CL-217 OBJECT ORIENTED PROGRAMMING LAB

LAB MANUAL 8

INSTRUCTOR: MUHAMMAD HAMZA

SEMESTER SPRING 2020

Q1.

Using classes, design an online address book to keep track of the names,addresses, phone numbers, and dates of birth of family members, close friends, and certain business associates. Your program should be able to handle a maximum of 500 entries.

a. Define a class addressType that can store a street address, city, state, and ZIP code. Use the appropriate functions to print and store the address. Also, use constructors to automatically initialize the member variables.

b. Define a class extPersonType using the class personType (implementation given), the class dateType (implementation given), and the class addressType . Add a member variable to this class to classify the person as a family member, friend, or business associate. Also, add a member variable to store the phone number. Add (or override) the functions to print and store the appropriate information. Use constructors to automatically initialize the member variables.

c. Define the class addressBookType using the previously defined classes. An object of the type addressBookType should be able to process a maximum of 500 entries.
The program should perform the following operations:
I. Load the data into the address book from a disk.
II. Sort the address book by last name.
III. Search for a person by last name.
IV. Print the address, phone number, and date of birth (if it exists) of a given person.
V. Print the names of the people whose birthdays are in a given month.
VI. Print the names of all the people between two last names.
VII. Depending on the user's request, print the names of all family members, friends, or business associates.

```
#include <string>
using namespace std;
class personType
{
public:
void print() const;
//Function to output the first name and last name
//in the form firstName lastName.
```

```
    void setName(string first, string last);
      //Function to set firstName and lastName according
      //to the parameters.
      //Postcondition: firstName = first; lastName = last

    string getFirstName() const;
      //Function to return the first name.
      //Postcondition: The value of firstName is returned.

    string getLastName() const;
      //Function to return the last name.
      //Postcondition: The value of lastName is returned.

    personType(string first = "", string last = "");
      //Constructor
      //Sets firstName and lastName according to the parameters.
      //The default values of the parameters are null strings.
      //Postcondition: firstName = first; lastName = last

 private:
    string firstName; //variable to store the first name
    string lastName;  //variable to store the last name
};
```

Figure 10-11 shows the UML class diagram of the `class` **personType**.

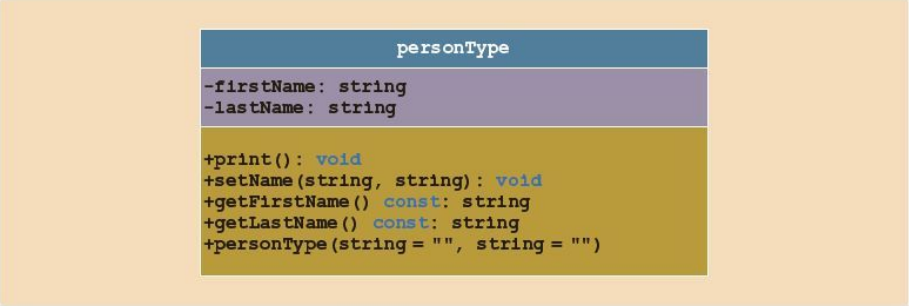

FIGURE 10-11 UML class diagram of the `class` **personType**

We now give the definitions of the member functions of the `class` **personType**.

```
void personType::print() const
{
    cout << firstName << " " << lastName;
}

void personType::setName(string first, string last)
{
    firstName = first;
    lastName = last;
}
```



```
string personType::getFirstName() const
{
```

```
return firstName;
}
string personType::getLastName() const
{
return lastName;
}
//constructor
personType::personType(string first, string last)
{
firstName = first;
lastName = last;
}
class dateType
{
public:
void setDate(int month, int day, int year);
//Function to set the date.
//The member variables dMonth, dDay, and dYear are set
//according to the parameters.
//Postcondition: dMonth = month; dDay = day;
//
dYear = year;
int getDay() const;
//Function to return the day.
//Postcondition: The value of dDay is returned.
int getMonth() const;
//Function to return the month.
//Postcondition: The value of dMonth is returned.
int getYear() const;
//Function to return the year.
//Postcondition: The value of dYear is returned.
void printDate() const;
//Function to output the date in the form mm-dd-yyyy.
dateType(int month = 1, int day = 1, int year = 1900);
//Constructor to set the date
//The member variables dMonth, dDay, and dYear are set
//according to the parameters.
//Postcondition: dMonth = month; dDay = day; dYear = year;
//
If no values are specified, the default//
values are used to initialize the member
//variables.
```

Previously you have developed the above program. Extend your program by adding below functionalities.

- Add a copy constructor in class addressType which will copy all the attributes of a previously created object to a newly created object.
- Add a private data member in extPersonType class (**nickName**), create a class myFriend, add basic data members like name, age etc (as many as you wish), make the myFriend class **friend class** of extPersonType class, myFriend should have a setter and getter function to set and get **nickName** of extPersonType class.
- Add a function **displayAll** which will be a friend function of class addressType, displayAll will print all the saved records of the address book in a formatted order.