Name: Ali Albayrak

Student ID: P304320

Practice Activities 2.1

Vocabulary:

Constructor: A specialized method that creates an instance of a class.

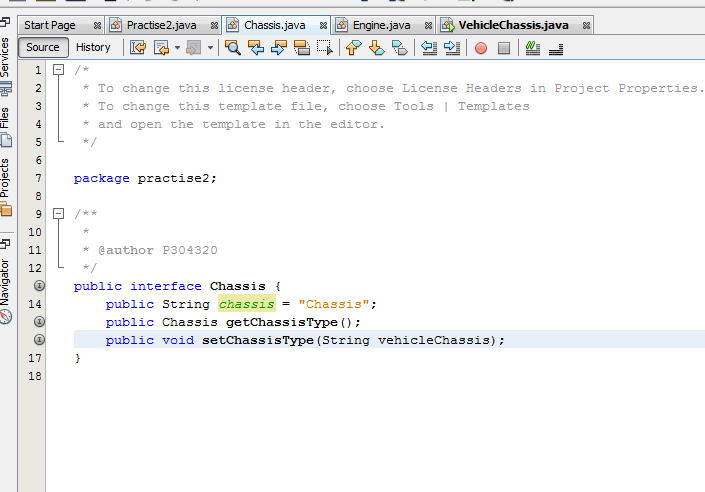
Final: A keyword that qualifies a variable as a constant and prevents a method from being overridden in a subclass.

Immutable: A class that it can't be overridden by a subclass, in fact it can't be subclassed.

Interface: Defines constants and methods without implementation

1.Create an interface named Chassis. Add the following to the interface:

* A public constant string named chassis with a value of “Chassis”.
* The definition of a public getChassisType method that returns an instance of Chassis.
* The definition of a public setChassisType that accepts a string named vehicleChassis and returns a void.



2. Create an interface Engine with the following list of public method definitions that return a void:

setEngineCylinders(int engineCylinders);

setEngineManufacturedDate(Date date);

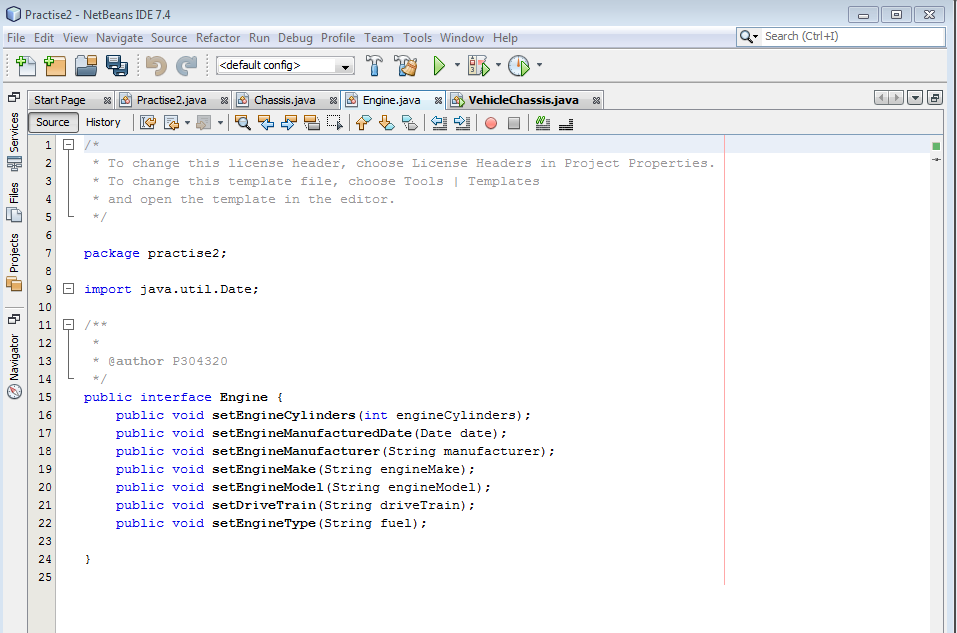
setEngineManufacturer(String manufacturer);

setEngineMake(String engineMake);

setEngineModel(String engineModel);

setDriveTrain(String driveTrain);

setEngineType(String fuel);



3.Create a concrete class named VehicleChassis that implements the interface Chassis with the following:

• Create a String named chassisName instance variable.

