SE116 – LAB#5 2021-2022 SPRING

Aim: Understanding constructors, overloaded methods, "this" reference, "has-a" relationship, static class members, "final" instance variables.

TO DO @ LAB:

1. Implement a program to hold the general information about cars. Your source codes should contain the following two classes: `Car` and `Test`.

The class `Car` will contain the following `private` data members to represent a car: `int modelYear`, `String model`, `double topSpeed` and `double price`. Remember to implement appropriate `public` set/get methods for these data members. Additionally, implement a method named `printInfo` to print out all data members.

Additionally, implement a default constructor (i.e., non-parameterized constructor) for the class `Car`. Keep in mind that the main objective of a constructor is to initialize the data members. In your program, your default constructor of class `Car` will initialize its data members with the following values:

```
modelYear = 0
model = "Unknown"
topSpeed = 0.0
price = 0.0
```

The class `Test` only deals with the method `main`. In `main`, define an array of 5 `Car` references. Do not forget that the array object initially holds no `Car` instances (i.e., the initial contents of the array are `null`). Therefore, you first need to construct 5 new `Car` instances and then need to make array contents refer to these instances. In order to see the effect of the `Car` constructor you implemented, directly call the method `printInfo` for each `Car` instantiated. After that, fill the information for each car instance using corresponding set methods and reprint out the information for each `Car`.

2. Modify your project implemented in Question#**1** in the following manner:

In your project, implement the class `Engine`. This class will represent the engines of the cars. The class `Engine` will contain the following two `private` data members: `double volume` and `int numberOfCylinders`. Remember to add proper set/get methods and a default constructor (non-parameterized) as well.

Modify your class `Car` in order to use the "has-a" relationship between class `Car` and class `Engine`. The objective is to mention the following statement: A car "has-a" engine. In this manner, a reference of class `Engine` named `eng` (i.e., `Engine eng;`) will be a `private` data member of class `Car`. Do not forget to add required `public` set/get methods for this data member. Additionally, update `printInfo` to print out the engine data as well as the other information. Furthermore, modify the constructor of class `Car` to make the reference `eng` refer to a new `Engine` instance.

Modify your class `Test` in a way that it also deals with the data about engine. In `main`, use a `final` variable with the constant value 5 as the size of the array of `Car` references. After defining the array, construct the new `Car` instances to be referred by the array contents and fill all information (engine data included) for each car with the values to be read from the user. Finally, print out all information of each `Car` referred by the array elements.

TO DO @ HOME:

3. Modify your project implemented in Question#**2** in the following manner:

Modify the class `Engine` by overloading the constructor method. In this manner, provide the class `Engine` with a non-parameterized and a parameterized constructor. The parameterized constructor should take 2 parameters to be assigned to corresponding `volume` and `numberOfCylinders` data members. By the way, use corresponding set methods to assign data.

Modify the class `Car` by overloading the constructor method. In this manner, provide the class `Car` with both non-parameterized and parameterized constructors. The parameterized constructor should take 5 parameters to be assigned to corresponding data members (i.e., `modelYear`, `model`, `topSpeed`, `price` and `eng`). By the way, use corresponding set methods to assign data.

Moreover, add the following `private static` data member into class `Car`: `int carCounter`. Its initial value is 0. The data member `carCounter` will count the number of `Car` objects to be constructed in the program. Therefore, you need to revise the definitions of the constructor methods accordingly. Additionally, define a `public static` get method to return the value of `carCounter`.

Modify the class `Test` in the following manner: In `main`, construct some instances from class `Car` using both non-parameterized and parameterized constructors. Design the program in such a way to reveal the objective of using a `static` data member. For example, once you construct a new `Car` instance, you would print out the value of `carCounter`.

Furthermore, in every constructor method of every class in your project, print out a sentence to inform that a new object from the corresponding class is being constructed. (In order to return the name of a class the following statement can be used: “ `getClass().getName();` ”)

4. Modify your project implemented in Question#**3** in the following manner:

Modify the class `Test` in the following manner: In `main`, use an `ArrayList` object to store `Car` references. Your program should permit its user to enter input data for new `Car` instances to be sequentially added into the `ArrayList` object. Your program may regularly keep on reading input data using a loop until the user enters `"Quit"` after a `Car` is inserted into the `ArrayList` object.