Aircraft Tracking System

Assignment brief:

Time 2hr 30 min (open book – no external websites).

Create a **project solution** (named **<your name and student id> e.g. AidanMcGowan6048201). Ensure your name and student number are placed in the Javadoc comments of all the classes / Test classes etc you create.**

Part 1 – 40%

You have been tasked to design, implement, and unit test part of an Aircraft Flight tracking application.

The system is expected to support several specific **Aircraft** such as **Passenger plane**, **Helicopter**, **Cargo Plane**, **Private** and **Military**. You are the first developer of the system and are tasked with designing and implementing the **Passenger Plane**, although you are encouraged to design the system to support the future development of other specific aircraft.

Data required for the system.

	Description	*Application*	*Business rules*
type	The **type** of the Aircraft	All Aircraft	Text: min length 3 and max 15 E.g. Sikorsky s20, Boeing 777 etc.
wingspan	The **wingspan** of the aircraft	All aircraft	Whole number: range 25 and 50 (inclusive)
power	**Power** category	All aircraft	Allowable values **Fixed**, **Prop**, **Rotor**
country	**Country code** : two-letter code representing where the Passenger Plane is registered.	Passenger planes only	Allowable codes UK, FR and US only.

Additional methods:

showAll()

you have been asked to create a **showAll** method that will output to screen (console) all data for each Aircraft instance. *Note you do not need to Unit Test this functionality*.

For example, the output for a Passenger Plane would be:

```
TYPE         : Boeing 777
WINGSPAN     : 34
POWER        : FIXED
COUNTRY      : FR
```

ping()

It is a requirement of all Aircraft i.e. Passenger Plane, Helicopter, Cargo Plane, Private and Military etc to have a method to be named **ping**. However, each of the Aircraft will implement the method differently. The method takes no parameter arguments but generates and returns a string.

The implementation of the ping method for the **Passenger Plane** will return a string in the format **PP-CODE-POWER-WINGSPAN-TYPE** e.g. for a passenger plane from France (FR country code) PP-7082-FIXED-34-Boeing 777. Note the hyphen between each part. The **CODE** is generated from the ASCII value for the two letters in the country code, e.g. FR=**7082** i.e. F = **70** and R= **82**.

Part 2 - Aircraft Search class - 20%

Create an ***AircraftSearch*** class to support the system for searching. Each search method should be ***static*** and accept and return an ***ArrayList*** of the **appropriate type**. Using your knowledge of OOP you should create the following functionality based on the following:

1. Create a **searchByWingSpan** method i.e. search for all objects in the parameter argument ArrayList that have a wingspan *within a specified lower and upper range* e.g. between 30 and 35 (range inclusive of both values). You should return an ArrayList containing any that satisfy the search criteria.

2. Create a **searchByCountry** method i.e. search for all objects in the parameter argument ArrayList that match a specified Country (eg FR, UK, US). You should return an ArrayList containing any that satisfy the search criteria.

Part 3 – Testing – 40%

1. Unit Test the application.

When complete compress (zip) the entire ***Eclipse solution*** and upload to **Assignments** (P2 assessment) on CANVAS.

Now : check the uploads to ensure you have submitted the correct files.

[END]