COMP 1409 – Assignment #1 (15 marks)
Due: 11:59 p.m. the night before session 6

Your client "**Jalopies Are Us**" is a small used car lot that is looking to expand their business. They have asked you to develop a system that tracks their auto sales. Your task is to start by developing a Vehicle class that will hold all data as it pertains to any vehicle they have on their lot. The second and third stage of the project will see additional features added.

• Create a new BlueJ project called **`<student_number>_Jalopies`**.
• Create a class called **`Vehicle`**. It hold the following fields:
• **`stockCode, make, model, year, dealerCost, sellingPrice, profitMargin`**.
• Choose appropriate data types. Do not add any other fields.

• Create a default constructor.
• Create a constructor that has parameters for **`stockCode, make, model`** and **`year`**. DO NOT provide constructor parameters for any of the other fields. The constructor must check the parameters to ensure they are not null or empty. The year parameter must be between 1970 and the current year. If any of the parameters are not valid then this constructor will use the default value for that field.

• For all fields in the class, provide accessor methods that return the value of the field. These methods must be named to start with "get", eg. **`getStockCode()`**.

• Provide mutator methods for **`stockCode, make, model`** and **`year`** that set the value for each field. These must use the same validation as required in the constructor. If a parameter is not valid the field will remain unchanged and an error message is displayed. These methods must be named to start with "set", eg. **`setStockCode()`**.

• Provide a method called **`setDealerCost(data_type)`**. This mutator method takes a parameter that specifies the value the dealer paid to purchase a jalopy from the scrap dealer and assigns it to the **`dealerCost`** field. The value must not be negative.

• Provide a method called **`checkStandardSellingPrice(data_type)`**. This mutator method guarantees the selling price is at least 25% higher than **`dealerCost`**, and assigns the value to the **`sellingPrice`** field. If the parameter value is rejected be sure to display a message to the user that includes both the **`dealerCost`** and **`sellingPrice`**.

• Provide another method called **`setSellingPrice(data_type)`**. This mutator method guarantees the selling price is not less than 0, and assigns the value to the **`sellingPrice`** field.

• Provide a method called **`calculateProfitMargin()`**. This mutator calculates the profit made on selling a vehicle as a percentage. It also assigns the value to the **`profitMargin`** field. To calculate the profit margin use the following formula:

```
profitMargin = (sellingPrice - dealerCost) / sellingPrice
```

• Provide a method called **`calculateProfit()`**. This method calculates and **returns** the profit made on selling a vehicle as a dollar value.

• Provide a method called **`printDetails()`**. This method displays item information on the screen, and should be formatted as below:

```
Jalopies Are Us Vehicle Summary:
Vehicle: 1974 Chevrolet Monte Carlo
Stock Code: 1974ChevMC
Dealer Cost: $250.00
Selling Price: $395.95
Profit Margin: 37%
Dollar Profit: $145.95
```

FINALLY provide the driver class JelopiesDriver that implements the main(...) method. It must create and test an object thoroughly and demonstrate all the program functions.

Be sure each method has a Javadoc-style descriptive comment above the method header and that the class itself has a descriptive comment that includes your name as author.
Use BlueJ to interactively test your methods as you write them. Write them one at a time, and test each immediately to be sure it is correct. Testing requires creating an object and invoking the method. Be sure to test with both valid and invalid data, i.e. test the "set" methods with positive values to be sure the field is changed correctly, and test with negative values to be sure the error message is displayed and the field not changed.

Marks will be given for:
• Comments – appropriate and complete.
• Style – see the style guide Appendix J of your textbook.
• Correctness and completeness – code meets the requirements listed above.

Create a .zip file containing your entire BlueJ project (zip the folder, not the individual files). Name the .zip file with Student Number and the assignment number, e.g.
"**`A00123456_Assignment_1.zip`**". Upload the file to D2L before the cutoff time.