LAB 3
Spring 2011, BESE- 16
C++ Classes and Objects

Objectives

- Introduction to Classes and Objects in C++
- Difference between classes and structures
- Create classes with public and private members
- Create member functions of class
- Constructor overloading
- Create array of class objects
- Pass class objects as arguments
- Pass arguments to class members

Submission Requirements

You are expected to complete the assigned tasks within the lab session.

Class Definition:

The standard format for a C++ class definition is as follows. Note that definitions are terminated with an ending semi-colon after the closing bracket.

```
class <classname>
{
        private:
                // private members go here
        protected:
                // protected members go here
        public:
                // public members go here
};
```

Member Functions:

Class objects call member functions, or methods, using the following syntax. In object-oriented programming, we think of calling member functions as “sending a message” to the object.
<object_name>.<function_name>(<arguments>);

C++ Classes vs Structures:

A **struct** in C++ is defined to be a variant of a class where all the members default to **public** (all members in a **class** default to **private**). C++ structures can be defined to have the same features as a class in C++.

Access Privileges

A C++ class provides three levels of information hiding and protection using the following access keywords.

private

- accessible **only** by other members of the class
- cannot be accessed by derived classes

protected

- level of data hiding between private and   public

- normally used when deriving a new class
- considered private to users (outside class)
- accessible by other members of the class
- accessible by other classes derived from the class

public

- accessible by other members and any user of the class

The keywords **private, protected, public** are optional and, by default, all members are **private** unless specified otherwise. Class members take on the level of security by the keyword preceding them in the class definition. These keywords can be placed in any order and can be included more than once. With regard to these keywords, a good practice when designing classes is as follows: **Make all members private that do not need to be accessed outside of the class (hide the implementation details and keep data internal)**

Task 1:

The class declaration for a Phone Book class is given below. Implement the class functions.

```
class PhoneBook
{
private:
        char name[30];
        char address[50];
        char city[20];
        double phoneNumb;

public:

void insert(char[], char[], char[], double);
        //This function is called when a new entry has to be made into the phonebook. The name,
        address, city and phone number are passed to the function and are assigned to the calling
        object's members.
void update(char[],char[],char[],double);
        //This function is called when a record has to be updated. The new values for the members
        are given as arguments.
void display (double price);
        //this function is called to display all data members of the invoking object
};
```

1. Write a function called total_count() which gives the value of the total number of objects created at any time.
2. In the main function, create an array of PhoneBook Objects which can hold 6 PhoneBook Objects The syntax for making an array of objects is given below:

```
    const int SIZE = 6;
    PhoneBook list[SIZE];
```

3. Write a search function (outside the class) that takes from user a string, representing a name. An array of PhoneBook objects is passed to it as an argument. That string is matched with the ‘name’ attribute of all objects. If the object with required name is found, the record is displayed.
    void Search(PhoneBook []);

Do the following in main:
  Display the information of all 4 objects
  Call the update function on the 4objects.
  Display the information of all updated objects

Task 2:

Modify the above program to add constructors.

PhoneBook();
  // constructor sets name to “”, address to “”, city to “” and numb to 0. Use the strcpy function to copy character arrays
PhoneBook(char n[], char a[], char c[], double nm);
  // constructor sets name to n, address to a, city to c and numb to nm.

Task 3:

Create an Employee class. The data members of this class are:

a. Name
b. NIC (store in numeric form)
c. Designation
d. BasicPay

The member functions of the class are:

a. A default constructor
b. Another constructor which takes four arguments and initializes the data members to values provided in the main function
c. An “UpDate” function that updates the designation of the employee. It would take as an argument a string provided in the main function.
d. An UpDate function that takes an integer as an argument and sets the value of BasicPay equal to the value provided.
e. An UpDate function that takes as argument a string and an integer. The Designation is set equal to the input string and the BasicPay is set equal to the integer value provide.
f. A CalculateBonus function. It requires an integer input that signifies the percentage of basic pay added as bonus
   Formula: Bonus = BasicPay * BonusPercentage
   void CalculateBonus (int);
g. A CalculatePay function that would calculate the total net salary of any employee.
   Formula: BasicPay + Medical Allowance (5% of BasicPay) + HomeAllowance (105 of BasicPay) + Miscellaneous allowances (5% of BasicPay)
   void CalculatePay();