



SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Information Technology
Diploma in Cybersecurity & Digital Forensic
Diploma in Data Science

PROGRAMMING II

Year 2024/25 - Semester 2

ASSIGNMENT

Duration	: 2 weeks (20 Jan 2025 to 2 February 2025)
Weightage	: 30% of total coursework
Individual/Team	: Team of 2
Format	: Programming - Class Implementation (10%) Git Usage (5%) Basic Features (50%) Advanced Features (20%) Presentation (15%)

Cut-Off Date/Time: Sunday, 2 February 2025, 23:59hrs

Penalty for late submission:

- 10% of the marks will be deducted for each day (inclusive of Saturdays, Sundays and public holidays) after the deadline.
- No submission will be accepted after 09 Feb 2025, 23:59hrs.

There is a total of 10 pages (including this page) in this handout.

WARNING

If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action will also be taken. Similar action will be taken for the student who allows other student(s) to copy his/her work.

Flight Information Display System

In this assignment, you are to apply Object Oriented Programming to develop a simple **Flight Information Display System**. The assignment requirements described below are broken down into 2 stages of development, described in this document as '**Basic Features**' and '**Advanced Features**'. You are advised to do your programming progressively in these stages. Refer to the '**Grading Criteria**' to have an idea of how the different components are graded.

1. BACKGROUND

Singapore has begun construction on Terminal 5 at Changi Airport, and you have been tasked with developing a Flight Information Display System (FIDS) that will provide real-time flight information to passengers. The system will use electronic display boards and monitors throughout Terminal 5 to display and provide vital information regarding arriving and departing flights, including Flight Numbers, Airline Names, City of Origin or Destination, Expected Departure/Arrival Time, and Flight Status. More data is available in the data files: **airlines.csv** (for the available Airlines) and **flights.csv** (for the intended Flights).

Flight Number	Airline Name	Origin	Destination	Expected Departure/Arrival	Status
SQ 693	Singapore Airlines	Tokyo (NRT)	Singapore (SIN)	10:30am	On Time
MH 722	Malaysia Airlines	Kuala Lumpur (KUL)	Singapore (SIN)	11:00am	Delayed
JL 122	Japan Airlines	Tokyo (NRT)	Singapore (SIN)	12:15pm	On Time
CX 312	Cathay Pacific	Singapore (SIN)	Hong Kong (HKD)	1:00pm	Boarding
QF 981	Qantas Airways	Singapore (SIN)	Sydney (SYD)	2:45pm	Delayed

Table 1 – Sample Flight Data that an FIDS should display for flights in Terminal 5

In addition to these core functions, FIDS can provide information on Boarding Gates.

Flight Number	Boarding Gate
SQ 693	A13
MH 722	B2
CX 312	C22

Table 2 – Flight Numbers with an assigned Boarding Gate

Amelia, the diligent Flight Controller, relies on the FIDS to coordinate flights and manage air traffic efficiently. Otherwise, she might be replaced by automated systems. When Amelia receives a flight plan from an airline, it is placed in a Dictionary with the 4 required flight specifications (i.e. Flight Number, Origin, Destination, and Expected Departure/Arrival Time) to be added to the FIDS. Most importantly, at any point in time, there would be no more than 1 Flight with the same Flight Number on the same day.

Note: A flight plan can also include additional information. For example, if the flight has a Special Request Code that will affect the Boarding Gate, it would need.

Flight Number	Special Request Code	Special Request Description
SQ 693	DDJB	Double-decker jet bridge requested
CX 312	CFFT	Connecting flight fast transfer requested
QF 981	LWTT	Longer waiting time requested

Table 3 – Flight Numbers that included Special Request Codes in their flight plans

Amelia also ensures that she assigns the appropriate Boarding Gates to the Flight Numbers based on whether any Special Request Code is included in the flight plan when received. At Terminal 5, there are 66 Boarding Gates, ranging from A1 to A22, B1 to B22, and C1 to C22. Some of the Boarding Gates can accommodate these Special Request Codes, as per the table below, with the complete data available in the ***boardinggates.csv*** file.

