IE5600 – Applied Programming for Industrial Systems

AY 2022/23 Semester 1

Individual Assignment 2 – PyFlights

Objectives

At the completion of the individual assignment, you should:

1. Develop a better understanding of how to apply the computational problem solving process to a complex problem that requires the use of appropriate data structures and algorithms.
2. Implement procedural and object-oriented programming paradigms in Python.
3. Apply appropriate user-defined data structures and algorithms.

General Requirements

You are required to develop a Python program known as **PyFlights**, which is essentially a flight ticket reservation system for a typical online travel agency. PyFlights allows passengers to book flight tickets from multiple airlines based on the desired criteria such as availability, time, lowest airfare, lesser number of stopovers, shortest stopover time and shortest flight time.

Commercial airlines mainly use two different types of network structure when planning and scheduling flights – hub-and-spoke network structure and point-to-point network structure.

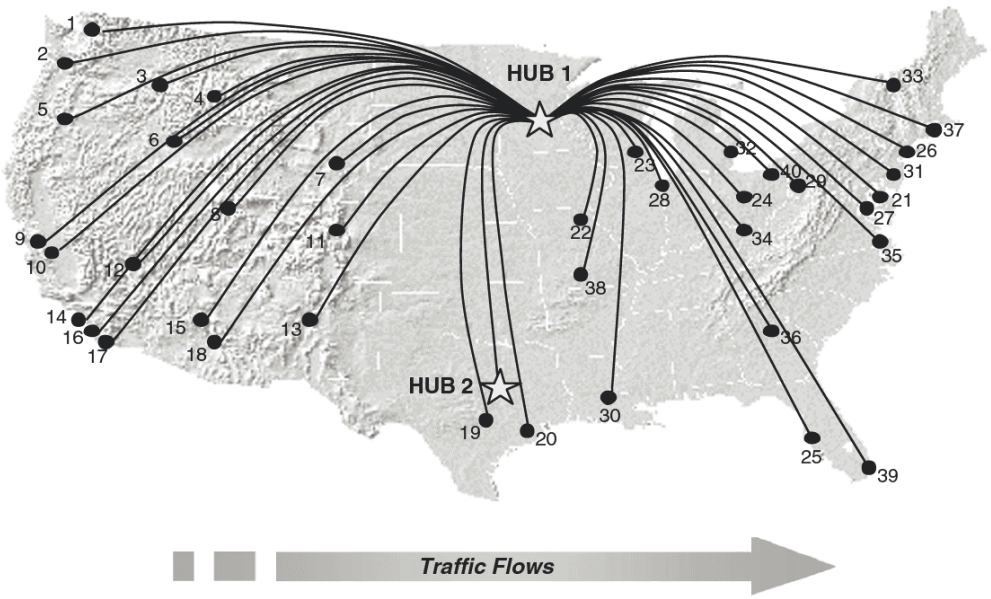

Figure 1 – An example of a connecting hub network. (Belobaba, Odoni and Barnhart 2009, pp. 163, Figure 6.5)

In a hub-and-spoke network structure, flights are organised into multiple flight segments. Each flight segment represents an origin–destination (O-D) market, typically between two cities (or rather two airports of the two cities). A hub-and-spoke network structure allows airlines to serve many O-D markets with fewer flight departures, requiring fewer aircrafts and generating fewer ASK (Available Seat Kilometers) at lower total operating costs than in a complete point-to-point route network. An example of a simple connecting hub network with 20 flights into and 20 flights out of a single "connecting bank" at a hub airport is shown in Figure 1.

A "connecting bank" refers to a hub operation in which many aircrafts arrive at the hub airport, passengers and baggage are moved between connecting flights, and the aircrafts then depart with the connecting passengers and baggage on board. Connecting banks last from approximately 1 hour in smaller domestic hub networks to 2–3 hours in larger international hub networks.

In this example of a connecting bank with 20 arriving flights followed by 20 departing flights, each flight leg arriving or departing the hub simultaneously serves 21 O-D markets – one "local" market between the hub and the spoke, plus 20 additional "connecting" markets if we assume a single direction of passenger flow. This single connecting bank thus provides service to a total of 440 O-D markets with only 40 flight legs and as few as 20 aircraft flying through the hub. In contrast, a complete "point-to-point" network providing non-stop service to each market would require 440 flight legs and hundreds of aircrafts, depending on scheduling requirements. Of course, it is entirely possible for an airline to operate a mixture of direct flights (especially for high-value markets) and hub-and-spoke network.

