San Francisco Bay University

CS360 - Programming in C and C++
Homework Assignment #6

Due day: 4/14/2024

Instruction:

1. Push the answer sheets/source code to Github
2. Please follow the code style rule like programs on handout.
3. Overdue homework assignment submission can't be accepted.
4. Take academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)

1. 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. Design the class *dateType* was designed to implement the date in a program, but the member function *setDate* and the constructor do not check whether the date is valid before storing the date in the member variables. Rewrite the definitions of the function *setDate* and the constructor so that the values for the month, day, and year are checked before storing the date into the member variables. Add a member function, *isLeapYear*, to check whether a year is a leap year. Moreover, write a test program to test your class.

   b. 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.

   c. Define a class *extPersonType* using the class *personType* as follows, the class *dateType*, 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.

   d. 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.

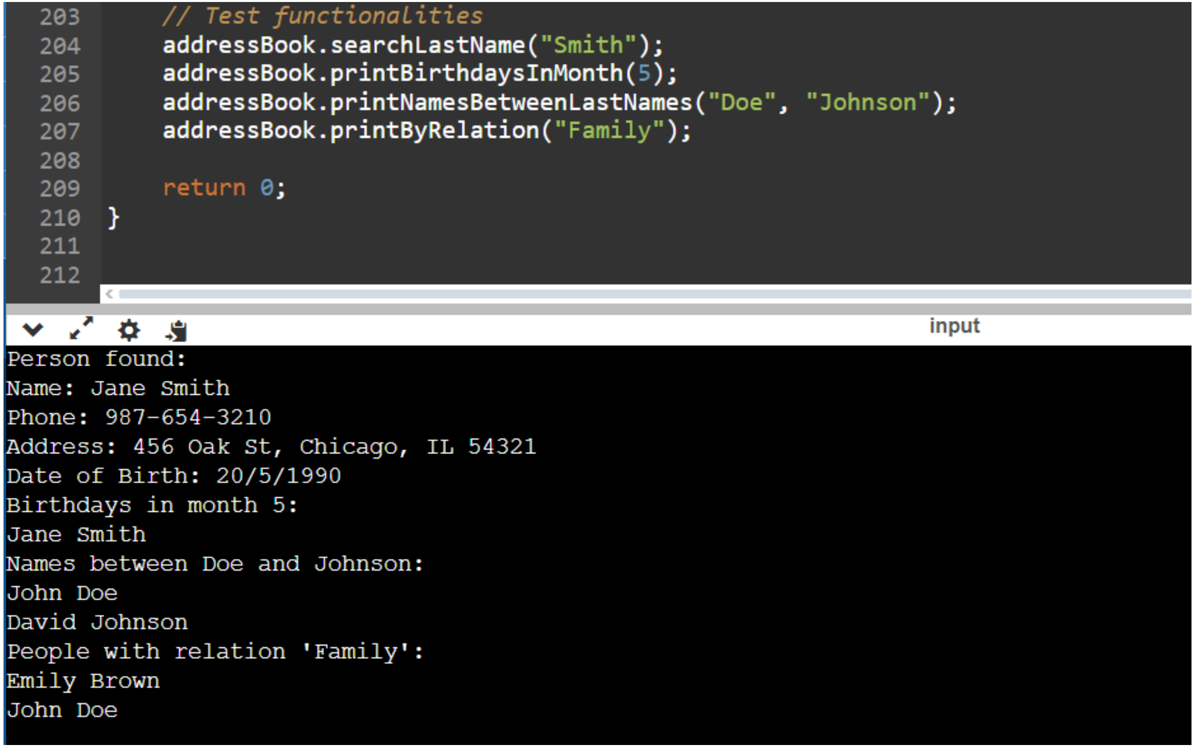
vi. Print the names of all of the people between two last names.

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
};

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;
}
```

Here is the output and the main code is in the cpp file



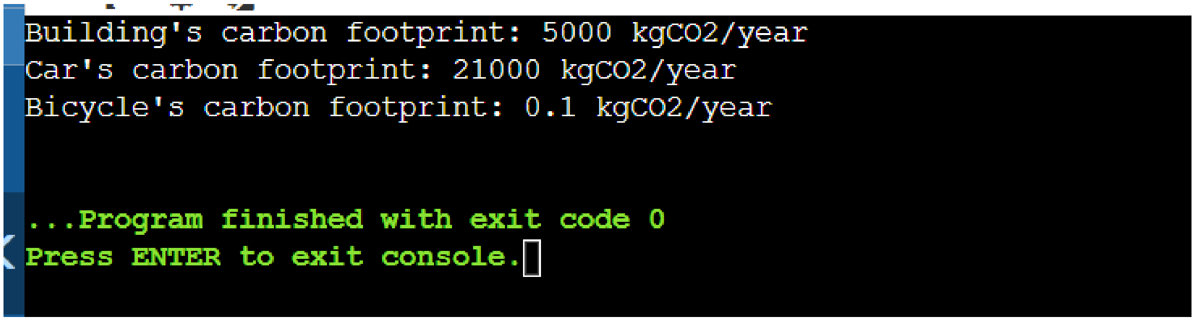
```cpp
    // Test functionalities
    addressBook.searchLastName("Smith");
    addressBook.printBirthdaysInMonth(5);
    addressBook.printNamesBetweenLastNames("Doe", "Johnson");
    addressBook.printByRelation("Family");

    return 0;
}
```

```
Person found:
Name: Jane Smith
Phone: 987-654-3210
Address: 456 Oak St, Chicago, IL 54321
Date of Birth: 20/5/1990
Birthdays in month 5:
Jane Smith
Names between Doe and Johnson:
John Doe
David Johnson
People with relation 'Family':
Emily Brown
John Doe
```

2. Using an abstract class with only pure virtual functions, you can specify similar behaviors for possibly disparate classes. Governments and companies worldwide are becoming increasingly concerned with carbon footprints (annual releases of carbon dioxide into the atmosphere) from buildings burning various types of fuels for heat, vehicles burning fuels for power, and the like. Many scientists blame these greenhouse gases for the phenomenon called global warming. Create three small classes unrelated by inheritance -- classes *Building*, *Car* and *Bicycle*. Give each class some unique appropriate attributes and behaviors that it does not have in common with other classes. Write an abstract class *CarbonFootprint* with only a pure virtual *getCarbonFootprint* method. Have each of your classes inherit from that abstract class and implement the *getCarbonFootprint* method to calculate an appropriate carbon footprint for that class *(check out a few websites that explain how to calculate carbon footprints, such as https://www.youtube.com/watch?v=wWeI0c1m14Y).* Write an application that creates objects of each of the three classes, places pointers to those objects in a vector of *CarbonFootprint* pointers, then iterates through the vector, polymorphically invoking each object's *getCarbonFootprint* method. For each object, print some identifying information and the object's carbon footprint.

Here is the output and the main code is in the cpp file



```
Building's carbon footprint: 5000 kgCO2/year
Car's carbon footprint: 21000 kgCO2/year
Bicycle's carbon footprint: 0.1 kgCO2/year


...Program finished with exit code 0
Press ENTER to exit console.
```