CENG 218 2023-2024 Spring

Lab 2: Flight information and passenger operations
Design and implement a C++ program to manage Flight information and passenger operations. This system provides a user-friendly interface for flight management, allowing users to add and remove flights, view flight details, and handle passenger reservations and cancellations. The core of the system revolves around three **primary** classes: FlightManager, Flight, and Passenger.

Core Classes
FlightManager: It be responsible for maintaining the list of flights. This class will encapsulate operations related to flights, including adding, removing, and listing flights. The FlightManager will interact with the Flight class for individual flight operations

FlightManager Class:
This class is responsible for managing the list of flights. It provides functionality to add, remove, and list flights.

Member Variables:
- flights: std::vector<Flight>

Methods:
- addFlight(const Flight& flight)
- removeFlight(const std::string& flightNumber)
- listAllFlights()
- getFlightByNumber(const std::string& flightNumber): Flight
- getFlightByDestination(const std::string&destination): Flight

Flight Class: It encapsulates essential information about each flight, including a unique flight number, destination, maximum available seats (default set to 40 and number of seats must be even and upper limit is 80), the current number of passengers, and a list of passengers stored in a vector. Various methods within this class enable operations such as reserving seats, canceling reservations, retrieving the number of passengers, printing passenger lists, and checking the destination of the flight.

- **Member Variables:**
  - flightNo: string
  - destination: string
  - maxSeats: int (default = 40)
  - numPassengers: int
  - passengers: std::vector<Passenger>
- **Methods:**
  - reserveSeat(const Passenger& passenger)
  - cancelReservation(const Passenger& passenger)
  - int numberOfPassengers()
  - void printPassengers()
  - bool isFlyingTo(const std::string& destination)

Passenger Class: The Passenger class represents individuals with attributes such as name, surname, and gender. This class includes constructors, getters, setters, and overloaded operators to facilitate easy manipulation and retrieval of passenger information.

- **Member Variables:**
  - name: string
  - surname: string
  - gender: char
- **Constructors, Getters, Setters, Overloaded Operators:** (As before)

The program will be based on a structured menu system to guide users through flight and passenger management operations. The top-level menu focuses on flight management, offering options to add or remove flights, list all available flights, select a specific flight for passenger management, and exit the program. The second-level menu, accessible by selecting a flight. It provides options for reserving seats, canceling reservations, viewing the passenger list, and returning to the flight management menu.

Menu Structure

Top-Level Menu: Flight Management

1. Add a Flight
2. Remove a Flight
3. List All Flights
4. Select a Flight and Manage passengers
5. Exit

Functionality

1. **Adding Flights:** Obtain the flight number and destination from the user. Create a new Flight object and store it appropriately (vector).
2. **Removing Flights:** Allow the user to specify a flight (potentially by flight number) and remove it from your storage.
3. **Listing Flights:** Display basic information (flight number, destination, number of available seats) for all current flights.
4. **Selecting a Flight:** Prompt the user for a flight number (or some way to identify the flight) and transition to the passenger management menu for that specific flight.

Second-Level Menu: Passenger Management (For a selected flight)

1. Reserve a Seat (By displaying seating plan)
2. Cancel a Reservation
3. View Passenger List
4. Back to Flight Management Menu

Second-Level Menu details
Passenger Management (For a selected flight)
1. Reserve a Seat
- While user in this menu following seating plan will be displayed and get user details and seat number.
Consider all airplanes with rows of seats arranged in a grid-like structure. Each row has four seats labeled A, B, C, and D. Use the following legend:

```
Legend:
     X - Occupied Seat
     O - Vacant Seat

Seating Plan:
------------Front-----------
| 1A O | 1B O | |1C O | 1D O |
| 2A X | 2B O | |2C O | 2D O |
| 3A O | 3B O | |3C O | 3D O |
| 4A O | 4B O | |4C O | 4D X |
     ... up to number of seats
```

- Choose a row and seat to reserve. Provide passenger details for reservation

2. Cancel a Reservation
- Cancel an existing reservation. Provide passenger details for verification.

3. View Passenger List
- Upon selecting the "View Passenger List" option, the system will display a well-formatted passenger list for the selected flight. The passenger list format includes the following details for each passenger:

1. Seat Assignment:The seat assigned to the passenger, denoted by the row number and seat label (e.g., Row 1, Seat 1A).

2. Passenger Name: The passenger's full name, including both the first name and surname.

3. Gender:The gender of the passenger, represented by a single character (e.g., M for Male, F for Female).

List will in table format in which, each column has a fixed width to provide a consistent and organized appearance. The width of Seat is fixed to 5, and the width of the "Passenger Name" column is set to 15 characters(names exceeding this limit will be truncated with a period appended.), and final column has 1. And as a seperator the pipe symbol will be used between columns.

Passenger List for Flight XYZ123:

```
Seat|Passenger Name |Gender
----|---------------|------
1A  |John Doe       |M
2B  |Nathaniel Wash.|F
3C  |Abdulkadir Gor.|M
4D  |Emily Williams |F
```

4. Back to Flight Management Menu
- Return to the main menu for flight management.

Input Validation

- Valid flight numbers
  In the aviation industry, a flight number or flight designator is a code for an airline service consisting of two-character airline designator and a 1 to 4 digit number. For example, "BA 222" is a British Airways. (assume we have only two airline company TK THY PG for Pegasus, if you are interested more on this you may refer to: https://en.wikipedia.org/wiki/Flight_number#cite_note-37)
- Names consisting of letters only.

Error Handling

- Add/remove non-existent flights.
- Reserve on full flights.
- Cancel non-existent reservations.
- Select non-existent flights for passenger management.