In practice, each flight route consisting of an origin-destination (O-D) pair is referenced by two points. Each point refers to a specific airport identified by its IATA code located in a particular city, state/province and country. Suppose we have a hypothetical airline Merlion Airlines (IATA code: ML). Let's take an O-D pair of Singapore Changi Airport (IATA code: SIN) and Narita International Airport in Narita, Chiba, Japan (IATA code: NRT), i.e., SIN-NRT, as an example. A passenger in Singapore could take a direct flight from Singapore to Narita, e.g., flight ML123 SIN-NRT. A passenger travelling from Sidney Airport in Sidney, New South Wales, Australia (IATA code: SYD) to Narita could take a flight ML456 SYD-SIN to Singapore before connecting to ML123 SIN-NRT to complete the entire trip SYD-SIN-NRT. It is also possible that the airline offers a direct flight from Sydney to Narita, e.g., ML789 SYD-NRT. Passengers choose between direct flight and connecting flight with one or more stopovers according to various factors such as availability, time, price, and stopover duration.

Each aircraft is configured into one or more cabin classes or booking classes, each with its own seating capacity. An airline typically offers four cabin classes, i.e., first class (F), business class (J), premium economy class (W) and economy class (Y). Each cabin class has its own seat configuration organised into one or two aisles with up to seven to ten seats abreast. For example, the economy cabin class of a wide-body aircraft is typically configured into 3-4-3 with two aisles. The left and right columns have three seats abreast and the middle column has four seats abreast. For simplicity, you may ignore booking classes for this assignment. In other words, all seats in a same cabin class would be sold at the same rate and restrictions.

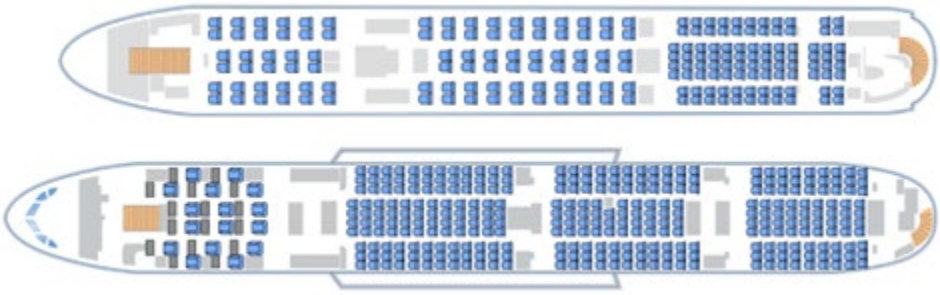

Figure 2 – An Airbus A380 layout with 519 seats displayed.

Each seat is designated by a row number followed by a letter. For the purpose of this assignment, you may assume that row numbers and seat letters are sequential with no skipping. It is also not necessary to skip letters that may be confused with numbers (e.g., I, O, Q, S, or Z). In the preceding example of a 3-4-3 seating configuration, the seats in row 32 would be designated as 32A to 32J.

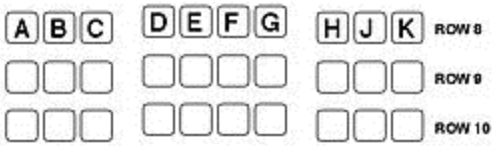

Figure 3 – Real-world seat designation with right-hand 3 seats skip over letter "I"

Design and Implementation of Classes

Design a set of suitable classes together with the necessary inheritance and/or association relationships to represent a set of airlines, aircraft types, flights, passengers and flight tickets to enable the online travel agency to maintain a database of flights for selling flight tickets to passengers.

For simplicity, you may assume that each airline operates a few types of aircrafts, but it is not necessary to manage the actual fleet of aircrafts. It is also not necessary to assign an aircraft to a particular flight. It would suffice to assign an aircraft type to a flight. When searching for available flights, you may assume that connecting flights must be from the same airline. In real-world context, it is possible to have connecting flights from different airlines but that would require a passenger to retrieve his/her luggage and recheck the luggage on the connecting flight with a different airline.

It is mandatory to apply the object-oriented programming paradigm to this assignment. Otherwise, you would only be awarded 50% of the marks for the use cases.

Use Cases

Use the classes that you have defined to implement the following use cases for PyFlights without the use of any predefined Python modules other than those specified below:

- `datetime`

Your program should **NOT** contain any other `import` statement unless you are importing your own user-defined module(s).

