of y[ ] onscreen using the statement:

```
puts(record.y);
printf("%s",record.y);
```

Example

```
struct data
{
    int x[4];
    char y[10];
};

void main()
{
    struct data record={1,2,3,4},"NIELIT";
    printf("x - %d %d %d %d\n", record.x[0],record.x[1],record.x[2],record.x[3]);
    printf("y - %s",record.y)
}
```

Output

```
x - 1 2 3 4
y - NIELIT
```
Arrays of Structures

- After the structure has been defined, one can declare an array as follows:
  ```
  struct entry list[1000];
  ```
  This statement declares an array named list that contains 1,000 elements.
- Each element is a structure of type entry and is identified by subscript like other array element types.
- Each of these structures has three elements, each of which is an array of type char.
- Structure is used to store the information of one particular object but if we need to store such 100 objects then Array of Structure is used.
- Book structure is used to store the information of one Book
  ```
  struct BookInfo
  {
    char[20] bname;
    int pages;
    int prince;
  }
  Book[100];
  ```
- In case if we need to store the information of 100 books then Array of Structure is used.
- b1[0] stores the information of 1st Book, b1[1] stores the information of 2nd Book and so on. We can store the information of 100 books.

Initializing Array of Structures

```
struct car
{
  char make[20];
  char model[30];
  int year;
};
struct car arr_car[2] = {
  {"Audi", "TT", 2016},
  {"Bentley", "Azure", 2002}
};
```
Example

```c
#include <stdio.h>
struct Bookinfo
{
    char[20] bname;
    int pages;
    int price;
};
book[3];
int main(int argc, char *argv[])
{
    int i;
    for(i=0;i<3;i++)
    {
        printf("\nEnter the Name of Book  : ");
        gets(book[i].bname);
        printf("\nEnter the Number of Pages : ");
        scanf("%d",book[i].pages);
        printf("\nEnter the Price of Book   : ");
        scanf("%f",book[i].price);
    }
    printf("\n--------- Book Details --------- ");
    for(i=0;i<3;i++)
    {
        printf("\nName of Book  : %s",book[i].bname);
        printf("\nNumber of Pages : %d",book[i].pages);
        printf("\nPrice of Book   : %f",book[i].price);
    }
    return 0;
}
```

**Output:**
Enter the Name of Book  : ABC
Enter the Number of Pages : 100
Enter the Price of Book   : 200
Enter the Name of Book  : EFG
Enter the Number of Pages : 200
Enter the Price of Book   : 300
Enter the Name of Book  : HIJ
Enter the Number of Pages : 300
Enter the Price of Book   : 500

--------- Book Details ---------

Name of Book  : ABC
Number of Pages : 100
Price of Book   : 200
Name of Book  : EFG
Number of Pages : 200
Price of Book   : 300
Name of Book  : HIJ
Number of Pages : 300
Price of Book   : 500
Example

```c
#include<stdio.h>
#include<string.h>

struct studentInfo
{
    char name[20];
    int roll_no;
    float marks;
};

void main()
{
    struct studentInfo st[5];
    int i;

    for(i = 0; i < 5; i++)
    {
        printf("Enter details of student %d\n\n", i+1);

        printf("Enter name: ");
        scanf("%s", st[i].name);

        printf("Enter roll no: ");
        scanf("%d", &st[i].roll_no);

        printf("Enter marks: ");
        scanf("%f", &st[i].marks);
    }

    printf("\n");

    printf("Name	Roll no	Marks\n");

    for(i = 0; i < 5; i++)
    {
        printf("%s\t%d\t%f\n", 
                st[i].name, st[i].roll_no, st[i].marks);
    }

    }
```