CSE201 Advanced Programming (CSE Section)
Lab Assignment 03
IIIT-Delhi. 31st August 2018. Due by 11:59 pm on 2nd Sep 2018

This is a take-home lab assignment. No extensions whatsoever will be provided. Any submission after the deadline will not be evaluated. If you see any ambiguity or inconsistency in a question, please seek a clarification from the teaching staff. Please read the entire text below very carefully before starting your implementation.

Plagiarism: All submitted lab assignments are expected to be the result of your individual effort. You should never misrepresent someone else's work as your own. In case any plagiarism case is detected, it will be dealt as per IIITD plagiarism policy and without any relaxations: https://www.iiitd.ac.in/sites/default/files/docs/education/AcademicDishonesty.pdf

Please note that you are **not** allowed to discuss the design/solution of the lab assignment (e.g., backpack discussions, etc.). Anyone who is found doing this will be treated as plagiarism case.

Topics Covered: Inheritance and Polymorphism

Your objective is to build a software system for vehicle insurance settlement using object oriented approach. You must also use inheritance and polymorphism in this assignment.

The system should be able to support both engine powered and non-engine powered vehicles. Each registered vehicle should have the name of its owner, name of the model, and total number of wheels. Engine powered vehicle can either be a four wheeler or two wheeler. The system even allows registrations of non-engine powered vehicles such as Rickshaw and Bicycle, although these vehicles are not provided any insurance schemes. All non-engine powered vehicles are treated equally.

Insurance policies that are available for engine powered vehicles are of two types: third party insurance policy and comprehensive insurance policy. All non-engine powered vehicles are categorized as null policy holder. Irrespective of the type of holding policy, while settling insurance claim, all engine powered vehicles are by default entitled to claim for 10% of the total damage to self (owner vehicle).

An engine-powered four-wheeler holds a comprehensive policy, which provides damage cover of 50% to the owner vehicle and 80% to the oncoming vehicle. An engine-powered two-wheeler holds a third-party policy, which only provides cover for the damage to the oncoming vehicle by 50% (nothing for self). Insurance policies have a date of expiry, after which they become invalid and cannot disburse any claim.

Design the insurance settlement system with following features:

1. In the event of a collision between two vehicles, the system should assign some damage amount to both the vehicles involved in the collision. The insurance settlement system should have a single collide method which takes two vehicles as input, assigns damages and processes settlement on both sides.
2. Figuring out the damage settlement (for self and oncoming vehicle) should be done by the collision initiating vehicle itself, according to its insurance policy. More specifically, each vehicle should have a settle method which takes the oncoming vehicle as an input parameter.

Testing of Insurance Settlement System:

To demonstrate the correct working of your insurance settlement system, you will have to invoke a simulation in which each vehicle registered in the system will collide with all other registered vehicles and during this collision it will appropriately settle the insurance claim as per its insurance policy. For this simulation you must have at least one type of vehicle under each vehicle category. Moreover, each vehicle category (if applicable) should have one vehicle that has a valid policy and another one that has an expired policy.

During collision:

1. First assign any damage amount to both self and oncoming vehicle.
2. If self vehicle has a valid policy then self would do a settlement with oncoming vehicle and reduce damages.
3. If self is not having a valid policy, then there can be no settlement from its side. Print a message stating the reason.

Output of the Program:

1. Print in tabular form the details of all the vehicles that are registered in the system. Ex: Model, owner name, Type of vehicle(two-wheeler, three-wheeler etc.), type of insurance policy class, policy validity status
2. Print some message showing the start of the simulation.
3. Simulation of collision of each pair of vehicle objects and then printing the resulting settlement details (for each collision) in below format:

   Ex: vehicle1 <model1, owner_name1> collided with vehicle2 <model2, owner_name2>
   Damages awarded to vehicle1:
   Damages awarded to vehicle2:
   Settlement details.
   vehicle1 damage status, after settlement.
   vehicle2 damage status, after settlement.

   Ex (in case of any issues with settlement):
   vehicle1 <model1, owner_name1> collided with vehicle2 <model2, owner_name2>

Damages awarded to vehicle1:
Damages awarded to vehicle2:
Settlement details:
    Print a message indicating what went wrong in settlement