Structures

By U.Sandhya
Asst. Prof.
CSE Dept.

JYOTHISHMATHI INSTITUTE OF TECHNOLOGY & SCIENCE

Structures

- Derived data type.
- Collection of related variables under one name.
- May contain variables of many different data types.
- Defining a structure

```
struct [structure tag] {

   member definition;
   member definition;
   ...
   member definition;
} [one or more structure variables];
```

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

- Use member access operator (.) to access the structure members
- Ex: book.author ; book.book_id

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

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

int main( ) {

   struct Books Book1;
   struct Books Book2;
```

```c
   strcpy( Book1.title, "C Programming");
   strcpy( Book1.author, "Nuha Ali");
   strcpy( Book1.subject, "C Programming Tutorial");
   Book1.book_id = 6495407;

   /* book 2 specification */
   strcpy( Book2.title, "Telecom Billing");
   strcpy( Book2.author, "Zara Ali");
   strcpy( Book2.subject, "Telecom Billing Tutorial");
   Book2.book_id = 6495700;

   /* print Book1 info */
   printf( "Book 1 title : %s\n", Book1.title);
   printf( "Book 1 author : %s\n", Book1.author);
   printf( "Book 1 subject : %s\n", Book1.subject);
   printf( "Book 1 book_id : %d\n", Book1.book_id);

   /* print Book2 info */
   printf( "Book 2 title : %s\n", Book2.title);
   printf( "Book 2 author : %s\n", Book2.author);
   printf( "Book 2 subject : %s\n", Book2.subject);
   printf( "Book 2 book_id : %d\n", Book2.book_id);

   return 0;
}
```

typedef

- **typedef** can be used to give a name to user defined data types

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

- Declaration:
  - Book book1;

Structures as function arguments

```
/* function declaration */
void printBook( struct Books book );
```

```
   /* print Book1 info */
   printBook( Book1 );

   /* Print Book2 info */
   printBook( Book2 );
```

```
void printBook( struct Books book ) {

   printf( "Book title : %s\n", book.title);
   printf( "Book author : %s\n", book.author);
   printf( "Book subject : %s\n", book.subject);
   printf( "Book book_id : %d\n", book.book_id);
}
```

Arrays of structures

```
#include <stdio.h>
#include <string.h>
struct student
{
    int id;
    char name[30];
    float percentage;
};
```

Arrays of structures

```
int main()
{
    int i;
    struct student record[2];
    // 1st student's record
    record[0].id=1;
    strcpy(record[0].name, "Raju");
    record[0].percentage = 86.5;
    // 2nd student's record
    record[1].id=2;
    strcpy(record[1].name, "Surendren");
    record[1].percentage = 90.5;
    // 3rd student's record
    record[2].id=3;
    strcpy(record[2].name, "Thiyagu");
    record[2].percentage = 81.5;
```

Arrays of structures

```
    for(i=0; i<3; i++)
    {
        printf("     Records of STUDENT : %d \n", i+1);
        printf(" Id is: %d \n", record[i].id);
        printf(" Name is: %s \n", record[i].name);
        printf(" Percentage is: %f\n\n",record[i].percentage);
    }
    return 0;
}
```