• A public default constructor and an overloaded constructor with the following value:

A String with a parameter value of chassisName

• Set the chassisName instance variable in both, use the interface constant as the default String value.

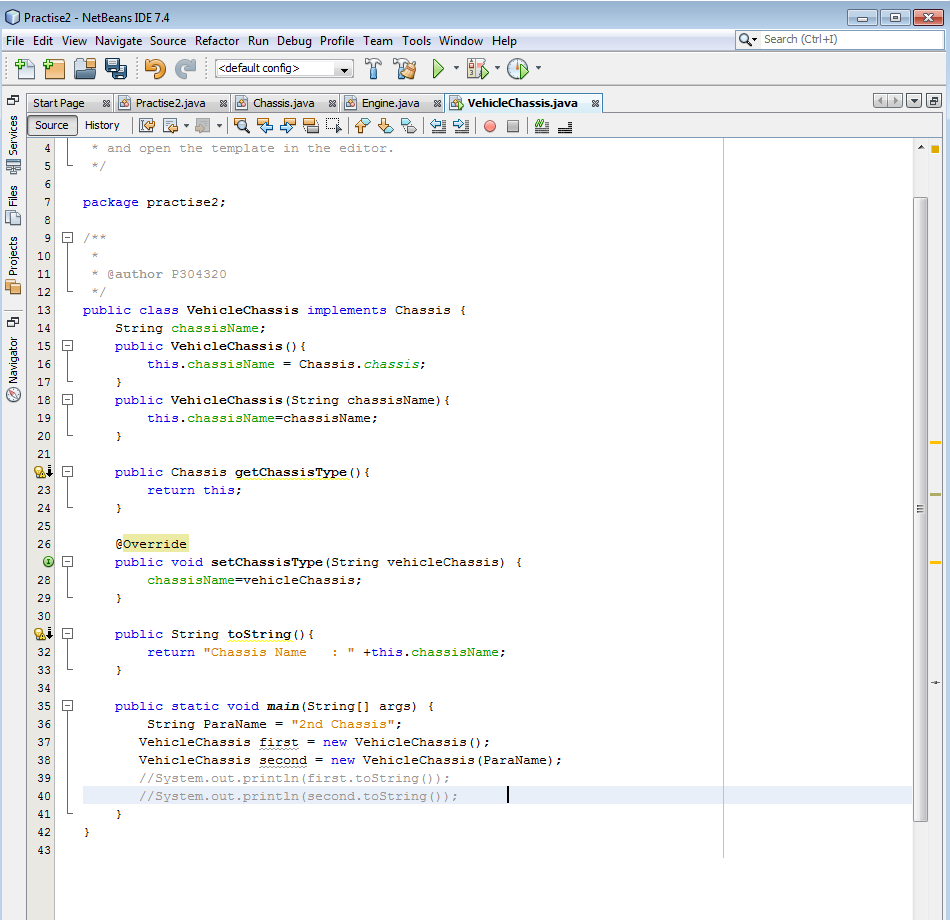
• A public method named getChassisType that doesnt have a formal parameter and that returns an instance of the interface Chassis (hint that would be a copy of this class).

• A public method setChassisType that takes a String parameter vehicleChassis and that returns a void. It should set the instance variable chassisName.

• A public toString method that returns the following:

Chassis Name : Chassis

• Write a static main method that creates two objects, one with the default constructor and the other with the constructor with parameters. Give sample data for the parameters.



4. Create a concrete class named VehicleFrame that implements the interface Chassis with the following:

• Create a String named vehicleFrameType instance variable.

• A public default constructor and an overloaded constructor with the following value:

A String with a parameter value of vehicleFrameType

• Set the vehicleFrameType instance variable in both, use “Unibody” as the default String value.

• A public method named getChassisType that didn't have a formal parameter and that returns an instance of the interface Chassis (hint that would be a copy of this class).

• A public method setChassisType that takes a String parameter vehicleFrameType and that returns a void. It should set the instance variable vehicleFrameType.

