COMP 1409 – Assignment #2 (15 points)
Due: 11:59 p.m. the night before session 8

Vehicle Purchase Project

This project is meant to give you practice working with multiple classes which work as a "partnership".

Customer class

The Customer class has the following fields:

```
private String firstName
private String lastName
private String driversLicense
private String address
private String phoneNumber
```

The overloaded constructor:
Will receive data to initialize all the fields above.

The class provides both accessor and mutator methods for each field, and also a method that returns the full name. The first letter of each name component must be uppercase and the rest of the letters must be lowercase, no matter how the names are passed to the constructor.

You must ensure that names, address and driversLicense fields are not null and are at least one character in length, otherwise the fields will not be "mutated".

PurchaseDate class

The PurchaseDate class has the following fields:

```
private int year
private int month
private int day
```

The constructor looks like this:

```
public PurchaseDate(int theYear, int theMonth, int theDay)
```

The constructor must ensure that the year is no later than CURRENT_YEAR, that months are JANUARY (1) to DECEMBER (12) and that days are FIRST_DAY (1) to LAST_DAY (31). Use constants. If any of the parameters passed to the constructor are incorrect, use CURRENT_YEAR, JANUARY, and FIRST_DAY as the default settings. Do not worry about months with fewer than 31 days.

Provide both accessor and mutator methods for every field, and also a method that returns the full date as a String in the exact format of yyyy-mm-dd (for example 2014-03-30).

The mutator methods must ensure that the year is no later than CURRENT_YEAR, that months are JANUARY to DECEMBER and that days are FIRST_DAY to LAST_DAY; otherwise the fields will not be “mutated”.

Vehicle class

As per Assignment 1(use A-1 solution code).

VehiclePurchase class

The **VehiclePurchase** class has the following fields:

```
private Customer customer
private PurchaseDate  purchaseDate
private Vehicle vehiclePurchased
private boolean servicePackage
public static final double SERVICE_FEE = 500.00
```

The single VehiclePurchase constructor looks like this:

```
public VehiclePurchase(Customer renter, PurchaseDate  purchaseDate, Vehicle
vehiclePurchased, boolean servicePackage)
```

Provide accessor methods for each field in the class. Note that the accessors for `customer`, `purchaseDate` and `vehiclePurchased` will return the relevant object types. There should be NO mutator methods for `customer, purchaseDate` or `vehiclePurchased.`

Provide a mutator method for `servicePackage`.

Provide the method `calculatePurchasePrice(double purchasePrice)`that first uses the existing Vehicle method to validate the parameter. If the `servicePackage` field is true then the `SERVICE_FEE` is added to the parameter and the Vehicle sellingPrice is reset to the new value.

Provide the method that displayDetails() on the screen all the information for a purchase agreement, e.g.

```
Customer: Darby Dog
Purchase Date: 2014-05-20
Vehicle Description: Jalopies Are Us Vehicle Summary:
Vehicle: 1974 Chevrolet Monte Carlo
Stock Code:1974CevMC
Dealer Cost: $250.00
Selling Price: $895.95
Profit Margin: 72%
Dollar Profit: $645.95
SERVICE PACKAGE INCLUDED
```

Running the application

- First, you create Objects of three classes `Customer`, `PurchaseDate`, and `Vehicle`
- Second, you create a `VehiclePurchase` object by passing each of the other three object reference names to the `VehiclePurchase` constructor along with the boolean value.

Marks will be given for:

- Functionality – Your project meets the functional requirements listed above.
- Programming style – Your project uses correct style as outlined in Appendix J of your text. This includes comments, indentation, correct use of naming conventions etc.

Create a .zip file containing your entire BlueJ project. Name the .zip file with your name and the assignment number, e.g. “A00123456_Jalopies_2.zip”. Upload the file to the D2L dropbox before the cutoff time.