Special Request Code	Boarding Gates
DDJB	A10, A11, A12, A13, A20, A21, A22, B10, B11, B12
CFFT	B1, B2, B3 + All C Gates (C1 to C22)
LWTT	A1, A2, A20, A21, A22, C14, C15, C16 + All B Gates (B1 to B22)

Table 4 – Boarding Gates that can satisfy Special Request Codes from flights

Amelia coordinates the flights in the sequence they are received, making sure they are successfully assigned a Boarding Gate to a Flight Number. When assigned, the Status for each Flight is set to “On Time”, unless otherwise specified. Furthermore, as part of slowing beginning operations at Terminal 5, Amelia can only assign 1 Boarding Gate to 1 Flight Number per day.

Of course, Amelia also needs to calculate the daily fees for the different airlines based on the number of flights they send into or out of Terminal 5 and the types of Boarding Gates they use, so that Changi Airport can send the correct bill to each airline.

Terminal Fees	Fee Per Action/Usage	Fee Description
Arriving Flight	\$500	Arriving Flights are Flights with Destination as Singapore (SIN)
Departing Flight	\$800	Departing Flights are Flights with Origin as Singapore (SIN)
Boarding Gate Base Fee	\$300	Base Fee for All Boarding Gates
DDJB Code Request Fee	\$300	Additional Fee when a Flight indicates the Special Request Code of DDJB
CFFT Code Request Fee	\$150	Additional Fee when a Flight indicates the Special Request Code of CFFT
LWTT Code Request Fee	\$500	Additional Fee when a Flight indicates the Special Request Code of LWTT

Table 6 – Terminal 5 Fees and their specific Fee Per Action/Usage

To encourage more airlines to use Terminal 5, Changi Airport provides incentives to airlines through discounts based on the following promotions. **Note:** The Promotions are stackable, which means as long as the airline fulfils any of the Promotional Conditions, they will be able to enjoy the discounts received.

Promotional Condition	Discount Received
For every 3 flights arriving/departing, airlines will receive a discount	\$350
For more than 5 flights arriving/departing, airlines receive an additional discount	3% off the Total Bill
For flights arriving/departing before 11am or after 9pm	\$110
For airlines with the Origin of Dubai (DXB), Bangkok (BKK) or Tokyo (NRT)	\$25
For not indicating any Special Request Codes	\$50

Table 7 – Promotions and Discounts received by airlines for Terminal 5

Based on the promotions above, an airline with 7 flights receives a discount of \$350 off and 3% off their Total Bill **before any deductions**. Likewise, an airline with a flight that is arriving at midnight, from Dubai (DXB), and not indicating any Special Request Codes, will receive discounts of \$110, \$25 and \$50 respectively, for a total discount of \$185.

The Class Diagram for the Flight Information Display System (FIDS) is shown in Figure 1 on the next page.

