# CSE201 Advanced Programming
# Tutorial 4

## Topics Covered: Inheritance and Polymorphism

Build a software system for a vehicle insurance settlement. There are two types of vehicles - *Engine Powered vehicles* and *Non-Engine powered vehicles*. Each registered vehicle should have the name of its owner, the name of the model, and the total number of wheels.

Insurance policies that are available for engine powered vehicles are of two types: Third-party insurance policy and Comprehensive insurance policy. There is no insurance for non-engine powered vehicles, hence they are categorized as a null policyholder. 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 provides 50% damage cover to the oncoming vehicle and nothing for self. Irrespective of the type of holding policy, while settling an insurance claim, all engine powered vehicles are by default entitled to claim for 10% of the total damage to self (owner vehicle).
Insurance policies have a date of expiry, after which they become invalid and cannot disburse any claim.

Each vehicle should have a “settle” method which takes the oncoming vehicle as an input parameter.

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.