CS 2400 – Fall 2022

Project 5 – Classes

Due: 11:59 PM Friday, December 2nd

Purpose: This project is a continuation of the last. Now that you are an expert in storing letter data, you are going to create a "mini" post office. To do this, you will need to write 3 classes: Home, Neighborhood, and PostOffice.

Requirements:

1. You are allowed to use any code you have already written for projects 3 and 4.
2. ALL CLASSES NEED TO HAVE:
    a. At least a default constructor.
    b. Accessor functions.
    c. Mutator functions.
    d. An output function that neatly generates a report of all the data the object knows about.
        i. For the Neighborhood output, you should be able to call the Home output on each object in the vector.
        ii. For the PostOffice output, you should be able to call the Neighborhood output on each object in the vector.
3. The Home class should consist of an Address and a vector of Letter objects. Your mutator function for the Letter vector should add whatever Letter was passed in to the vector.
    a. Write an overloaded == or != operator that you can use to see if two Home objects have all matching attributes for their Address portion (you can ignore the vector of Letters).
4. The Neighborhood class should consist of a name for the neighborhood as well as a vector of Home objects. The mutator for the Home vector should add whatever Home was passed in to the vector.
5. The PostOffice will service 5 neighborhoods so, you should have an array that can hold exactly 5 Neighborhood objects. Use a constant in the public section of the class to declare the size of the array. You will also need a private counter variable of some kind to keep track of how many Neighborhood objects have been "set".

6. Write a main program
    a. Declare a post office object.
    b. Create a menu that allows the user to:
        i. Create a new neighborhood (just enter the name at this time) and add it to the PostOffice.
        ii. Enter the information for a Home and add it to the Neighborhood the user selects.
            1. You will need to prompt for the Neighborhood the user wants to add a Home to as well as the Home information.
            2. Before you add this Home to the Neighborhood, make sure that it is not already a part of the Neighborhood using your overloaded operator.
        iii. Add a letter that was sent to a particular Home in a particular Neighborhood.
            1. You will need to prompt for the Neighborhood name, Home address, and Letter information.
        iv. Output all the PostOffice data (all Neighborhoods and Homes). Output this information to a file in a way that is easy for the reader of the file to understand.

You are encouraged to add other functions as you find them useful.

Submit your source code and data file on Blackboard.

Grading:

Programs that contain syntax errors will earn zero points.

Programs that do not include functions other than main will earn zero points.

Programs that do not include any structures will earn zero points.

Programs that use global variables other than constants will earn zero points.

Programs that use libraries NOT discussed in class will earn zero points.

Your grade will be determined using the following criteria:

- Correctness: the program works as requested above (95 points).
    - (5 points) Home class contains all the correct data and functions
    - (5 Points) Neighborhood class contains all the correct data and functions
    - (5 points) PostOffice class contains all the correct data and functions
    - (60 points) The main program works as requested
    - (20 points) Output is correct and easy to read
- Documentation and Style (5 points)
    - Include Doxygen style comments for each function with meaningful comments for the purpose of the function and the parameters
    - Include comments in your functions explaining what you are doing