CS19001: Programming and Data Structures Laboratory

Aritra Hazra;
CSE, IIT Kharagpur

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Autumn2019/

27-Oct-2019

Multi-dimensional Arrays

One-dimensional arrays are quite able to represent many natural collections. There are some other natural collections that may better be conceptualized as 2-dimensional data. Example: a matrix.

0	1	2	3
4	5	6	7
8	9	10	11

Two-Dimensional Arrays


```
int a[3][4] = {
 {0, 1, 2, 3} ,    /*for row index 0 */
 {4, 5, 6, 7} ,    /*for row index 1 */
 {8, 9, 10, 11}    /*for row index 2 */
};
```

or

```
int a[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};
```

Matrix addition

```
int m, n, c, d;
int first[9][9], second[9][9], sum[9][9];
printf("Enter no. of rows and columns\n");
scanf("%d%d", &m, &n);
printf("Enter the elements of matrix1\n");
for ( c = 0 ; c < m ; c++ )
  for ( d = 0 ; d < n ; d++ )
    scanf("%d", &first[c][d]);
/*now scan 2nd matrix*/

for ( c = 0 ; c < m ; c++ )
  for ( d = 0 ; d < n ; d++ )
    sum[c][d] = first[c][d] + second[c][d];
```

Structures

Structure is a user defined data type available in C, which allows you to combine data items of different kinds.

```
struct Books
{
    char  title[50];
    char  author[50];
    char  subject[100];
    int   book_id;
};
```

```
int main( )
{
    struct Books Book1;
    /* Declare Book1 of type Book */
    struct Books Book2;
    /* Declare Book2 of type Book */

    /* book 1 specification */
    strcpy( Book1.title, "C Programming");
    strcpy( Book1.author, "Nuha Ali");
    strcpy( Book1.subject, "C Programming");
    Book1.book_id = 6495407;

    /* book 2 specification */
    strcpy( Book2.title, "Telecom Billing");
    strcpy( Book2.author, "Foolan Barik");
    strcpy( Book2.subject, "Telecoms");
    Book2.book_id = 6495700;
```

```
/* print Book1 info */
   printf("Bk1 title %s\n", Book1.title);
   printf("Bk1 author %s\n", Book1.author);
   printf("Bk1 sub %s\n", Book1.subject);
   printf("Bk1 id %d\n", Book1.book_id);

   /* print Book2 info */
   printf("Bk2 title %s\n", Book2.title);
   printf("Bk2 author %s\n", Book2.author);
   printf("Bk2 sub %s\n", Book2.subject);
   printf("Bk2 id %d\n", Book2.book_id);

   return 0;
}
```

Structures as Function Arguments

```
struct Books
{
    char  title[50];
    char  author[50];
    char  subject[100];
    int   book_id;
};/* function declaration */
void printBook( struct Books book );
int main( )
{ //after book specification
  /* print Book1 info */
  printBook( Book1 );
  /* Print Book2 info */
  printBook( Book2 );
  return 0;
}
```

Structures as Function Arguments


```
void printBook( struct Books book )
{
   printf("Book title: %s\n", book.title);
   printf("Book author: %s\n", book.author);
   printf("Book sub: %s\n", book.subject);
   printf("Book id: %d\n", book.book_id);
}
```

- The function body

Array of Structures

```
struct inventory {
        int part_no;
        float cost;
        float price;
};

struct inventory table[4];
```

Thank You