**Sample CA(3)**

In this exercise you are required to write a set of classes to store passenger details about a flight.

Read this document through in full before you start coding the solution in Java.

The requirements for each of the classes you need to create are documented below.

**Passenger class**

The Passenger class has 4 members: name (String), age (int), weight of bags (double array), size of bags (char array)

Create a suitable constructor to initialise all 4 members using a parameter list.

Create suitable getter and setter methods.

Create a calcBaggageCharge() method which calculates the charges for a passenger’s **combined** set of luggage using the information provided below:

There are two sizes of Bag – medium which has a weight of 15kg and large which has a weight of 20kg.

The charge for a medium bag is €30.

The charge for a large bag is €40.

Any overweight luggage for a passenger is charged at the additional rate of €10 for every 1kg over.

**Flight class**

The Flight class has 3 member variables:

* capacity (int) and duration (double) of the flight
* a list of passengers (array of type Passenger)
* an int variable i set to 0

Create a suitable constructor to:

* initialise the capacity and duration using a parameter list
* initialise the array to size 4

Create a void method fillList() that takes a reference variable of type Passenger as a parameter, adds it to the array and increments i.

Create a void print() method that prints the name, age and baggage charge for each passenger. Here you will need to call the calcBaggageCharge method in the Passenger class.

Create a method checkAvailability() that calculates and returns the number of seats (int) still available on the flight.

Create a method calcOldestPassenger() that calculates and returns the name and age of the oldest passenger as a string.

**Test Class**

Create a Flight object called **f** with a capacity of 200 and a duration of 3.5 hours.

The data for the passengers is shown below:

john,52,12,16,22,m,m,l

ken,26,12,13,14,m,m,m

kate,89,22,23,24,l,l,l

Peter,40,16,18,23,m,m,l

Create a folder called **files** inside your IntelliJ project. Inside this folder, create a file called **data.txt** and copy the data shown above into this file.

Write the Java code to read the data from this file and use the data to create Passenger objects. Include appropriate exception handling code for the file reading.

Add each Passenger object to the array using the fillList() method.

The passenger details from the file are given below. For Bag Size shown in the table below, m represents Medium and l represents Large

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Passenger Name | Passenger Age | Bag 1 Weight | Bag 2 Weight | Bag 3 Weight | Bag 1 Size | Bag 2 Size | Bag 3 Size |
| John | 52 | 12 | 16 | 22 | m | m | l |
| Ken | 26 | 12 | 13 | 14 | m | m | m |
| Kate | 89 | 22 | 23 | 24 | l | l | l |
| Peter | 40 | 16 | 18 | 23 | m | m | l |

Call the relevant methods to display the details as shown in the sample output below:

**Sample Output**

Passenger List for Flight

john 52 Baggage Charge:€100.00

ken 26 Baggage Charge:€90.00

kate 89 Baggage Charge:€210.00

Peter 40 Baggage Charge:€170.00

Number seats available:196

Oldest Person on the flight is:kate age:89