S/N	Use Case	Use Case Description/Business Rules
1	**Load Initial Data** (1 mark)	• Auto load the initial data given in Appendix A • You required to use appropriate data structures to represent the initial data in-memory. • It is not necessary to save the data to a file.
2	**Create Airline** (1 mark)	• Create a new airline record in the system. • If an airline already exists in the system, print a suitable error message, and discard the new record. Sample Input: • Airline Name – Singapore Airlines • IATA Code – SQ

3	**Create Aircraft Type**
	(2 marks)
		• Create a new aircraft type record for an airline in the system.
		• If an aircraft type already exists in the system, print a suitable error message, and discard the new record.
		• Perform input data validation as appropriate to ensure correct and logical input data.
		
		Sample Input:
		• Airline IATA Code – SQ
		• Aircraft Type Name – Boeing 747
		• Cabin Class Configuration
		  ○ Add Cabin Class – Y
		  ▪ Cabin Class – F
		  ▪ Seat Configuration – 1-1-1 (two dashes denote two aisles with seats A, B and C)
		  ▪ Starting Row Number – 1
		  ▪ Ending Row Number – 10
		  ○ Add Cabin Class – Y
		  ▪ Cabin Class – J
		  ▪ Seat Configuration – 2-2-2
		  ▪ Starting Row Number – 11
		  ▪ Ending Row Number – 20
		  ○ Add Cabin Class – Y
		  ▪ Cabin Class – Y
		  ▪ Seat Configuration – 3-4-3
		  ▪ Starting Row Number – 21

S/N	Use Case	Use Case Description/Business Rules
		▪ Ending Row Number – 50 o Add Cabin Class – N Sample Output: • Aircraft Type Boeing 747 created for SQ o Cabin Class F – Seat 1A to 1C through 10A to 10C, 30 seats o Cabin Class J – Seat 11A to 11F through 20A to 20F, 60 seats o Cabin Class Y – Seat 21A to 21J through 50A to 50J, 300 seats • Total 390 seats Repeat same Aircraft Type for Airline QF.
4	**Create Flight** (2 marks)	• Create a new flight record for an airline in the system tagged to a particular aircraft type. • The selected aircraft type will determine the number of available seats or tickets for that flight. • If a flight for an airline already exists in the system, print a suitable error message, and discard the new record. • Perform input data validation as appropriate to ensure correct and logical input data. Sample Input: • Airline IATA Code – SQ • Outbound Flight Number – 123 • Return Flight Number (Optional) – 321 • Origin Airport IATA Code – SIN • Destination Airport IATA Code – NRT • Departure Date/Time – 5 Dec 22, 10:00 AM • Flight Time – 6 hour 30 minute • Stopover Duration – 3 hour • Aircraft Type – Boeing 747 • Fare Amount: o Cabin Class F – $2,000 o Cabin Class J – $1,000 o Cabin Class Y – $500 Sample Input: • Airline IATA Code – SQ • Outbound Flight Number – 456 • Return Flight Number (Optional) – 654 • Origin Airport IATA Code – SYD • Destination Airport IATA Code – SIN • Departure Date/Time – 4 Dec 22, 10:00 PM • Flight Time – 8 hour 0 minute

S/N	Use Case	Use Case Description/Business Rules

		• Stopover Duration – 2 hour
		• Aircraft Type – Boeing 747
		• Fare Amount:
		  o Cabin Class F – $1,500
		  o Cabin Class J – $750
		  o Cabin Class Y – $300
		
		Sample Input:
		• Airline IATA Code – SQ
		• Outbound Flight Number – 789
		• Return Flight Number (Optional) – 987
		• Origin Airport IATA Code – SYD
		• Destination Airport IATA Code – NRT
		• Departure Date/Time – 5 Dec 22, 9:00 AM
		• Flight Time – 14 hour 0 minute
		• Stopover Duration – 2 hour
		• Aircraft Type – Boeing 747
		• Fare Amount:
		  o Cabin Class F – $3,500
		  o Cabin Class J – $2,500
		  o Cabin Class Y – $1,500
		
		Sample Input:
		• Airline IATA Code – QF
		• Outbound Flight Number – 789
		• Return Flight Number (Optional) – 987
		• Origin Airport IATA Code – SYD
		• Destination Airport IATA Code – NRT
		• Departure Date/Time – 5 Dec 22, 1:00 PM
		• Flight Time – 14 hour 30 minute
		• Stopover Duration – 2 hour
		• Aircraft Type – Boeing 747
		• Fare Amount:
		  o Cabin Class F – $3,200
		  o Cabin Class J – $2,200
		  o Cabin Class Y – $1,200

