LAB #3
More Classes and File I/O

Remember, if you need to get Lab #1 graded, you need to show your lab to the TAs within 10 minutes of getting to lab, and you and your partner will not receive lab credit if you do not get checked off before leaving each lab. Once you have a zero on a lab, then it cannot be changed because we have no way of know if you were there or not!!!

Reminder: All of our labs involve paired programming. You do not have to keep the same partner for each lab, but you MUST work with someone in each lab!!! First, find a partner for this lab. It can be the same partner from the previous lab or a different partner.

You are going to write the program from your design in Exercise #2…

This program will compute a patient's bill for a hospital stay. The different components of the program are

- The **PatientAccount** class will keep a total of the patient's charges. It will also keep track of the number of days spent in the hospital. The group must decide on the hospital's daily rate.
- The **Surgery** class will have stored within it the charges for at least five types of surgery. It can update the charges variable of the PatientAccount class.
- The **Pharmacy** class will have stored within it the price of at least five types of medication. It can update the charges variable of the PatientAccount class.
- The main (driver) program.

The main program will have a menu that allows the user to **enter a type of surgery**, **enter one or more types of medication**, and **check the patient out of the hospital**, which needs to know how many days were spent in the hospital. When the patient checks out, the total charges should be displayed to the screen, and the patient information should be appended to a file called **hospital.txt.**

You must make sure every class has a proper constructor, destructor, copy constructor, and **const** used in the correct places. In addition, your **headers must include preprocessor guards** for including the header, i.e. #ifndef PATIENTACCOUNT _H.

(2 pts) Design Interface: PatientAccount.h, Surgery.h, Pharmacy.h
(YOU MUST GET CHECKED OFF FOR THIS FIRST!!!)

Design a class called PatientAccount, Surgery, and Pharmacy. Make sure each of these have the proper constructors, destructors, copy constructors, and const keywords.

**Make sure you have your preprocessor guards for your .h file, e.g.
#ifndef PATIENTACCOUNT_H

YOU MUST GET CHECKED OFF FOR THIS FIRST BEFORE MOVING TO .CPP FILES

(2 pts) Design Implementation: PatientAccount.cpp, Surgery.cpp, Pharmacy.cpp
Now, design your implementation files. Write the pseudocode for each of the member functions designed in the header files first.

(2 pts) Design Testing Program: main.cpp
The main program will have a menu that allows the user to **enter a type of surgery**, **enter one or more types of medication**, and **check the patient out of the hospital**, which needs to know how many days were spent in the hospital. When the patient checks out, the total charges should be displayed.

(4 pts) Write the Program for all the classes and main (driver) files.
Now, implement your three designs/files to show you have written a complete and correct classes for PatientAccount, Surgery, and Pharmacy. Create a Makefile to compile and run your program.

Add File Output:
When the patient checks out, the **total charges should be displayed to the screen**, and the patient information should be appended to a file called **hospital.txt with the information separated by a pipe delimeter.**

Example hospital.txt:
Name|pharm|medX_name|medX_cost|surg|surgX_name|surgX_cost|Days_in

Jennifer Parham-Mocello|pharm|asprin|50.00|surg|tonsils|50.00|1

Example Code for File Output:
```
#include <fstream>

using std::fstream;
int main() {
   fstream output;

   output.open("hospital.txt", ios::app);

   output << p.get_name();
   output << "|pharm|";
   output << p.get_pharm_med(0);
   output << '|';
   output << p.get_pharm_med_cost(0);
   ...
   ouput << endl;

   output.close();

   return 0;
}
```