COEN 244 (Winter 2025) - Assignment 2

Assignment 2:	Class composition
Deadline:	Monday, February 17 by 11:59pm
Type:	You can work individually or in groups of two students.
Weight:	5%

Submission instructions:

- Add name and student ID of both group members at the top of each file.
- Compress the files using zip or other tools.
- Submit the zip file on Moodle; assignments sent by email will not be corrected
- Do not submit executable files.
- Late submissions will not be accepted.

Questions:

We want to create a flight reservation system for a company called CoenAir. The system should allow creating flights and passenger, and booking passengers on various flights. To do so, we need to create a certain number of classes and use the class composition concept to link objects of these classes.

Q1. (15 points) Create a class called Flight that represents CoenAir flights. A flight has a flight id (string), a departure city (string), an arrival city (string), a departure date (object of class Date seen in the lectures), an arrival date (object of type Date), and a duration in terms of hours (int). The class should have one or more constructors, accessing functions, a copy constructor, a destructor, a function that prints information about a flight. Write a driver to test the class Flight. Note: You can reuse the class Date as it is.

Deliverables: A zip file that contains: Time.h, Time.cpp, Date.h, Date.cpp, Flight.h, Flight.cpp, TestFlight.cpp

Q2. (15 points) Write a class called Passenger that represents the passengers of CoenAir. A passenger has an id (string), first name (string), last name (string), address (string), and email (string). The class should have one or more constructors, accessing functions, a copy constructor, a destructor, a function that prints information about a passenger. Write a driver to test the Passenger class.

Deliverables: A zip file that contains three files: Passenger.h, Passenger.cpp, TestPassenger.cpp

Q3. (25 points) We want to create a class called Airline that keeps track of the flights of CoenAir. The class Airline has the following attributes: the name of the company (string), the address of the headquarter of the company (string), an array of objects of class Flight. Note that vectors can be used instead of arrays. The question as to which data structure to use and why will be the subject of one of the lectures. The class should have one or more constructors, accessing functions, a copy constructor, a destructor, as well as the following functions:

- addFlight(const Flight&): This function adds a flight object to the array of flights.

- removeFlight(string): This function removes a flight from the array of flights. It takes the flight id as input.
- searchFlight(string): This function returns true if a given flight exists. It takes the flight id as input.
- displayFlights(): This function display all the flights of the airline.
- displayFlightsFromTo(string, string): This function takes the departure and arrival cities and displays information about flights that take off and land at these cities.

Deliverables: A zip file with all the header and cpp files.

Q4. (30 points) We need to book passengers on flights. One way to do this is to add an array (or vector) of Passenger objects to the Flight class. We also need to create the appropriate member functions to update the array. These include:

- addPasenger(const Passenger&): This function adds a passenger object to the array of passengers of the class Flight.
- removePassenger(string): This function removes a passenger from the array of passengers. It takes the passenger id as input.
- searchPassenger(string): This function returns true if a given passenger exists. It takes the passenger id as input.
- displayPassengers(): This function display all the passengers of a flight.

Improve the class Flight of Q1 to include the Passenger array. As usually, the class should have one or more constructors, a copy constructor, accessing functions, and a destructor.

Deliverable: A zip file that contains: Time.h, Time.coo, Date.h, Date.cpp, Flight.h. Flight.cpp, Passenger.h, Passenger.cpp, TestFlight.cpp

Q5 (15 points) We want to improve Q4 by adding a booking date, which is the date a passenger makes a booking. Note that a passenger can make bookings for multiple flights. Discuss your solution. No need to write code.

Q6 (Bonus Questions) (10 points). Implement your solution for Q5. This is a bonus question and it is optional. If you don’t do it, you can still get 100% by doing Q1 to Q5.

Assignment Marking Scheme for each question:

- Program correctness (80%)
- Program clarity, completeness, and accuracy and readability (5%)
- Comments - description of variables and constants (5%)
- Test cases should be comprehensive enough to cover your program to test if it is bug free (10%)

Note on using GenAI tools such as ChatGPT:

You are allowed to use GenAI as an educational resource. If you decide to do that, **you must provide the GenAI tool you have used, a list of all the prompts** you **used, and a detailed explanation of your contribution.** Note that the TAs reserve the right to meet students and ask them questions about their assignments and they suspect plagiarism due to the use of GenAI.