Car Pooling System - Requirement 1

You are a very active member of a Nature Club in your organization. In one of the meetings, it was discussed to build a car pooling system to help cut down the pollution. Being very active and tech savvy, you wish to contribute towards the development of system. One of the members being an architect has understood the requirement and would be sharing you with smaller requirements.

**Requirement 1:**

The users of the system are going to be general public who own cars. They are classified as members in our system.

1. Create a Member Class with the following attributes:

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| \_id | long |
| \_firstName | string |
| \_lastName | string |
| \_email | string |
| \_contactNumber | string |
| \_license Number | string |
| \_licenseStartDate | DateTime |
| \_licenseExpiryDate | DateTime |

1. Mark all the attributes as private
2. Create / Generate appropriate Properties
3. Add a default constructor and a parameterized constructor to take in all attributes.
4. When the “member” object is printed, it should display the following details:

Member: \_firstname, \_lastname

Member contact details: \_contactNumber, \_email

1. Two members are considered same if they have same email and contactNumber. Implement the logic in the appropriate function. (Case – Insensitive)

**Sample Input and Output 1:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**

Member1 :  
id:  
**1**  
first name:  
**Arun**  
last name:  
**Kumar**  
email:  
**arun123@gmail.com**  
contact number:  
**9878767655**  
license number:  
**TN38QW1232343**  
license start date:  
**12-12-2010**  
license expiry date:  
**13-12-2020**  
Member2 :  
id:  
**2**  
first name:  
**Mohamed**  
last name:  
**Safiq**  
email:  
**safiq1243@gmail.com**  
contact number:  
**9667826601**  
license number:  
**TN33VA1238743**  
license start date:  
**01-05-2013**  
license expiry date:  
**01-04-2125**  
  
Member 1  
Name: Arun , Kumar  
Member contact details: 9878767655 , arun123@gmail.com  
  
Member 2  
Name: Mohamed , Safiq  
Member contact details: 9667826601 , safiq1243@gmail.com  
Member 1 and Member 2 are different  
  
**Sample Input and Output 2:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Member1 :  
id:  
**1**  
first name:  
**Sam**  
last name:  
**Nath**  
email:  
**Sam123@gmail.com**  
contact number:  
**9456738498**  
license number:  
**TN45AS123456**  
license start date:  
**12-12-2010**  
license expiry date:  
**13-12-2021**  
Member2 :  
id:  
**2**  
first name:  
**Swamy**  
last name:  
**Nathan**  
email:  
**Sam123@gmail.com**  
contact number:  
**9456738498**  
license number:  
**TN54DF321456**  
license start date:  
**01-05-2012**  
license expiry date:  
**01-05-2123**  
  
Member 1  
Name: Sam , Nath  
Member contact details: 9456738498 , Sam123@gmail.com  
  
Member 2  
Name: Swamy , Nathan  
Member contact details: 9456738498 , Sam123@gmail.com  
Member 1 is same as Member 2

**Car Pooling System - Requirement 2**

You are a very active member of a Nature Club in your organization. In one of the meetings, it was discussed to build a car pooling system to help cut down the pollution. Being very active and tech savvy, you wish to contribute towards the development of system. One of the members being an architect has understood the requirement and would be sharing you with smaller requirements.

**Requirement 2:**

The next most important entity of the system is Car.  
  
 a)   Create a Car Class with the following attributes:

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| \_id | long |
| \_name | string |
| \_model | string |
| \_makeYear | Int32 |
| \_company | string |
| \_comfortLevel | Int32 |

b)    Mark all the attributes as private  
  
 c)   Create / Generate appropriate Properties  
  
 d)   Add a default constructor and a parameterized constructor to take in all attributes.  
  
 e)   One Member can pool one or many cars that he / she owns. To accommodate this, Create a MemberCar class with following attributes

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| \_id | long |
| \_member | Member |
| \_car | car |
| \_carRegistrationNumber | string |
| \_carColor | string |

Create / generate appropriate Properties and constructors.  
  
f)    Add a static FindMember method in Member class which takes id as input and returns the Member or null object if the member object is not found.  
  
 g)   Add a static FindCar method in Car class which takes id as input and returns the Car or null object if the car object is not found.  
  
 h)   Introduce an arraylist (carList) in the member class which holds the list of member-car objects. Include appropriate Properties. Given a member, Display the number of cars the member owns along with registration number of each car separated by a line.  
  
