Assignment 2 – Design and Testing with Classes

You will submit a reflections document with every assignment. This is your chance to explain why you did what you did. Or why something you tried didn’t work. You can explain what you did to test your program to make sure it works. You can also document features or bugs that inhibit the program in some way. These fit into Polya’s method for problem-solving which is all programming is.

Your grade is based on Polya’s 4 steps for solving problems:
• Understanding the problem. (Recognizing what is asked.) – 5 points
• Devising a plan. (Responding to what is asked.) – 5 points
• Carrying out the plan. (Developing the result of the response.) – see below
• Looking back. (Checking. What does the result tell me?) – 5 points

You need to design, implement, and test a program to maintain an inventory for a small used vehicle dealer. The program should maintain and display a list of vehicles. Your program should also calculate the sales price and monthly payments for a designated vehicle.

You will use a class for Vehicle. **The class should have data elements for the following information:** Lot #, make, model, color, mileage, fuel consumption and list price. You will have the following derived classes from Vehicle: Automobile, Motorcycle. What additional characteristics do each of these need? There are many to choose from for Automobile, maybe less so for Motorcycle. You will need to have a function to add miles to the mileage; from test drives, or employee use. You only need to update the mileage. If you want to be an unscrupulous dealer (see the extra credit) you COULD have another function to decrease the mileage. Do you need any other functions in the Vehicle class, or its children?

You will also need a List class. **The List class will use a dynamic array to store Vehicle objects.** You will have one list for Automobiles and one for Motorcycles. You will NEVER have an object of class Vehicle. As each vehicle is entered an object of the appropriate type must be created and added to the correct array. Remember you may need to increase the size of the array. Start each array with 2 elements so the grader can test your resizing. Double the size each time you increase capacity.

**Your program must perform the following activities:** create a list, add items, search for a vehicle, calculate the price and payments for a vehicle, remove a vehicle, and display the inventory. Which of these should be member functions of the List class, and which will be functions in your program? The user should be able to search for any attribute (data member) of the Vehicle, Automobile, or Motorcycle classes. To add an item you should prompt the user to enter the required information.

**Check the input when adding a vehicle (input validation).** This could be easy. Transmissions might be AT, CVT, and MT. The boss definitely doesn’t want negative prices, but also doesn’t want a vehicle sitting because an extra 0 got put in the price. So anything over $10,000 requires an acknowledgement. What else to check? Include the description in your design!

**Your program should calculate the final price of the vehicle.** We’ll keep things simple. There is only 1 list price. When the user selects a vehicle to purchase you start with the list price, you can deduct any incentives (just entered by the user), add a $95 administrative fee, add any taxes*, deduct the amount of the trade-in (if any), enter an interest rate and loan period to calculate* the monthly payment.
Display each of these values to the user. Give them the option to indicate the vehicle sold. If it is sold remove it from the inventory.

**The display should show:** Menu of choices (display all vehicles in the inventory, adjust mileage for a vehicle, search for a vehicle, add a vehicle, calculate purchase price for a vehicle, and exit. The calculate price option should remove the vehicle if it was sold. The search should display the information for each vehicle matching the criteria.

* Subjects to research- taxes to by a car in Oregon, how to calculate auto loan payments (WikiHow is a good place to start),

You must create a design document. It should include the design of the classes (class hierarchy) and how you will use the classes. Remember to free memory if no longer used. Figure that out, **before** you start coding. If you do the design properly the coding should be easier. The design should also include your development plan, i.e. which parts of the program you are doing in what order.

Your reflections document should include the design document, the testing results, and describe any changes you made to your original design.

You will provide a simple test plan.

**NOTE:** Please use incremental development!

When you have a (partial) program that compiles and runs properly save a copy!

A (partial) suggestion for the development plan. Start with the menu using stubs (as demonstrated in the slides). Then work on adding a vehicle. Pick either derived class and get it working. The List should be created, if you didn’t do it already. You may need to work on displaying the inventory at the same time so you can test adding. Then move on to calculating the sale price to remove a vehicle.

You must **include a makefile and put all files for your assignment in a zip file.** If you do not do this assignment will NOT be graded.

**Grading:**

- programming style and documentation (10%)
- create and use the List class and object (10%)
- create and use the Automobile class (inherit from Vehicle) and objects (10%)
- create and use the Motorcycle class (inherit from Vehicle) and objects (10%)
- add and remove vehicles to the inventory (5%)
- input validation when adding vehicles to the inventory (5%)
- correctly increase the mileage (5%)
- your arrays will be resized if adding and at capacity (5%)
- display the list of all vehicles in the inventory (5%)
- display the result of a search of the inventory (10%)
- correctly calculate the sale price and loan details of a vehicle (5%)
- you properly manage memory, i.e. NO leaks (5%)
- reflections document to include the design description, test plan, test results, and comments about how you resolved problems while designing and implementing
your program (15%)

NOTE: Your grader can make any size deduction he/she feels is appropriate if you have a memory management problem with your program. In the rubric you lose 5 points already if you have any memory leak.

Extra Credit-
The used car lot isn’t making enough profit. The owner wants to do some creative (i.e. unscrupulous) activities to increase the price of the cars. One simple one (for the program) is decreasing the mileage. For other ideas talk to friends or family or a mechanic or me. Many movies can be the source of ideas. Watch *Matilda* again! 😊 Maybe sand in the transmission? That quiets that whining noise but is only good for a few miles before it binds the tranny (let’s say 100). Be creative! Parts falling off would just be the same, a mileage limit with the inflated price before you’d HAVE to sell it. AS IS of course. 😊 Implement these. Come up with another (that doesn’t involve setting or limiting the mileage) and implement it. (5 points)