

Coding Exercise

Description

- For the purposes of this exercise, it's expected to implement a console application driven from a `main` method, the application does not need to be interactive
- It's expected that you use C# or Java for this exercise
- This exercise is meant to allow you to display your skills and experience, the use of best practices and object-oriented design principles such as SOLID is expected
- The use of a database is not required
- It is expected that the take-home part of this should not take more than 1.5 hours to complete
- Use the included orders file for the list of orders
- If you move on to the next phase of the interview, it will be expected that you bring this code in for a peer exercise where you'll be expected to work on new User Stories iterating on this code.

Scenario

Transport.ly is a brand-new company that aims to provide efficient and fast air freight services; they currently have 3 planes the planes are scheduled to fly daily at noon. For this exercise, there are only 2 days of flights scheduled.

Day 1:

Flight 1: Montreal airport (YUL) to Toronto (YYZ)

Flight 2: Montreal (YUL) to Calgary (YYC)

Flight 3: Montreal (YUL) to Vancouver (YVR)

Day 2:

Flight 4: Montreal airport (YUL) to Toronto (YYZ)

Flight 5: Montreal (YUL) to Calgary (YYC)

Flight 6: Montreal (YUL) to Vancouver (YVR)

With each flight returning to the YUL at midnight.

Each plane has a capacity of 20 boxes each.

The company's sales department has been able to sell 99 orders that are sending boxes to Toronto, Calgary, and Vancouver, these orders are found in the attached json file. Each box represents 1 order.

As a member of the software engineering department you are asked to develop an application that can automate the process of determining which boxes to load on each flight.

User Stories - Take-Home

USER STORY #1

As an inventory management user, I can load a flight schedule similar to the one listed in the **Scenario** above. For this story you do not yet need to load the orders. I can also list out the loaded flight schedule on the console.

Expected output:

```
Flight: 1, departure: YUL, arrival: YYZ, day: 1
```

```
...
```

```
Flight: 6, departure: <departure_city>, arrival: <arrival_city>, day: x
```

USER STORY #2

As an inventory management user, I can generate flight itineraries by scheduling a batch of orders. These flights can be used to determine shipping capacity.

- Use the json file attached to load the given orders.
- The orders listed in the json file are listed in priority order ie. 1..N

Expected output:

```
order: order-001, flightNumber: 1, departure: <departure_city>, arrival: <arrival_city>, day: x
```

```
...
```

```
order: order-099, flightNumber: 1, departure: <departure_city>, arrival: <arrival_city>, day: x
```

if an order has not yet been scheduled, output:

```
order: order-X, flightNumber: not scheduled
```