**Assignment for COSC 1200 – Object Oriented Programming I**

**Assignment 4: VIN Tester Due: Thursday, March 28th**

Prior to attempting this problem, you should have done the following:

1. Viewed the content, including assigned readings for weeks 1 through 9.

2. Viewed the lectures and practiced with the demo code for weeks 1 through 9

General Requirements

1. This assignment can be completed EITHER in a group of 2 or individually (your choice).

2. Submit your solution to the appropriate assignment folder on DC Connect by the due date provided. **Submit your java code and attach execution in single word file.**

3. Note: Each class needs comments at the top with at minimum your name(s), the date (date created or date due), a description of what the class does. In addition, it is expected that you provide a sufficient amount of commenting in your code.

Requirements

Complete and submit the program source code that satisfies the following requirements:

For this assignment you are going to create four (4) different classes. Three (3) that will be user defined classes (parent and children classes) and the fourth one will implement the other three (a tester class)

**1. You will create a Vehicle.java class file that will contain the following:**

a. Has four (4) attributes (all to be private access) entailing the following:

i. a String attribute named “make”

ii. a String attribute named “model”

iii. a int attribute named “year”

iv. a String named “VIN”

b. For each of your attributes create accessor methods (get() and set() methods), make these public accessible. Name them what you like but must they must start with ‘get’ and ‘set’ respectively.

c. Has a constant (i.e. unchangeable) class variable of type byte named VALID\_VIN\_LENGTH set equal to 17, this should be accessible to any class, from anywhere

d. Create a new constructor that takes 4 arguments (one for each of the attributes in the class), this method should use the set() methods you created in part (b).

e. Create a method named getVehicleInfo(), that is public accessible and returns a String. Specifically, a formatted (with new lines and tabs) String of the calling objects Vehicle information. The attributes should be retrieved using the get() methods (not calling the attribute directly). It should display in the format:

Make: Ford

Model: Mustang

Year: 2007

VIN: 1F4GFDYRXGS324567

**2. You will create a Automobile.java class file that will contain the following:**

a. Inherits all of the methods and attributes of a Vehicle class, but that also has the following private attributes:

i. An int named horsepower

ii. A String named fuelType

b. For each of your new attributes create accessor methods. Make these public accessible

c. Define a constructor that takes six (6) arguments (for the six attributes), and then sets each attribute using the six different set() methods (the four public ones in the parent, and 2 new methods in the derived class.

d. Override the base class’ getVehicleInfo() method so that the Automobile’s info is displayed. For this method have all six (6) pieces of information be displayed by making six calls to the get() methods (i.e. you are replicating code from the base class)

Make: Chevy

Model: Corvette

Year: 1975

VIN: 1G4EDYR2XAS123456

Power: 254 hp

Fuel Type: unleaded

**3. You will create a Motorcycle.java class file that will contain the following:**

a. Inherits all of the methods and attributes of a Vehicle class, but that also has the following private attributes:

i. An int named engineSize

ii. A String named engineType

b. For each of your new attributes create accessor methods. Make these public accessible

c. Define a constructor that takes six (6) arguments (for the six attributes), this constructor should make a call to the parent contructor (i.e. the constructor in the Vehicle class), and then set the last two (2) attributes using the appropriate set() method. N.B. this will not create an error.

d. Override the base class’ getVehicleInfo() method so that the Automobile’s info is displayed. For this method, call the parent’s vehicle info and append the motorcycle specific information (i.e. the last two attributes) be appended to the returned String using the two get() methods

Make: Honda

Model: Shadow

Year: 2004

VIN: JH2RC040XBM200124

Engine Size: 750 cc

Engine Type: 4 stroke

**4. Create a Java class named VehicleTester, that has a main method that performs the following:**

a. Instantiates three (3) objects of different types, one Vehicle, one Motorcycle and one Automobile. These objects should be passed the following arguments:

i. Your Vehicle object should be passed the values: “Ford”, “Mustang”, 2007, 1F4GFDYRXGS324567

ii. The second object should be of type Motorcycle and should be passed the values:

“Honda”, “Shadow”, 2004, JH2RC040XBM200124, 750, 4 (the last two arguments indicate a 750 cc engine, 4-stroke engine)

iii. The third object should be of type Automobile and should be passed the values:

“Chevy”, “Corvette”, 1975, 1G4EDYR2XAS123456, 254, “unleaded” (the last two arguments indicate a 254 hp engine that takes unleaded gasoline)

b. Each object’s information should then be displayed using a System.out.println() of the String returned from the displayInfo() of each method type

c. To illustrate the concept of polymorphism, create an Array of Vehicles, and place the three different object instances in it. In a for loop, have each Vehicle display its info with a System.out.println() and the String returned from vehicleInfo() method.

Note: Please submit word file with your java code and attach execution.

Style Guide Notes

To be eligible for full marks on this or any assignment in this course your application must conform to the requirements as outlined above as well as the Google style guide as well as appropriate and complete program documentation.