

# Airport record system

## Assignment brief:

### Time 3 hr

You have been given some code that is currently being developed as part of an **Airport** record application. Not all the requirements have been implemented. It is your task to implement these and raise the coding standards of all the code.

Create a **project solution** (named **<Your Name><Student Number>p3** e.g. **AidanMcGowan3048614p3**). Create a package named **p3**. Add **StartApp.java** to the solution and the **airport\_data.csv**. Ensure **your name and student number** are placed in the Javadoc comments of all the classes you create. The StartApp has been partially written with a menu.

The application will run (start) from the StartApp.java, initially reading in the data from the **airport\_data.csv** file and then perform a number of menu driven operations.

## Part 1 – Data mapping, storage and read from file - 50%

Using your knowledge of OOP you should add/update the code based on the following:

1. Analyse the data in the **airport\_data.csv** and create a class (**Airport.java**)
2. Conduct a simple unit test for the **Airport** class. No other validation or business rules required so simple verification of fields being set by setters and constructors sufficient
3. In the **StartApp.java** class read and store the data (**airport\_data.csv**) in an appropriate JCF container. Some of the fields for some records may be empty, null or not have the expected data types. These records should not be included in the list of Airports in the JCF container.

include the following re-codes...

- I. **Airport Code** set to uppercase and only the first three characters. E.g. **BFsX** should be **BFS**
- II. **Airport type should be recoded as** 1 = COMMERCIAL, 2 = MILITARY and 3 = UNKNOWN
- III. **The country code IRE** should be recorded as **IRL**

**[CONTINUED OVER]**

## Part 2 – Functions – 50%

Having read the data from the csv file complete the menu driven functions as outlined below. An example of the expected format is shown for each function. Note if you were unable to recode the input data as specified in Part 1 use the uncoded raw data (as appropriate).

1. Display all airports to screen. Example output...

Enter option ...

1

All airports

```
Name      : Belfast International Airport
City      : NI
Country   : UK
Code      : BFS
Alt       : 268
Type      : COMMERCIAL
```

```
Name      : George Best Belfast City Airport
City      : NI
Country   : UK
Code      : BHD
Alt       : 15
Type      : COMMERCIAL
etc...
```

2. Display all airports in IRL. Example output...

Enter option ...

2

All airports in IRL

Cork Airport ORK

Galway Airport GWY

Dublin Airport DUB

Ireland West Knock Airport NOC

etc...

3. Display the airport with the highest altitude. Example output...

*e.g. Newcastle Aerodrome UK 234*

(note: not actual answer based on csv data provided)

4. Display each region (in alphabetical order) with total number of airports in the region. Example output...

Enter option ...

4

CONNAUGHT : 6

ENG : 88

LEINSTER : 4

etc...

**[CONTINUED OVER]**

5. Heliports are identifiable from “H” as the last character in their Airport Code. e.g London Biggin Hill Airport BQH.
- i.) Display each **COMMERCIAL Heliport** (order in descending altitude shown below) and
- ii.) Using **RECURSION** re-order to display each **COMMERCIAL Heliport** (order in ascending altitude shown below.)

Example output...

Enter option ...

5

Helipads sorted by alt (DESCENDING)

London Biggin Hill Airport BQH 598

Plymouth City Airport PLH 476

Bournemouth Airport BOH 38

Tresco Heliport TRH 20

London Heliport LOH 18

Penzance Heliport PZH 14

Shoreham Airport ESH 7

Newcastle Aerodrome NEH 1

Glasgow City Heliport GCH 0

Helipads sorted by alt (ASCENDING) \*RECURSIVE CALL

Glasgow City Heliport GCH 0

Newcastle Aerodrome NEH 1

Shoreham Airport ESH 7

Penzance Heliport PZH 14

London Heliport LOH 18

Tresco Heliport TRH 20

Bournemouth Airport BOH 38

Plymouth City Airport PLH 476

London Biggin Hill Airport BQH 598

**[CONTINUED OVER]**

6. In a new **Thread** export/write to a new file (*airport\_data\_recoded.csv*) in the format for each record of

**name, new\_code**

**name** : is the capitalisation of the Airport Name

**new\_code** : a new code for each Airport consisting of the current Code, Country code, Altitude and a Parity bit based on Altitude, where if the Altitude is **even** add a **0** and if **odd** add a **1**.  
(Example shown below). Note, include the header as shown. Example csv output ...

name, new\_code

BELFAST INTERNATIONAL AIRPORT,BFSUK2680

GEORGE BEST BELFAST CITY AIRPORT,BHDUK151

etc...

When complete compress (zip) the entire ***Eclipse solution*** and upload to **Assignments** (P3 assessment) on CANVAS.

**Now : check the uploads to ensure you have submitted the correct files (in the correct area).**

**[END]**