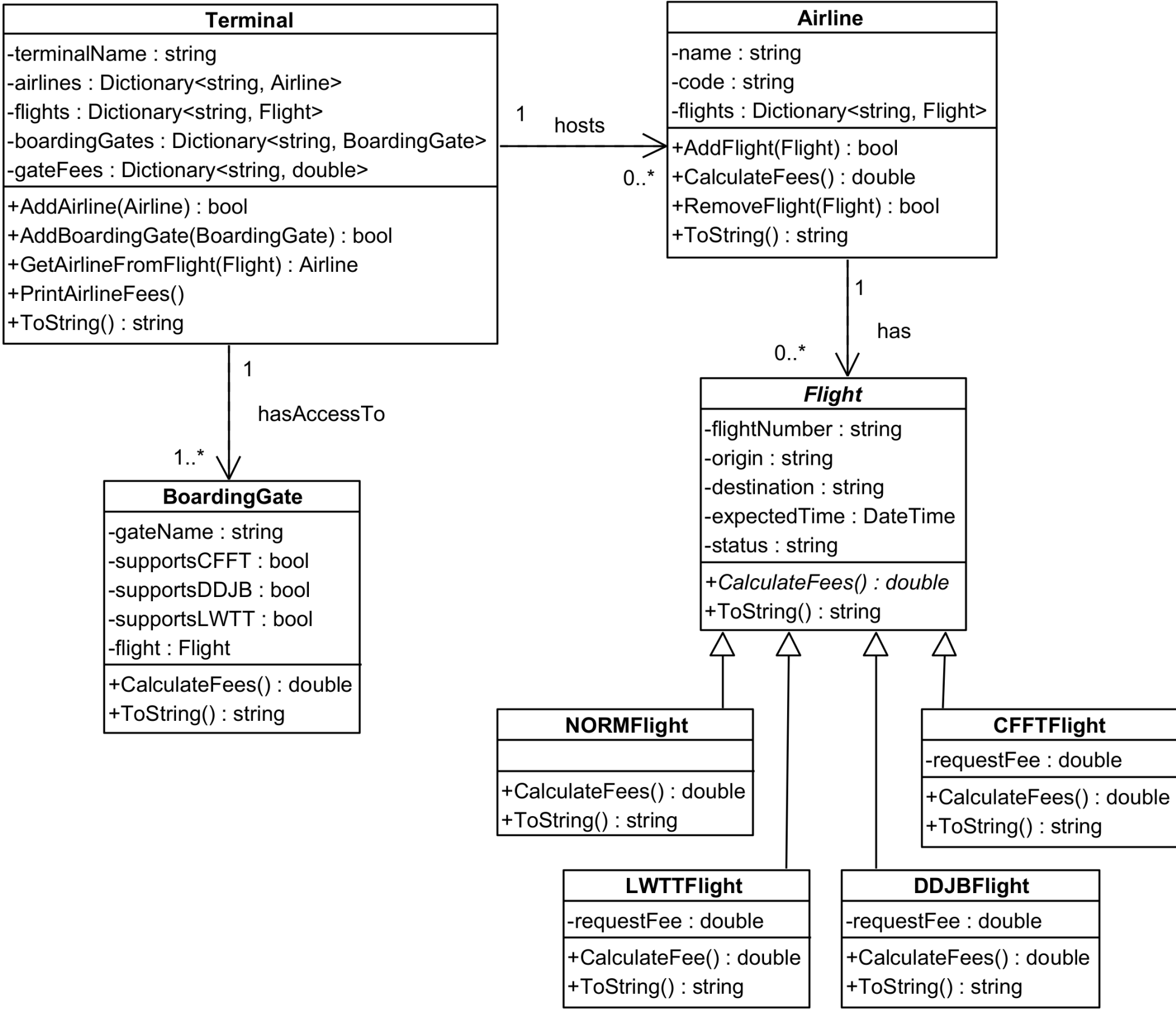


Figure 1: Class Diagram for Flight Information Display System

2. CLASS IMPLEMENTATION – 10% GROUP

The team has to divide the work when implementing the classes.

3. GIT USAGE – 5% GROUP

For this assignment, all teams must use Git and GitHub to collaborate effectively and track the progress of their project. Proper usage of these tools is a key learning objective and will form part of the evaluation criteria during your final presentation.

This means that:

1. Each team member must have an individual GitHub account
2. Teams must create a shared repository on GitHub and ensure all work is committed and pushed to this repository, with good use of branches and pull requests that show the collaboration.
3. Each team member is required to make individual contributions, which must be visible through commit histories with clear and concise commit messages.
4. During the final presentation, you will be required to demonstrate your team's use of Git and GitHub and explain how you used them to work collaboratively.

4. BASIC FEATURES – 50% INDIVIDUAL

Your program is required to create Airlines, Boarding Gates and Flights from the given data files at the onset, which represent a single day at Terminal 5 (as indicated in Basic Features 1 and 2) run only at the start of the program. After which, it displays a menu for the user to choose to perform each of the features described below repeatedly (for Basic Features 3 to 9, and later on to include the Advanced Features as well) until the user chooses to exit from the menu.

