ECS 40 **Program #5 (50 pts, 2? hours)** **Spring 2016**

Due:Wednesday, May 4th, 11:59 PM Executable name: airline.out Handin to cs40a p5

Filenames: authors.csv, main.cpp, utilities.cpp, utilities.h, plane.cpp, plane.h, flight.cpp, flight.h, passenger.h, passenger.cpp, flights.h, flights.cpp, linkedlist.cpp, linkedlist.h, and Makefile.

New concepts: linked lists, operator overloading.

There are three parts to this assignment. First, you will implement a singly linked list using two new classes, ListNode, and LinkedList. You will replace each row of the passengers array with a LinkedList object. Most of the LinkedList methods will be overloaded operators. The second part of the assignment is to create overloaded operators either anew, or from pre-existing functions. The third part of the assignment adds the ability to print information about the passengers of the entire airline. You are welcome to use my p4 code as a starting point. ~ssdavis/40/p4/SeansSrc will be open Thursday morning. You can find passengers*.dat, flights*.csv, my airline.out, and PrintDatFile.out in ~ssdavis/40/p5.

Specifications, and order of development. I implemented them in the order in which they are presented.

1. LinkedList, is a singly linked list.
   - 1.1. ListNode class holds an int named offset and a pointer to another ListNode. Its only method is a private "standard" constructor. The LinkedList class is a friend of the ListNode class. ListNode is implemented in linkedlist.h, and linkedlist.cpp.
   - 1.2. The LinkedList class holds just a pointer to a ListNode, named head.
     - 1.2.1. Each Plane will now hold a dynamically allocated array of LinkedList—one linked list for each row. You should simply change passengers in Plane to a LinkedList*.
     - 1.2.2. The constant EMPTY should be moved from the Plane class to the LinkedList class.
     - 1.2.3. The LinkedList class will have the following methods
       - 1.2.3.1. A function named initialize() that takes the number of seats in the row. This should create a ListNode with an EMPTY offset for each seat in the row. Have the Plane constructor and Plane::addFlight() call this method once it knows it knows the width of the plane. There is no need for a LinkedList constructor.
       - 1.2.3.2. A destructor.
       - 1.2.3.3. Two overloaded [], one const and another non-const, that returns the offset of a ListNode based on the index parameter, which is an int seatNum. Note that the non-const version allows the offset in the ListNode to be changed. The seatNum 0 would access the first ListNode in the list.
   - 1.3. Though you will have to make some changes in Plane, you will not need to make any other changes in Flights, Flight, or Passenger to make the LinkedList implementation work, and the program run perfectly! This is an example of the advantage good modularity and data encapsulation.
2. Overloaded operators
   - 2.1. Passenger class
     - 2.1.1. Our goal is to add overloaded operators so that is no longer necessary to have the Plane class as a friend. To start, we will need to temporarily make the Plane constructor that takes an ifstream a friend of the Passenger class.
     - 2.1.2. Add an overloaded == operator that takes an int as its parameter, and returns true if the flightNum of the Passenger matches the parameter. Change the Plane(inf) constructor to rely on this operator instead of accessing the Passenger::flightNumber.
     - 2.1.3. Add an overloaded == operator that takes a const char* as a parameter, and returns true if the name of the passenger matches the parameter. Change the Plane::removePassenger() to rely on this operator instead of accessing the Passenger::name.
     - 2.1.4. Add an overloaded ! operator that sets the flightNum to Passengers::CANCELLED. This will allow Plane::removePassenger() and Flight::removeFlight() to "remove" a passenger without accessing its flightNum directly.
     - 2.1.5. Add an overloaded << operator that prints the name of the passenger. This can be called from Plane::showPassengers().
     - 2.1.6. You should be able to no longer have the Plane class a friend of the Passenger class at this point.
   - 2.2. Plane class
     - 2.2.1. Before proceeding, add an int flightNumber to the Plane class, and change the default constructor to take the flightNumber as its only parameter. Remove flightNumber from the parameter list of all the other Plane methods.
     - 2.2.2. Change the Plane constructor that took an ifstream as its parameter to an overloaded extraction operator that is a friend of the Plane class, and a friend to the Passenger class. Eliminate the previous friendship in Passenger for the corresponding Plane constructor. You will have to add "plane." or "rhs." in front of every Plane variable in the function. You will have add a call to the constructor of 2.2.1 to Flight::readFlight() before the call to this function.

