جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الأمن السيبراني

Lecture: 7

Structures

Subject: Structured Programming
First Stage
Lecturer: Asst. Prof. Dr. Ali Kadhum Al-Quraby

```
d3.inches -=12.0;
d3.feet ++;
}
d3.feet +=d1.feet + d2.feet;
cout<<d1.feet<<"\'-"<<d1.inches<<"\"+";
cout<<d2.feet<<"\'-"<<d2.inches<<"\"=";
cout<<d3.feet<<"\'-"<<d1.inches<<"\"\n";
}
```

4. Structures within Structures:

You can nest structures within other structures. Here's a variation on the English system program that shows how this looks. In the bellow program we want to create a data structure that stores the dimensions of a typical room: its length and width. Since we're working with English distances, we'll use two variables of type distance as the length and width variables.

Example 3:

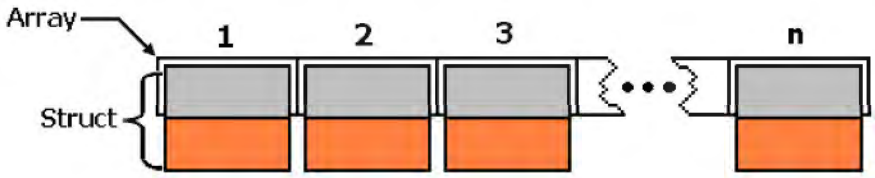
Write C++ program to find the area of the room in English system.

```
#include<iostream.h>

struct distance
{
int feet;
float inches;
}

struct room
{
distance length;
distance width;
};

Void main ()
{
room dining;
dining.length.feet=13;
dining.length.inches=6.5;
dining.width.feet=10;
dining.width.inches=0.0;
float L=dining.length.feet+dining.length.inches/12;
```

```
float W=dining.width.feet+dining.width.inches/12;
cout<<"\n Dining room area is"<<L*W<<"Square feet";
}
```

5. Array of Structures:

The **struct** is a data-type. So we can define an array as an array of struct, like define an array as an array of int, or of any other C++ data-types.



However, the following simple example shown how can create and use an ***array of struct***.

Example 4:

This simple example to show how can create and use an array of structure.

```
#include<iostream.h>

typedef struct
{
        char *name;
        int age;
} student;

void main ( )
{
        student  array [10];
        array [1] . name = "ahmed";
        array [1] . age = 20;
        cout << array[1] . name << endl;
        cout << array[1] . age;
}
```

```
cin >> array [1] . name ;
cin >> array [1] . age ;
```

Example 5:

Write a C++ Program, using structure type, to read name and age for ten students.

```
#include<iostream.h>

typedef struct
{
        char *name;    //Or name[10]
        int age;
}  student;

void main (  )
{
        student   array [10];

        for ( i = 0 ; i < 10 ; i++ )
        {
            cin >> array [i] . name;
            cin >> array [i] . age;
        }

        for ( i = 0 ; i < 10 ; i++ )
        {
        cout  <<  array[i] . name  <<  endl;
        cout  <<  array[i] . age;
        }

}
```

6. Functions and Structures:

A structure can be passed to a function as a single variable. The scope of a structure declaration should be an external storage class whenever a function in the main program is using a structure data types. The field or member data should be same throughout the program either in the main or in a function.

```
# include <iostream.h>
Struct sample  {
Int x;
Float y;
};
Sample first ;
Void main (void)
{
Void display (struct  sample  one);   //  function declaration
-  - -
- - -
Display (one);     // function calling
     - - -
     - - -
     }
Void display (struct  sample  out)   // function definition
{
     - - -
     - - -
Out.x=10;
Out.y=-20.20;
     - - -
     - - -
}
```

Example 6:

Write C++ program to display the contents of a structure using function definition.

```
#include<iostream.h>

struct date   {
int day;
int month;
int year;
}  ;
Void main(void)
{
date today;
void display (struct date one);   // function declaration
today.day=10;
today.month=3;
today.year=2011;
display (today);
}
Void display (struct  date one)
{
cout<<"Today's date is =" << one.day << "/";
cout<< one.month;
cout<<"/" << one.year << endl;
}
```

Output

Todat's date is = 10/3/2011

WORK SHEET (8)

Structures

Q1: Write a C++ program, that declares the structure called Employee_Info, which having the following members:

1- Employee name. (must be less than 25 characters)
2- Employee age. (must be 2 digits)
3- Employee address. (must be less than 20 characters)
4- Phone number. (must be 8 or 11 digits)
5- Country name. (must be less than 29 characters)

Then read and print this information for the 100 Employees.

Q2: Show the declaration of the following:

Employees:
- **Ind- Employees:**
  - **ID.**
  - **Name.**
  - **Sex.**
  - **Rate.**
- **Home:**
  - **Street.**
  - **City.**
  - **State.**
- **BirthDate:**
  - **Month.**
  - **Day.**
  - **Year.**
- **StrartDate:**
  - **Month.**
  - **Day.**
  - **Year.**

By using your declaration, write a C++ program that reads and stores data, then print only employees whose ID number less than 100.