1) Load files (airlines and boarding gates)

- load the **airlines.csv** file
- create the Airline objects based on the data loaded
- add the Airlines objects into an Airline Dictionary
- load the **boardinggates.csv** file
- create the Boarding Gate objects based on the data loaded
- add the Boarding Gate objects into a Boarding Gate dictionary

2) Load files (flights)

- load the **flights.csv** file
- create the Flight objects based on the data loaded
- add the Flight objects into a Dictionary

3) List all flights with their basic information

- display the Basic Information of all Flights, which are the 5 flight specifications (i.e. Flight Number, Airline Name, Origin, Destination, and Expected Departure/Arrival Time)

4) List all boarding gates

- display all the Boarding Gates in Terminal 5 with all of the Special Request Codes they service (if any) and Flight Numbers assigned (if any)

5) Assign a boarding gate to a flight

- *prompt the user for the Flight Number*
- *display the basic information of the selected Flight, including the Special Request Code (if any)*
- *prompt the user for the Boarding Gate*
- *check that the selected Boarding Gate is not assigned to another Flight (Note: For Basic Features, there is no need to validate if the Special Request Codes between Flights and Boarding Gates match)*
 - *if the Boarding Gate selected is already assigned to another flight, display a message that the Boarding Gate is already assigned and repeat the previous step*
- *display the basic information of the selected Flight, Special Request Code (if any), and Boarding Gate entered*
- *prompt the user if they would like to update the Status of the Flight, with a new Status of any of the following options: "Delayed", "Boarding", or "On Time" [Y] or set the Status of the Flight to the default of "On Time" and continue to the next step if [N]*
- *display a message to indicate a successful Boarding Gate assignment*

6) Create a new flight

- *prompt the user to enter the new Flight, which minimally requires the 4 flight specifications (i.e. Flight Number, Origin, Destination, and Expected Departure/Arrival Time)*
- *prompt the user if they would like to enter any additional information, like the Special Request Code*
- *create the proper Flight object with the information given*
- *add the Flight object to the Dictionary*
- *append the new Flight information to the **flights.csv** file*
- *prompt the user asking if they would like to add another Flight, repeating the previous 5 steps if [Y] or continuing to the next step if [N]*
- *display a message to indicate that the Flight(s) have been successfully added*

7) Display full flight details from an airline

- *list all the Airlines available*
- *prompt the user to enter the 2-Letter Airline Code (e.g. SQ or MH, etc.)*
- *retrieve the Airline object selected*
- *for each Flight from that Airline, show their Airline Number, Origin and Destination*
- *prompt the user to select a Flight Number*
- *retrieve the Flight object selected*
- *display the following Flight details, which are all the flight specifications (i.e. Flight Number, Airline Name, Origin, Destination, and Expected Departure/Arrival Time, Special Request Code (if any) and Boarding Gate (if any))*

8) Modify flight details

- list all the Airlines available
- prompt the user to enter the 2-Letter Airline Code (e.g. SQ or MH, etc.)
- retrieve the Airline object selected
- for each Flight from that Airline, show their Airline Number, Origin and Destination
- prompt the user to either [1] choose an existing Flight to modify, or [2] choose an existing Flight to delete
 - if [1] is selected, have the user select which flight specification to modify, then prompt the user for the new information for the modifications they wish to make to the Flight selected: Basic Information except the Flight Number itself (i.e. Origin, Destination, and Expected Departure/Arrival Time), or Status, Special Request Code or Boarding Gate and update the Flight object's information accordingly
 - if [2] is selected, have the user select which Flight Number to delete, prompt for a confirmation [Y] or to delete [N], and then remove that Flight object (Note: For purposes of this assignment, you do not need to modify or delete records in the file)
- display the new updated Flight details with Basic Information of all Flights, which are all the flight specifications (i.e. Flight Number, Airline Name, Origin, Destination, and Expected Departure/Arrival Time, Status, Special Request Code (if any) and Boarding Gate (if any))

9) Display scheduled flights in chronological order, with boarding gates assignments where applicable

