

## Phase # 3

### Objectives:

1. Create Classes and Objects.
2. Process Array of objects.
3. To apply class abstraction to develop software.
4. To discover the relationships between classes.
5. To design programs using the object-oriented paradigm.
6. To define a subclass from a superclass through inheritance.
7. To design and use abstract classes.
8. To store, retrieve, and manipulate objects in an **ArrayList**.

### Specification

Submission: **Online through ITC.**

What to submit: Your **own** well-structured and well-commented JAVA files (.java) and PDF file

Deadline: **20/8/2021** by 8pm. (The online submission will be disabled after this time).

### Important

- Academic **honesty**:
  - This is an individual assignment. Individual assignments must be each student's own work.
  - Copying one line from a friend or the internet will be considered cheating.
  - Cheating will result in an official university disciplinary review and the University regulations will be strictly enforced.

### Task: A320 Flight Trip Reservation System –enhanced version-

Consider the following trip related objects details:

An abstract base **Person** object has the following attributes:

- First name
- Last name
- Gender (M/F)
- Date of birth

A **Passenger** object, who is a **Person**, has the following extra attributes:

- Passport number
- Nationality
- Has a valid entrance VISA
- Has flight seat preference which could be *Window*, *Aisle*(ممر), or *None*

The A320 flight has rows of seats. The abstract base **Seat** object has the following attributes:

- The row number (from 1-31)
- The column number (from A to F)
- The seat number which consists of column number and row number (e.g. A12)
- The Passenger details if the seat is reserved.

The flight seats are two types, first class seats and economy seats. The **First Class Seat** has the following extra attributes:

- Food preferences (either **Caviar** or **Steak**)
- Drink preferences (either **Liquorice** or **Carob**)

While the **Economy Seat** has the following extra attributes:

- Food preferences (either **Chicken** or **Fish**)
- Drink preferences (either **Cola** or **Juice**)

Finally, we have a flight **Trip** that has the following attributes:

- A ragged array of seats.
- Flight number.
- From airport name.
- To airport name
- The trip distance
- The departure time
- The arrival time

### Task1: UML diagram

Draw the UML diagram for all the classes as mentioned above and clearly show the relationship between those classes. (Use any free online tool to draw UML and save it as a PDF file. e.g. <https://www.draw.io/>)

### Task2: Class Implementations

Implement the classes according to the mentioned attributes and relationship between classes. For the constructor and behaviours:

- Setters and getters to the class attributes as necessary.
- Constructors using the instance attributes as necessary.

### Task3: A320 Flight Trip Reservation System

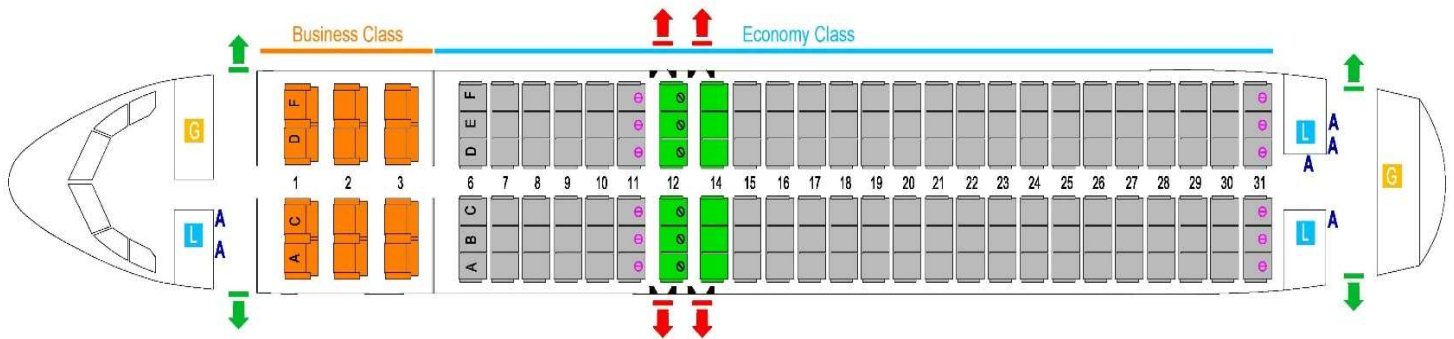
Create a driver class and in the main method, loop to show the following menu:

- 0) To add a new trip. Use **ArrayList** to add the trips.
- 1) To reserve a new empty seat suggested by the system (first class or economy) for a new passenger into a specific trip taking into consideration the passenger's seat preference.

- 2) To display the passenger information for a specific seat (if the seat is not empty) in a specific trip.
- 3) To search for a passenger by last name in a specific trip.
- 4) To print a list of the first class seat's passengers and the economy seat's passengers sorted by seat numbers in a specific trip.
- 5) To cancel a passenger reservation in a specific trip using passenger's last name.
- 7) To exit execution

Finally, adequately implement the previous options (from 0 to 7)

The flowing figure shows the A320 flight seat Configuration for your reference:



Good Luck!