**Input Output Format:**  
Refer Sample I/O for formatting specifications.  
  
  
**Sample Input and Output:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Menu:  
 1) Add a Member  
 2) Add a Car  
 3) Assign Car to Member (Assume valid data is supplied)  
 4) Cars Owned  
 5) Exit  
**1**  
id:  
**6**  
first name:  
**arun**  
last name:  
**kumar**  
email:  
**arun@gmail.com**  
contact number:  
**9089786756**  
license number:  
**TN12ER3423**  
license start date:  
**13-12-1998**  
license expiry date:  
**12-12-2008**  
Menu:  
 1) Add a Member  
 2) Add a Car  
 3) Assign Car to Member (Assume valid data is supplied)  
 4) Cars Owned  
 5) Exit  
**2**  
id:  
**5**  
name:  
**Verna**  
model:  
**CRDI**  
makeYear:  
**2008**  
company:  
**Hyundai**  
comfort level:  
**8**  
Menu:  
 1) Add a Member  
 2) Add a Car  
 3) Assign Car to Member (Assume valid data is supplied)  
 4) Cars Owned  
 5) Exit  
**3**  
member car id  
**12**  
member id  
**3**  
car id  
**5**  
car registration  
**TN38BR9689**  
color  
**White**  
Menu:  
 1) Add a Member  
 2) Add a Car  
 3) Assign Car to Member (Assume valid data is supplied)  
 4) Cars Owned  
 5) Exit  
**3**  
member car id  
**13**  
member id  
**6**  
car id  
**2**  
car registration  
**TN66AB4214**  
color  
**Brown**  
Menu:  
 1) Add a Member  
 2) Add a Car  
 3) Assign Car to Member (Assume valid data is supplied)  
 4) Cars Owned  
 5) Exit  
**4**  
member id  
**6**  
Number of cars : 1  
Registration Numbers :  
TN66AB4214  
Menu:  
 1) Add a Member  
 2) Add a Car  
 3) Assign Car to Member (Assume valid data is supplied)  
 4) Cars Owned  
 5) Exit  
**5**

**Car Pooling System - Requirement 3**

You are a very active member of a Nature Club in your organization. In one of the meetings, it was discussed to build a car pooling system to help cut down the pollution. Being very active and tech savvy, you wish to contribute towards the development of system. One of the members being an architect has understood the requirement and would be sharing you with smaller requirements.

**Requirement 3:**

There are always typical human entry errors that need to be validated so that the data being saved in the system is valid and can be used for later processing. Simple Rules that needs to be taken care:  
  
 a)    Validate Car registration pattern.  A typical registration pattern would be  
                2 chars-2 numbers – 2 chars – 4 numbers [TN-07-AS-1273]. Assume all characters are upper case.  
  
 b)   Convert it to appropriate registration pattern –  
During entry, some of them have used different alpha numeric characters as separator and the case sensitivity of characters have been left out. Ensure these two are fixed and the format given in (sub question) “a” is returned as output. Assume valid inputs are provided.  
  
 c)   Check for validity of driving license with respect to date – Every Driving license is valid for 10 years. Assume today’s date as 15-06-2017. Check if the given driving license date is valid.  
  
**Input Output Format:**  
  
Refer Sample I/O for formatting specifications.  
  
**Sample I/O 1:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Menu:  
1) Valid Car registration Number  
2) Convert Car registration Number  
3) Valid Driving License  
Enter choice  
**1**  
car registration number  
**TN38BR9689**  
TN38BR9689 is Invalid  
  
**Sample I/O 2:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Menu:  
1) Valid Car registration Number  
2) Convert Car registration Number  
3) Valid Driving License  
Enter choice  
**1**  
car registration number  
**TN-38-BR-9689**  
TN-38-BR-9689 is Valid  
  
**Sample I/O 3:  
[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Menu:  
1) Valid Car registration Number  
2) Convert Car registration Number  
3) Valid Driving License  
Enter choice  
**2**  
car registration number  
**TN@38\*BR@9689**  
TN-38-BR-9689  
  
**Sample I/O 4:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Menu:  
1) Valid Car registration Number  
2) Convert Car registration Number  
3) Valid Driving License  
Enter choice  
**3**  
driving license issue date  
**29-12-1990**  
26 years old license - expired  
  
**Sample I/O 5:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Menu:  
1) Valid Car registration Number  
2) Convert Car registration Number  
3) Valid Driving License  
Enter choice  
**3**  
driving license issue date  
**29-12-2010**  
6 years old license – valid

**Car Pooling System - Requirement 4**

You are a very active member of a Nature Club in your organization. In one of the meetings, it was discussed to build a car pooling system to help cut down the pollution. Being very active and tech savvy, you wish to contribute towards the development of system. One of the members being an architect has understood the requirement and would be sharing you with smaller requirements.

**Requirement 4:**

Very soon you discover that there are different classes of Cars that can be pooled. Each class of cars does share some common properties as well as some specific properties. Represent the above requirement in Object Orientation. Mark the base class as abstract.  
  
 a.   Use the Car class created in Requirement 2. Retain attributes \_id and \_name. Lets’ ignore rest of the attributes for this requirement. Make the appropriate access modifier change to enable the attributes to be accessible to child classes.  
  
**Car**

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| \_id | long |
| \_name | string |

  b.   Create 3 child classes with Car as base class and the following attributes.  
  