- display all flights for the day ordered by earliest first
 - ensure your flights implement the `Comparable<T>` interface
- for each flight, ensure that:
 - all Flight details are displayed with Basic Information of all Flights, which are all of the flight specifications (i.e. Flight Number, Airline Name, Origin, Destination, and Expected Departure/Arrival Time, Status, Special Request Code (if any) and Boarding Gate (if any))
- **Validations (and feedback)**
 - The program should handle all invalid entries by the user, including empty responses and unexpected input datatypes (e.g. invalid 2-Letter Airline Code, invalid Flight Number, invalid Boarding Gate, etc.)
 - If the user makes a mistake in the entry, the program should inform the user via appropriate feedback

IMPORTANT INSTRUCTIONS:

- One student is required to implement features 2, 3, 5, 6 & 9, and another is required to implement features 1, 4, 7 & 8.
- An Individual student without a team is required to implement features 1, 2, 3, 4, 6 & 7.

4. ADVANCED FEATURES – 20% INDIVIDUAL

When implementing the Advanced Features below, you are free to add onto but NOT MODIFY the Class Diagram where appropriate to complete them (i.e. changing the Class Diagram for Basic Features is NOT ALLOWED).

(a) Process all unassigned flights to boarding gates in bulk

- *for each Flight, check if a Boarding Gate is assigned; if there is none, add it to a queue*
- *display the total number of Flights that do not have any Boarding Gate assigned yet*
- *for each Boarding Gate, check if a Flight Number has been assigned*
- *display the total number of Boarding Gates that do not have a Flight Number assigned yet*
- *for each Flight in the queue, dequeue the first Flight in the queue*
 - *check if the Flight has a Special Request Code*
 - *if yes, search for an unassigned Boarding Gate that matches the Special Request Code*
 - *if no, search for an unassigned Boarding Gate that has no Special Request Code*
 - *assign the Boarding Gate to the Flight Number*
 - *display the Flight details with Basic Information of all Flights, which are the 5 flight specifications (i.e. Flight Number, Airline Name, Origin, Destination, and Expected Departure/Arrival Time), Special Request Code (if any) and Boarding Gate*
- *display the total number of Flights and Boarding Gates processed and assigned*
- *display the total number of Flights and Boarding Gates that were processed automatically over those that were already assigned as a percentage*

(b) Display the total fee per airline for the day

- *check that all Flights have been assigned Boarding Gates; if there are Flights that have not been assigned, display a message for the user to ensure that all unassigned Flights have their Boarding Gates assigned before running this feature again*
- *for each Airline, retrieve all their Flights*
 - *for each Flight*
 - *check if the Origin or Destination is Singapore (SIN), and apply the respective fee of \$800 or \$500 accordingly*
 - *check if the Flight has indicated a Special Request Code and charge the appropriately listed Additional Fee*
 - *apply the Boarding Gate Base Fee of \$300*
 - *compute the subtotal of fees to be charged for each Airline for the day*
 - *compute the subtotal of discounts to be applied for each Airline based on the Promotional Conditions that they qualify for*
 - *display the total final fees to be charged with a breakdown of the original subtotal calculated against the subtotal of discounts for the day*
- *compute and display the subtotal of all the Airline fees to be charged, the subtotal of all Airline discounts to be deducted, the final total of Airline fees that Terminal 5 will collect, and the percentage of the subtotal discounts over the final total of fees*

(c) Recommend an additional feature to be implemented (bonus marks are only awarded if the advanced feature is completed)

- *you may gain up to 5 bonus marks if you propose and successfully implement an additional feature per student. **You are REQUIRED to check with your tutor about your idea before implementing the advanced feature.***

IMPORTANT INSTRUCTIONS:

- *A student is to implement feature (a), and another implements feature (b).*
- *An Individual student without a team can choose to implement either one of the features.*
- *Please note that you should implement the advanced features only AFTER all the basic features have been fully implemented and working.*
- *NO MARKS will be awarded for the advanced features if the basic features have NOT been fully implemented and working.*
- *Marks will be deducted if you are not able to show your understanding of the program, both basic and advanced features (if applicable), during the presentation.*