5	**Search and Book Flight** (4 marks)	• Prompt user to input the required trip details and return all available flights within ±1 day. • Search results should be listed by flight and cabin class • The fare should be computed according to the number of passengers and cabin class. • If user proceeds to book a particular flight, prompt user to select the number of required seat(s) and validate that the selected seat(s) is/are available. Sample Input for Search:

S/N	Use Case	Use Case Description/Business Rules

		• Trip Type – One-Way (Return or One-Way)
		• Origin – SYD
		• Destination – NRT
		• Departure Date – 5 Dec 22
		• Cabin Class – Y (F, J, W, Y or No Preference)
		• Number of Passengers – 1
		
		Sample Output for Search:
		• Result 1:
		  ○ SQ456, SYD to SIN, Depart 4 Dec 22 10:00 PM, Arrive 5 Dec 22 6:00 AM, Class Y $300
		  ○ SQ123, SIN to NRT, Depart 5 Dec 22 10:00 AM, Arrive 5 Dec 22 4:30 PM, Class Y $500
		  ○ 1 stopover, Total Fare $800
		• Result 2:
		  ○ SQ789, SYD to NRT, Depart 5 Dec 22 9:00 AM, Arrive 5 Dec 22 11:00 PM, Class Y $1,500
		  ○ 0 stopover, Total Fare $1,500
		• Result 3:
		  ○ QF789, SYD to NRT, Depart 5 Dec 22 1:00 PM, Arrive 6 Dec 22 3:30 AM, Class Y $1,200
		  ○ 0 stopover, Total Fare $1,200
		
		Sample Input for Book:
		• Select Flight – Result 3
		• Select Seat for Passenger 1 – 21A
		
		Sample Output for Book:
		• Your flight is confirmed.

Deliverable Submission

The assignment deliverable to be submitted to the Canvas Assignments tool are to be placed in a single zip archive file with the following folders structure:

- **source** subfolder containing:
  - All Python source files that constitute your program.
  - The main source file containing the program entry point should be named as pyflights.py, i.e., your program should be runnable with the command `python pyflights.py`

Upload this zip archive file to the designated Canvas Assignment: Assignments > Individual Assignment 2.

Your deliverables must be submitted latest by **Wednesday, 23 November 2022, 11:59 pm**. No submission will be accepted for assessment after this date/time and you will be awarded **0** marks.

-- End of Assignment Specification –

Appendix A – Initial Data

Airline

SQ, Singapore Airlines
QF, Qantas

Aircraft Type

SQ, Boeing 747,
    F, 1-1-1, 1, 10
    J, 2-2-2, 11, 20
    Y, 3-4-3, 21, 50
QF, Boeing 747,
    F, 1-1-1, 1, 10
    J, 2-2-2, 11, 20
    Y, 3-4-3, 21, 50

Flight

SQ, 123, 321, SIN, NRT, 5 Dec 22 10:00 AM, 6 hour 30 minute, 3 hour, Boeing 747
    F, 2000
    J, 1000
    Y, 500

SQ, 456, 654, SYD, SIN, 4 Dec 22 10:00 PM, 8 hour 0 minute, 2 hour, Boeing 747
    F, 1500
    J, 750
    Y, 300

SQ, 789, 987, SYD, NRT, 5 Dec 22, 09:00 AM, 14 hour 0 minute, 2 hour, Boeing 747
    F, 3500
    J, 2500
    Y, 1500

QF, 789, 987, SYD, NRT, 5 Dec 22, 01:00 PM, 14 hour 30 minute, 2 hour, Boeing 747
    F, 3200
    J, 2200
    Y, 1200

SQ, 123, 321, SIN, NRT, 12 Dec 22 10:00 AM, 6 hour 30 minute, 3 hour, Boeing 747
    F, 2000
    J, 1000
    Y, 500

SQ, 456, 654, SYD, SIN, 11 Dec 22 10:00 PM, 8 hour 0 minute, 2 hour, Boeing 747
    F, 1500
    J, 750
    Y, 300

SQ, 789, 987, SYD, NRT, 12 Dec 22, 09:00 AM, 14 hour 0 minute, 2 hour, Boeing 747
F, 3500
J, 2500
Y, 1500

QF, 789, 987, SYD, NRT, 12 Dec 22, 01:00 PM, 14 hour 30 minute, 2 hour, Boeing 747
F, 3200
J, 2200
Y, 1200