OOP Example 01

Department of Motor Traffic issues Vehicle license for each vehicle. Vehicle owner can revenue their vehicle license from any Assistant Government Agents (AGA) office. You are required to develop a Java program to revenue vehicle licenses for motor coaches, three-wheelers and motor cycles. The following list gives criteria for calculate license fee per year.

- Motor Cycles Rs. 700/=
- Three-Wheelers (Diesel) Rs. 850/=
- Three-Wheelers (Petrol) Rs. 950/=
- Private Motor Coaches Rs. (Diesel) 500/= Per Passenger Seat
- Private Motor Coaches Rs. (Petrol) 400/= Per Passenger Seat

You program should store owner name, address, registration date and vehicle number for the each vehicle. Fuel type is recorded for the Three-Wheelers and the number of passenger seats and fuel type has been recorded for the private motor coaches. Figure 1 shows the class diagram of the proposed system

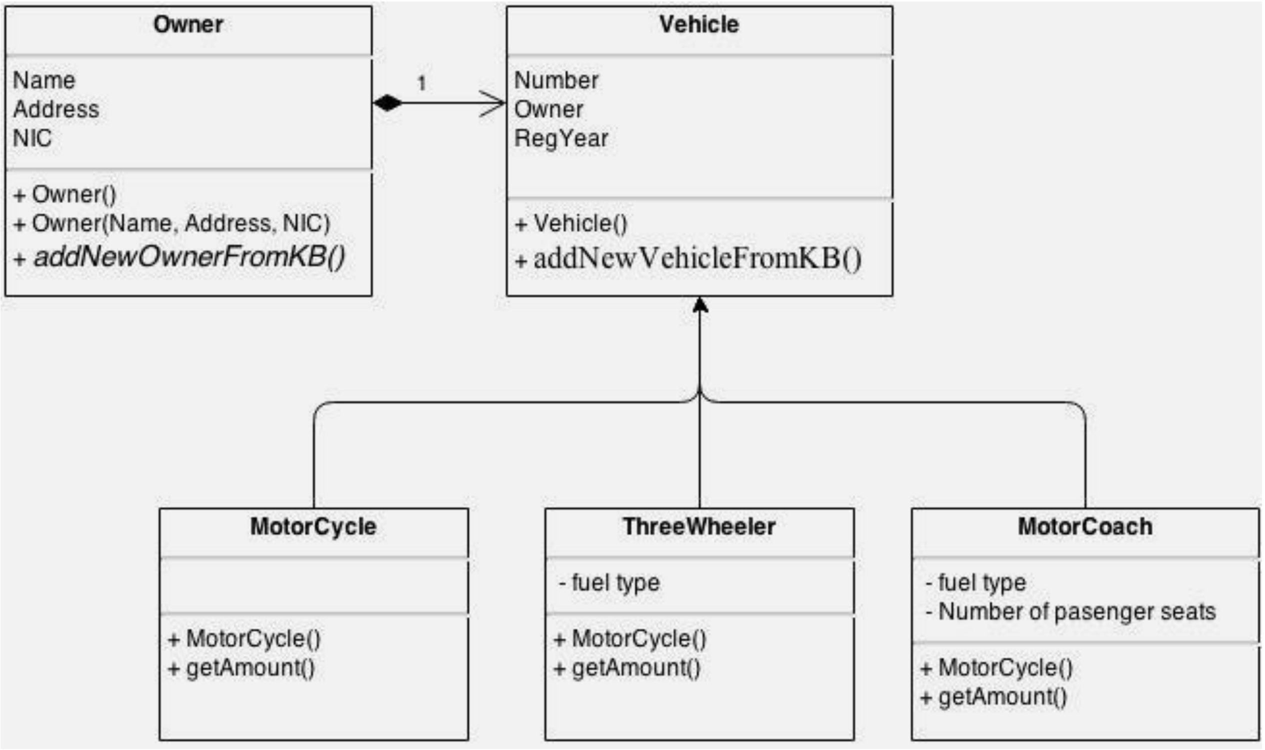

Figure 1: Class diagram of the proposed system

a)

i. Create a Java class "**Owner**" with three (3) private instance variables to represent ***name***, ***address*** and ***NIC*** to represent owner of a vehicle.

ii. Define a default constructor and a parameterize constructor to initialize an Owner object.

iii. Update the existing method *toString()* to return owner name and NIC number in the format as given below.

```
SAMAN KUMARA (783454678v)
```

The value shown above ***name*** *"Saman Kumara" and* **NIC** *"783454678v"*

iv. Define a method name "input()" to add owner details from a keyboard.

b)

i. Create a Java class "**Vehicle**" with the following.