5. ACTIVITY PLAN

Suggestions for Getting Started

There are many ways that you could complete this assignment. It is most important part to think about the entire assignment first so that it is easy to integrate the various parts.

a) Analysis

1. Understand the program specification and the requirements before attempting the assignment.
*e.g. the relationships between the classes
the use of the attributes in each class*

b) Program Design

2. Work out the User Interface needed to get the user input for suitable output.
3. Work out the main logic of the program using Object-Oriented programming techniques;
i.e. use the inheritance and the association of the classes properly.
4. You are required to use suitable classes and methods appropriately for this assignment.
Marks will be deducted for inefficient use of the classes/methods or improper use of classes/methods

c) Implementation & Testing

5. Determine the order in which the classes are to be implemented. (Certain classes need to be implemented before other classes can be implemented)
6. Implement the classes **ONE** at a time.
7. Test your program logic to make sure that it works.
You must prepare test data to see that your program works correctly. All data entry should be validated and illegal data entry should be highlighted to the user so that the user can enter correct data.

6. DELIVERABLES

You are to develop the project using the Console App. You should name your project as **Snnnnnnnn_PRG2Assignment** (note: **Snnnnnnnn** is your Student Number).

You are required to submit your work in **TWO** stages:

Stage 1 – Due date: Sunday, Week 15 (26 January 2025) @ 23:59 hours

- ALL the 9 classes (source files) shown in the class diagram and at least **1** basic feature to your **Brightspace Stage 1 Submission**.

In **EACH** of your source files, you **MUST** include a block comment at the top stating your **student number**, **student name**, and **partner name** as shown below:

```
//=====
// Student Number : S12345678
// Student Name   : Michael Jordan
// Partner Name   : Scottie Pippen
//=====
```

Note: If a student does not submit his/her stage 1 work by the deadline, 20 marks will be deducted.

Stage 2 – Due date: Sunday, Week 16 (2 February 2025) @ 23:59 hours

- ALL the classes (source files) that you have written for the whole assignment to your **Brightspace Stage 2 Submission**.

In **EACH** of your source files, you **MUST** include a block comment at the top stating your **student number**, **student name**, and **partner name** as shown below:

```
//=====
// Student Number : S12345678
// Student Name   : Michael Jordan
// Partner Name   : Scottie Pippen
//=====
```

Presentation – During PRG2 lessons

- You are to present your application to your tutor during the PRG2 classes right after the submission deadline.

7. GRADING CRITERIA

This assignment constitutes 30% of this module.

Performance Criteria for grading the assignment is as described below. Marks awarded will be based on **program code** as well as student's degree of understanding of work done as assessed during the **presentation**.

Grading criteria for the program is given below.

A Grade

- ◆ Program implements the *Basic Features* successfully and efficiently
- ◆ Program implements all the basic *input validations* successfully
- ◆ Program implements *Advanced Features* successfully and efficiently
- ◆ Program demonstrates good design with the correct use of methods
- ◆ Program provides strong evidence of good programming practice
- ◆ Program has been tested adequately

B Grade

- ◆ Program implements the *Basic Features* successfully
- ◆ Program implements some basic *input validations* successfully
- ◆ Program attempts to use methods
- ◆ Program provides sufficient evidence of good programming practice
- ◆ Program has been tested adequately

C Grade

- ◆ Program implements the *Basic Features* successfully
- ◆ Program provides some evidence of good programming practice
- ◆ Program has been tested adequately

D Grade

- ◆ Program implements the *Basic Features* successfully
- ◆ Program has been tested adequately
- ◆ Presentation is awarded at least a 'D' grade

NOTE

- *Evidence of good programming practice include appropriate use of methods, the use of meaningful variable names, proper indentation of code, appropriate and useful comments, adoption of standard naming conventions etc.*
- *Basic Input validation refers to the checking of the inputs entered by the user. e.g. invalid option, invalid date*