2.2.3. Change showPassengers() to an overloaded insertion operator that takes an ostream as its first parameter, and is a friend of the Plane class. The call to this function will be one of the few times it is correct to use "this", albeit you will use "*this".

2.2.4. Change addPassenger() to an overloaded pre-increment (++) operator. Since traditionally this operator returns the object "incremented", so should this function. Now that Plane has its flightNumber, move the warning about being full from Flight::addPassenger to this function. Remember that plane is a Plane* in the Flight class, and the ++ operator has very high precedence.

2.2.5. Change removePassenger() to an overloaded post-decrement (--) operator. This function should also return the object "decremented."

2.2.6. Change removePlane to an overloaded ! operator. This function should also return the object negated.

2.3. LinkedList class

2.3.1. Create an overloaded insertion operator that prints out the grid 'X' and '-' for the entire row. This should traverse the linked list, so it will have to be a friend of ListNode as well. This will be called by Plane::showGrid() and use some of the code from that function.

2.4. Flight class

2.4.1. Change addPassenger() to an overloaded post-increment operator.

2.4.2. Change printFlightInfo() to an overloaded insertion operator.

2.4.3. Change readFlight() to an overloaded extraction operator.

2.4.4. Change removeFlight() to an overloaded ! operator.

2.4.5. Change removePassenger() to an overloaded pre-decrement operator.

2.5. Flights class

2.5.1. Change addFlight() to an overloaded pre-increment operator.

2.5.2. Change insert() to an overloaded += operator.

2.5.3. Change removeFlight() to an overloaded post-decrement operator.

2.5.4. Change readFlights() to an overloaded extraction operator. Now have main() open and close flights.csv, as well as calling this function.

3. (15 minutes) You will add the capability to find a passenger by name to your program. You may not write any new methods for the Passenger class. You may assume that the name will be the correct size, but not necessarily in the flights.
4. All methods must have the appropriate const declarations.
5. Code must be submitted by exactly one member of each team. Double submissions, or errors in the authors.csv format will result in the team losing five points. Your handin command line will be:
   handin cs40a p4 authors.csv main.cpp plane.h plane.cpp flight.h flight.cpp flights.h flights.cpp passenger.h passenger.cpp utilities.h utilities.cpp linkedlist.h linkedlist.cpp Makefile

```
[ssdavis@lect1 p5]$ airline.out

ECS Flight Reservation Menu
0. Exit.
1. Add Passenger.
2. Remove Passenger.
3. Add Flight.
4. Remove Flight.
5. Find Passenger.

Please enter your choice: 5

Name of passenger: Dupont Ayala,Tsz Chi
Flight #561 Row: 1 Seat: B

ECS Flight Reservation Menu
0. Exit.
1. Add Passenger.
2. Remove Passenger.
3. Add Flight.
4. Remove Flight.
5. Find Passenger.

Please enter your choice: 5

Name of passenger: Castaneda,Hasith
Flight #118 Row: 0 Seat: A

Flight #768 Row: 6 Seat: C

ECS Flight Reservation Menu
0. Exit.
1. Add Passenger.
2. Remove Passenger.
3. Add Flight.
4. Remove Flight.
5. Find Passenger.

Please enter your choice: 5

Name of passenger: Castaneda,Hasit
Castaneda,Hasit not found.

ECS Flight Reservation Menu
0. Exit.
1. Add Passenger.
2. Remove Passenger.
3. Add Flight.
4. Remove Flight.
5. Find Passenger.

Please enter your choice: 0
Goodbye.
[ssdavis@lect1 p5]$
```