Lab 6 COMP1161 – Introduction to Object Oriented Programming

DESIGN OF CLASSES AND OBJECTS

THIS LAB HAS 4 EXERCISES

Exercise #1

Design and implement a class called *Engine* that will store data about a motor vehicle's engine. Data for an engine includes:

- engine number (up to 12 alphanumeric characters)
- cc rating (the size of the engine in cubic centimeters, eg. 1200, 1800, etc.)
- number of cylinders (a value in the range 1-6)
- condition of engine (Poor, Fair, Good, Excellent)

Your implementation should include:

a) constructor which initializes all instance variables
b) Getter method for each instance attribute
c) A setter method for engine condition only (no setter should be written for the other variables)
d) A `toString()` method which should return a string with the format:

Engine Information:
Engine number : xxxxxxxx
CC rating: xxxxx
Number of cylinders: xxxxxx
Condition: xxxxxxxxx

Exercise #2

Design and implement a class called *Vehicle* that will store data about a motor vehicle. Data for a vehicle includes:

- make (e.g. "Toyota")
- year (e.g. 2010)
- engine_info
- mileage (number of miles)
- Value (e.g. $1.5M)

Your implementation should include:

a) A constructor which initializes all instance variables;
b) A second constructors which initializes just make, model, and year
c) A *toString()*method which s should return a string with the format:

Vehicle Information:
Make: xxxxxx
Year: xxxxx
Engine Information:
Engine number : xxxxxxx
CC rating: xxxxxxx
Number of cylinders: xxxxxxx
Condition: xxxxxxxx
Mileage: xxxxxx
Value: xxxxxxx

NB: Use number format where necessary.

Exercise #3

In the Vehicle class, define the following methods that will calculate the current value of the vehicle after depreciation cost:

`calculateValue (int rate, int newMileage)`

- `compute the difference between old and new mileage.`
- recalculate the value of the car using the formula: **value = value – (rate *(new mileage – old mileage))**
- **update the new mileage**

`calculateValue (int rate)`

recalculate the value of the vehicle using the formula: **value = value – (rate * value)**

Exercise #4

Implement a driver class called Rental that will do the following:

a. Declare and initialize a list of vehicles.

b. Add up to 5 vehicles of your choice

c. Change the value of second vehicle using the first version of the calculateValue method.

d. Change the value of the fourth vehicle in the list using the second version of the calculateValue method.

Discussion

Can you identify each of the following concepts in the classes that you have written:

a) Encapsulation

b) Dependency

c) Aggregation

d) Method overloading