CSCE 1040 Homework 5 Fall 2017

For this assignment we are going to re-design and modify the code for a system to schedule flight crews and aircraft for Mean Green Airlines that we started in Homework 3 and Homework 4.

For this we will need the following entities, plus collections for each of the entities: Plane/Aircraft, Crew Member and Flight.

The data for a Plane will contain at least the following:
Make (e.g. Boeing)
Model (eg 737)
Tail Number (eg B171A)
Number of Seats (eg 137)
Range (e.g. 800 miles)
Status (out, in, repair)

You may add other data needed for your implementation as well as you will need accessor and mutator functions for the data.

The data for a crew member will contain at least:
Name
ID number
Status (available, on leave, sick)

The Crew Member class will be a base class for three child classes of

Pilot, CoPilot and Cabin

Pilot will need to add data for
5 character rating code
cumulative flight hours as pilot

CoPilot will need data for
4 character rating code
cumulative flight hours as a CoPilot

Cabin crew will need data for
Position : First Class, Business Class EconomyFront, EconomyRear, Lead

There should be a virtual PrintInfo function that will print all the info for a given crew

member starting with the base information and then the details for the particular type
Note this function will be useful in the collection class reporting methods. It should first be declared in the base class.

You may add other data needed for your implementation as well as you will need accessor and mutator functions for the data.

The data for a Flight (The transaction entity) will contain at least the following:

Plane ID (Tail number)
Pilot ID
CoPilot ID
Crew IDs for 3 Cabin Crew Members
Start Date/Time with TZ
End Date/Time with TZ
Starting Airport code (3 letters)
Ending Airport Code (3 letters)
Number of Passengers
Status (active, cancelled, completed)

You may add other data needed for your implementation as well as you will need accessor and mutator functions for the data.

For the collections of each of the 3 Entity Classes identified above you will need to include the ability to:

Add
Edit
Delete
Search/Find based on appropriate criteria
Print a list of all entries in the collection
Print the details for a single entity (do a find first)

Note that the crew collection class will now be a mixed collection and will have to use run-time polymorphism for certain capabilities.

for the Flights collection when you add a flight you will need to verify that
a. the plane selected is available during the defined time period
b. the plane selected has number of seats sufficient for the passengers
c. the crew selected are of the appropriate types and assigned to the

proper roles, and that they are not already assigned to another flight at the same time. Also that they are available to assign

Note that a particular plane or crew member could have multiple assignments as long as they do not conflict with dates or times. For this assignment you do not need to worry about verifying availability based on starting and ending points.

You will also need to provide in the Flights collection the ability to print an assignment schedule for a particular aircraft or for a particular crew member of all the active assignments. Also to print a list of flights based on their status. You should also provide a means to delete all cancelled flights or all completed flights form the menu. You should also provide a means to periodically update all flights from active to completed based on time and date.

You will need to provide an appropriate menu system that can be multi-level if you like.

You will need to load and store the data. This can be done automatically when the program starts and ends. You should also want to store after an add, delete or edit to make sure changes to the data are preserved.

For this design portion you will need to turn in an updated design including the following:

A UML diagram set consisting of:

1. A title page with your name, assignment, course and title
2. a single class diagram showing only the relationships between the entities
3. a set of nine individual class diagrams showing the attributes and methods for each of the classes in #1
4. A set of State Diagrams for any stateful variables (depends on your design but probably 2-3 at least). Note these may be the same a the ones form Homework 3.
5. Step by Step algorithms for every method defined in every class in pseudocode. Again you can reuse homework 3 and only add the new or revised methods affected by this design change
6. A 1-2 paragraph report about your design experience

All of these items should be gathered together, in order, in a single PDF File that you will turn in on BB Learn. This is worth 40% of the grade.

Then you will need to modify the code from Homework 4 to implement the new features you have designed. This will be worth 60% of the total grade for this assignment. You should collect all the cpp files and h files, as well as your makefile into a single zip file. You should also include 1-2 paragraphs in your report about the development experience in addition to the section about the design.