**HatchBack**

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| \_powerWindowsEnabled | bool |
| \_automaticGear | bool |

**Sedan**

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| \_absEnabled | bool |
| \_bootSpace | Int32 |

**UtilityCar**

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| \_rearCoolingVents | bool |

 c.   Mark the access modifiers appropriately, and create constructors using super keyword.  
  
d.    Create an abstract method CalculateDriveCost() which takes a km covered as double and returns the cost in rupees as double. (rounded to 0 decimal places)  
  
**The formula for computing it as follows:**

|  |  |
| --- | --- |
| **Rule** | **Cost (Rs)** |
| Hatchback without automatic gear | 10 |
| Hatchback with automatic gear | 12 |
| Sedan | 15 |
| Sedan with bootspace > 600 | 15 and 20% additional cost |
| Utility | 18 |

**Input Output Format:**  
  
Refer Sample I/O for formatting specifications.  
  
  
**Sample I/O 1:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
id  
**1**  
name  
**i20**  
type  
**sedan**  
distance  
**20**  
abs enabled  
**false**  
boot space  
**350**  
Cost is Rs 300  
  
**Sample I/O 2:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
id  
**2**  
name  
**Verna**  
type  
**hatchback**  
distance  
**10**  
power windows  
**true**  
automatic  
**false**  
Cost is Rs 100  
  
**Sample I/O 3:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
id  
**3**  
name  
**Mahindra SUV**  
type  
**utility**  
distance  
**10**  
rear cooling vents  
**false**  
Cost is Rs 180

**Car Pooling System - Requirement 5**

You are a very active member of a Nature Club in your organization. In one of the meetings, it was discussed to build a car pooling system to help cut down the pollution. Being very active and tech savvy, you wish to contribute towards the development of system. One of the members being an architect has understood the requirement and would be sharing you with smaller requirements.

**Requirement 5:**

Create a custom checked Exception “InvalidEmailException” with custom error message. "InvalidEmailException: Invalid Email for the user".  
Read member details from Input (Console). One line would consist details relating one member in comma separated format. Create member objects for each line and add it to an arraylist.  
a.      During the parse, if an email id is invalid, A valid email has an @ and ends with “.com / .org”. Raise the custom exception and skip the entity to add into the list.  
b.      Sort the arraylist of members based on firstname ( hint: Use comparable).  
  
**Sample Input and Output:  
[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Enter the number of Members:  
**5**  
Enter the member details:  
**1,joe,root,joe.root@a.com,1234567890,AH1,12-12-2001,12-12-2010  
2,ben,stokes,ben.stokes!a.com,2345678901,AH2,12-12-2002,12-12-2011**  
InvalidEmailException: Invalid Email for ben  
**3,virat,kohli,virat.kohli@a.com,3456789012,AH3,12-12-2003,12-12-2012  
4,ravi,varma,ravi.varma@a.in,4357689010,AH4,11-10-2003,12-10-2013**  
InvalidEmailException: Invalid Email for ravi  
**5,mahendra,bagubali,mahi.bali@abc.com,1234798012,AH5,10-02-2003,12-02-2015**  
  
The member details are:  
1,joe,root,joe.root@a.com,1234567890  
5,mahendra,bagubali,mahi.bali@abc.com,1234798012  
3,virat,kohli,virat.kohli@a.com,3456789012

**Car Pooling System - Requirement 6**

You are a very active member of a Nature Club in your organization. In one of the meetings, it was discussed to build a car pooling system to help cut down the pollution. Being very active and tech savvy, you wish to contribute towards the development of system. One of the members being an architect has understood the requirement and would be sharing you with smaller requirements.

**Requirement 6:**

You need to migrate the existing data into your newly designed system, write a utility program that reads memberCar details in the given format and builds an arraylist.  
Assumption: In your template code, the member and car details would be prefilled to you. You can assume only those members and cars would be used for evaluation.  
Currently you have three arraylists, one of type member objects (prefilled in template code), one of type car objects (prefilled in template code) and the membercar arraylist constructed by you.  
-          Write a function GroupByColor which takes all three arraylists and returns a dictionary with car color as key and list of MenberCars as value.  
  
**Pre-filled member details:**  
1,joe,root,joe.root@a.com,1234567890,AH1,12-12-2001,12-12-2010  
2,ben,stokes,ben.stokes@a.com,2345678901,AH2,12-12-2002,12-12-2011  
3,virat,kohli,virat.kohli@a.com,3456789012,AH3,12-12-2003,12-12-2012  
  
**Pre-filled car details:**  
1,i10,sports,2007,Hyundai,8  
2,alto,kx1,2008,Maruti,6  
3,polo,topline,2010,Volks,5  
4,kwid,lxi,2010,Renault,5  
  
  
**Sample Input and Output:**  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
Color to search  
**white**  
Number of member cars  
**3**  
Enter the member car details  
**1,1,2,TN66AB4214,brown  
2,1,1,TN38BR9689,white  
3,2,4,TN61EB4004,white**  
white car registration numbers:  
TN38BR9689  
TN61EB4004