## Problem Statement: Implementing Polymorphism using abstract class and interface

**Define Vehicle interface, AbstractManufacturer abstract class,Car class,Bike class and VehicleService Class as given below:**

**Interface Vehicle**

create the following abstract method.

+maxSpeed(String type) : int

**Abstract class AbstractManufacturer**

Declare the following private properties.

-name : String -modelName : String -type : String

* Provide getter for all properties

Declare abstract method

+getManufacturerInformation() : String

**Car Class**

* Make the class as subclass of AbstractManufacturer and implements Vehicle interface.
* Define parameterized constructor passing three parameters to initialize name,modelName and type.
* Override the abstract methods and follow the instructions given as comments for the business logic.

**Bike Class**

* Make the class as subclass of AbstractManufacturer and implements Vehicle interface.
* Define parameterized constructor passing three parameters to initialize name,modelName and type.
* Override the abstract methods and follow the instructions given as comments for the business logic.

**VehicleService Class has the following three methods**

+createCar(String,String,String) : Car

+createBike(String,String,String) : Bike

+compareMaxSpeed(Vehicle,Vehicle) : int

* Follow the comments to complete the business logic for all three methods

## Instructions

* Avoid printing unnecessary values other than expected output as given in sample
* Take care of whitespace/trailing whitespace
* Do not change the provided class/method names unless instructed
* Follow best practices while coding