CSE 113 Structured Programming

Structure

Tasnim Zahan
Assistant Professor
Dept. of Computer Science & Engineering
North East University Bangladesh

Structure in C

- Structure is a user-defined datatype in C language which allows us to combine data of different types together
- It is similar to an Array, but an array holds data of similar type, but structure can store data of different types
- Structures are used to represent a record.
  - Students record: Reg#, name, semester
  - Book record: Title, Author, Subject
  - Address

Syntax of Structure

- To define a structure, you must use the ***struct*** statement. The ***struct*** statement defines a new data type, with more than one member.

```
struct struct_name {
   DataType var1;
   DataType var2;
   DataType var3;
   ....
};
```

Structure variable declaration

```
struct struct_name {
   DataType member1_name;
   DataType member2_name;
   DataType member3_name;
   …
} var_name;
```

OR

```
struct  struct_name  var_name;
```

```
struct Point
{
   int x, y;
   float dis;
} p1;
```

OR

```
struct Point
{
   int x, y;
   float dis;
};

int main()
{
   struct Point p1;
}
```

Accessing Structure Members

With structure

```
#include<stdio.h>

struct Point
{
    int x, y;
};
int main()
{
    struct Point p1;

    // Accessing members of point p1
    p1.x = 20;
    p1.y = 10;
    printf("x=%d, y=%d", p1.x, p1.y);

    return 0;
}
```

Without structure

```
#include<stdio.h>

int main()
{
    float x,y;

    x = 20;
    y = 10

    printf("x=%d, y=%d", x, y);

    return 0;
}
```

Array of structures

With structure

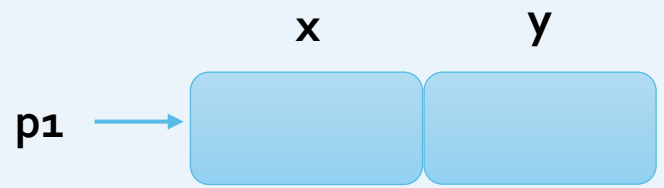
```
struct Point
{
    int x, y;
};

int main() {
    struct Point arr[10];

    arr[0].x = 10;
    arr[0].y = 20;

    printf("%d %d", arr[0].x, arr[0].y);
    return 0;
}
```

Without structure

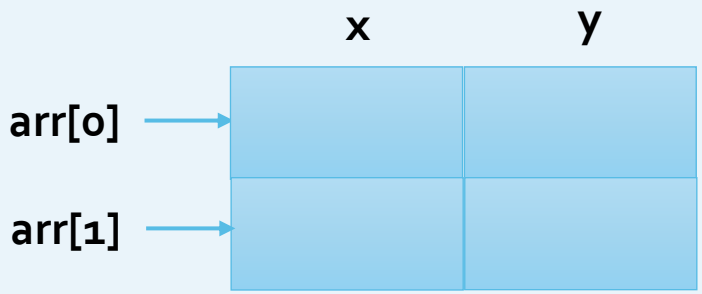
```
#include<stdio.h>

int main()
{
    int arr[10];

    arr[0] = 10;

    printf("%d %d", arr[0]);
    return 0;
}
```

Structure visualization

```
struct Point
{
    int x, y;
} p1;
```




```
struct Point
{
    int x, y;
};

int main()
{
    struct Point arr[10];
}
```



Example

```
struct StudentData{
    char stu_name[30];
    int stu_id;
    int semester;
};
int main()
{
     struct StudentData student;

     strcpy(student.stu_name, "ABC");//student.stu_name = "ABC"; ERROR
     student.stu_id = 1234;
     student.semester = 2;

     /* Displaying the values of struct members */
     printf("Student Name is: %s", student.stu_name);
     printf("\nStudent Id is: %d", student.stu_id);
     printf("\nSemester is: %d", student.semester);
     return 0;
}
```

Structures as Function Arguments

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

int main( ) {
  struct Books book1, book2;

  strcpy(book1.title,"Computer Programming");
  strcpy(book1.author,"Tamim Shahriar");
  strcpy(book1.subject,"C Programming Tutorial");
  book1.book_id = 10038;

  printBook( book1 );

  return 0;
}
```

```
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);
}
```

Use of typedef (type definition)

- typedef makes the code short and improves readability

With typedef

```
struct home_address {
   int local_street;
   char *town;
   char *my_city;
   char *my_country;
};

main(){
    struct home_address var;
    var.town = "Agra";
}
```

Without typedef

```
typedef struct {
   int local_street;
   char *town;
   char *my_city;
   char *my_country;
} home_address;

main(){
    home_address var;
    var.town = "Agra";
}
```

Pointers to Structures

- structure type pointer variable declaration same as other variable –

```
struct Books *struct_pointer;
```

- Assigning address of a structure variable in the pointer variable.

```
struct_pointer = &book1;
```

- Accessing the members of a structure using a pointer

```
struct_pointer->title;
```

Pointers to Structures

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

int main( ) {
  Books book1, book2;

  strcpy(book1.title,"Computer Programming");
  strcpy(book1.author,"Tamim Shahriar");
  strcpy(book1.subject,"C Programming Tutorial");
  book1.book_id = 10038;

  printBook( &book1 );

  return 0;
}
```

```
void printBook( 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);
}
```

Nested Structure

```
struct complex
{
    int imag;
    float real;
};

struct number
{
    struct complex comp;
    int integers;
}num1, num2;
```

num2.comp.imag = 11;