CMPT 225 Assignment 1: Creating a List ADT

Please, read the entire assignment first before starting it!

This assignment can be done in pairs - Form groups of 2 on CourSys when submitting your work.

Objectives

The objective of this assignment is for us to gain practice with the following:

- Design and implement a List (data collection) abstract data type (ADT) class that satisfies a set of given requirements.
- Aply the 4 steps of the software development process while creating an object-oriented (OO) solution to a problem.

Problem Statement

In this assignment, you are asked to develop a walk-in clinic patient system. The goal of this system is to manage the medical records of all the patients of the walk-in clinic.

The walk-in clinic keeps the following information about all its patients: the patient's name (first and last name), address, phone number, email address and care card number.

The walk-in clinic patient system must allow the receptionist to

- enter a new patient into the system
- remove a patient from the system
- search for an existing patient
- modify a patient's record (for example, adding the patient's information or making a change of address, etc...).
- print all its patients by ascending order of care card numbers.

About the "print" option from the above menu, here is what its output should look like (i.e., the format).

Requirements

Here is the list of the requirements your solution to this walk-in clinic patient system must satisfy:

1. The file that represents the walk-in clinic patient system, i.e., the file that contains the *main* function must be called walkIn.cpp.
2. Download Patient.h and Patient.cpp. As you read through them you will notice that they are incomplete. You need to complete them following the requirements and hints included in these files.
3. Download List.h and read its content. You can add more attributes/methods to this class, but you cannot remove its attributes/methods nor can you change them. You will also need to create List.cpp.
    - Why can you not mofidy the public interface given in Patient.h and List.h? Because, when we mark Assignment 1, we shall test the submitted classes with a test driver program (client code) that has been implemented using the specifications (Public Interface) of the Patient and the List classes, i.e., the public part of Patient.h and List.h. If you change the provided public interfaces (for example, if you change the type of a parameter from "float" to "int"), the test driver program used for marking your assignments will no longer compile with your submitted Patient and the List classes, and marks will be deducted.
4. All the classes you create and use as part of your solution must be designed and implemented as abstract data type (ADT) classes.
5. Your solution must not break any of the class invariants.
6. Your List must be implemented using an array (already done for you).
7. Each of your files must contain a comment header block composed of filename, class description, class invariant (if any), author, date of creation.
8. All your class methods (public and private) must have appropriate documentation: a description, a precondition (if any) and a postcondition (if any). Most of this documentation has already been provided for you. Make sure you understand its purpose. NOTE: This documentation must appear in the header file (.h) of each of your classes and be reproduced in the class's implementation file (.cpp).
9. Your solution must not make use of already existing container classes such as the STL vector class, etc...
10. Finally, make sure your code satisfies the *Good Programming Style* described on the Good Programming Style web page of our course web site.

You can use this Makefile to compile your assignment.

Marking Scheme

When marking Assignment 1, we shall consider some or all of the following criteria:

- Solving a problem: Does your solution solve the problem stated in this assignment?
- Correctness of the submitted classes: do they work with the "marking" test driver programs?
- Design and implementation of the List ADT: Does it satisfy the requirements stated in this assignment?
- Coding style: Have proper Good Programming Style and documentation been used?
- In general: Does your solution satisfy the requirements described in the Requirements section above?

Remember ...

- You can use any C++ IDE you wish to do your assignments in this course as long as the code you submit compiles and executes on the CSIL computers.

  Why? Because your assignments will be marked using this platform (Linux and C++ - version running on CSIL computers). We strongly suggest that you compile and execute your code using g++ at the Linux command line in CSIL before you submit your assignment to CourSys.

Submission

- Assignment 1 is due Wednesday, January 23 at 3pm.
- In the interest of fairness to our classmates, and expedient marking, late assignments will not be marked.
- Since this assignment can be done in pairs, you are required to form groups of 2 on CourSys in order to submit your work - even if you have done this assignment on your own. In this case, you must form a group of 1 on CourSys.
- Submit the 5 files: Patient.h, Patient.cpp, List.h, List.cpp and walkIn.cpp via CourSys. Only one group member is required to do the submission for the group.
