ARRAY OF STRUCTURES – Module 4

Module 4

Introduction

- Consider if we have a program to store data of 100 books, we would be required to use 100 different structure variables from **b1** to **b100**,(which is practically not possible)
- Better approach would be to use an array of structures.
- Look at the program at next slide

```
/* Usage of an array of structures */

struct book

{

        char name ;

        float price ;

        int pages ;

} ;

main( )

{

        struct book b[100] ;

        int i ;
```

```
for ( i = 0 ; i <= 99 ; i++ )

{

        printf ( "\nEnter name, price and pages " ) ;

        scanf ( " %c %f %d", &b[i].name,&b[i].price,
                                        &b[i].pages ) ;

}

for ( i = 0 ; i <= 99 ; i++ )

        printf ( "\n%c %f %d", b[i].name, b[i].price,
                                        b[i].pages ) ;

}
```

- This provides space in memory for 100 structures of the type **struct book**.
- we refer to zeroth book's price as **b[0].price**.Similarly, we refer first book's pages as **b[1].pages**.
- In an array of structures all elements of the array are stored in adjacent memory locations.

Uses of Structures

- The immediate application that comes to the mind is Database Management. That is, to maintain data about employees in an organization, books in a library, items in a store, etc.