- A private instance variable to represent vehicle ***number***.
- A private instance variables to represent ***owner of the vehicle***
- A private instance variables to represent ***registered year of the vehicle***
- Define a default constructor and a parameterize constructor to initialize a vehicle object.
- Define a method name "input()" to add new vehicle details from a key board.

ii. Create a Java class "**MotorCycle**" (extends from Vehicle) with the following.

- A default constructor and a parameterize constructor
- *A method* ***getAmount*** to return calculated registration fee
- A *toString* method that returns a String as given below.

```
SAMAN KUMARA (783454678v)   [Motorcycle]-XY-3456 (2001)       Rs. 700
```

The value shown above is name "Saman Kumara" and NIC is "783454678v"
XY-3456 is Motorcycle *number and 2001 is registered year*
The value shown above is calculated registration fee (Rs. 700.00)

iii. Create a Java class "**ThreeWheeler**" (extends from Vehicle) with the following.

- A private instance variable to represent fuel type
- A default constructor and a parameterize constructor
- *A method* ***getAmount*** to return calculated registration fee
- A *toString* method that returns a String as given below.

```
SAMAN KUMARA (783454678v)   [ThreeWheeler]-XY-3456 (2001)    Rs. 850
```

The value shown above is name "Saman Kumara" and NIC is "783454678v"
XY-3456 is Three Wheeler *number and 2001 is registered year*
The value shown above is calculated registration fee (Rs. 850.00)

iv. Create a Java class "**MotorCoach**" (extends from Vehicle) with the following.

- A private instance variable to represent fuel type
- A private instance variable to represent number of seats available
- A default constructor and a parameterize constructor
- *A method* ***getAmount*** to return calculated registration fee
- A *toString* method that returns a String as given below.

```
SAMAN KUMARA (783454678v)   [MotorCoach]-XY-3456 (2001)       Rs. 2000
```

The value shown above is name "Saman Kumara" and NIC is "783454678v"
XY-3456 is Motorcoach *number and 2001 is registered year*
The value shown above is calculated registration fee (Rs. 2000.00)

c)

Create a Java class "**RegistrationApp**" with the following instance variables and methods.

- A private instance variable to represent registration count.

i. Declare a private instance variable name **vehicle** (an array of 50 Vehicles) to represent 50 vehicles.

ii. Add a default constructor to initialize an array object (Vehicle) through its default values

iii. Define a method *input()* to add new registration through the keyboard.

iv. Define a method *printRegSummary* to print the summary of the existing registrations as shown in the below.

```
-----------------------------------------------------------
Vehicle Registration System

------------------------------------------------------------
Owner                     Vehicle                      Price
------------------------------------------------------------

SAMAN KUMARA(78567567v)  [MotorCycle]- 34-45657(1999)   Rs:  700

RUWAN SILVA(75678902v)   [MotorCoach]- 300-2345(2010)   Rs:  2000


------------------------------------------------------------
```

v. Define a method save*RegSummary* to save the summary of the existing registrations as the text file name "reginfo.txt" The output of the summary is same as the output of the part c (iv).

vi. Define a method *printCacheBalance* to print number of registered vehicles and the cache balance.

```
-------------------------------------
Number of Registrations     : 2
Total Balance (Rs)          : 2700.00
-------------------------------------
```

vii. Create a main menu to display *menu* information as shown in the below

```
-------------------------------------
Vehicle Registration System
-------------------------------------
              MAIN MENU
=====================================
[1] Add New Registration
[2] Print Registration summary
[3] Save Registration summary
[4] Print cache balance
[5] Exit
Enter you option [1-5] :
```

***