• A public toString method that returns the following:

Chassis : Chassis

Vehicle Frame : Unibody

• Write a static main method that tests two scenarios:

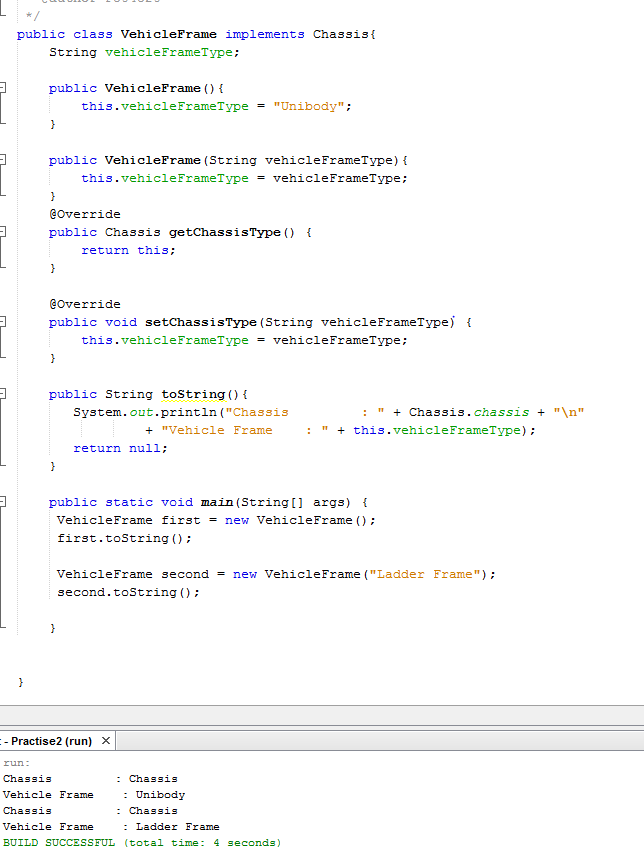
1.One that prints all fuel grade values, like:

Chassis : Chassis

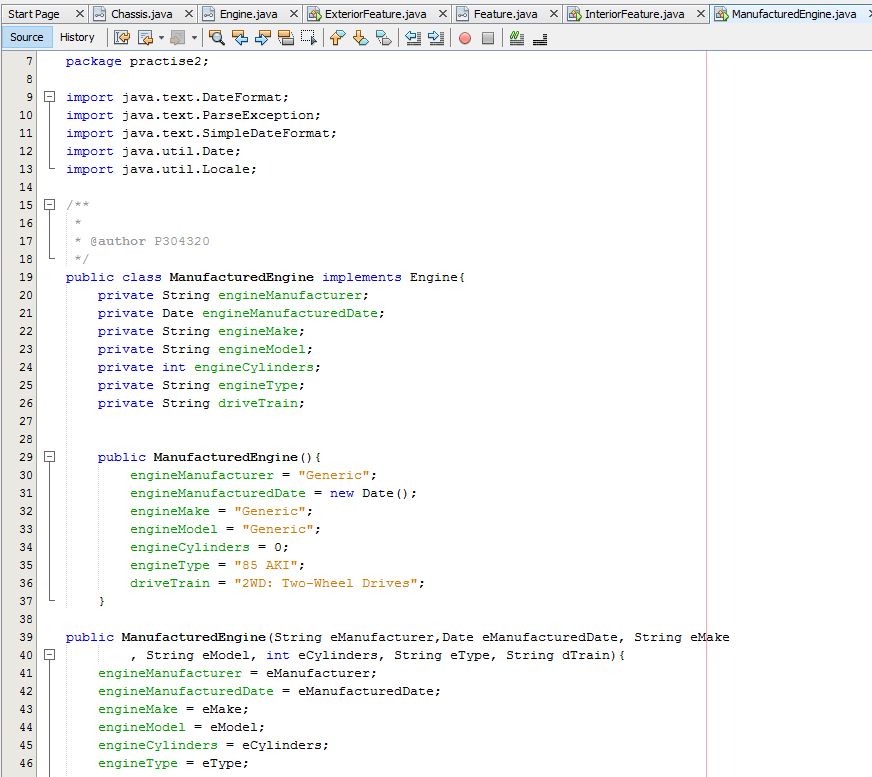
Vehicle Frame : Unibody

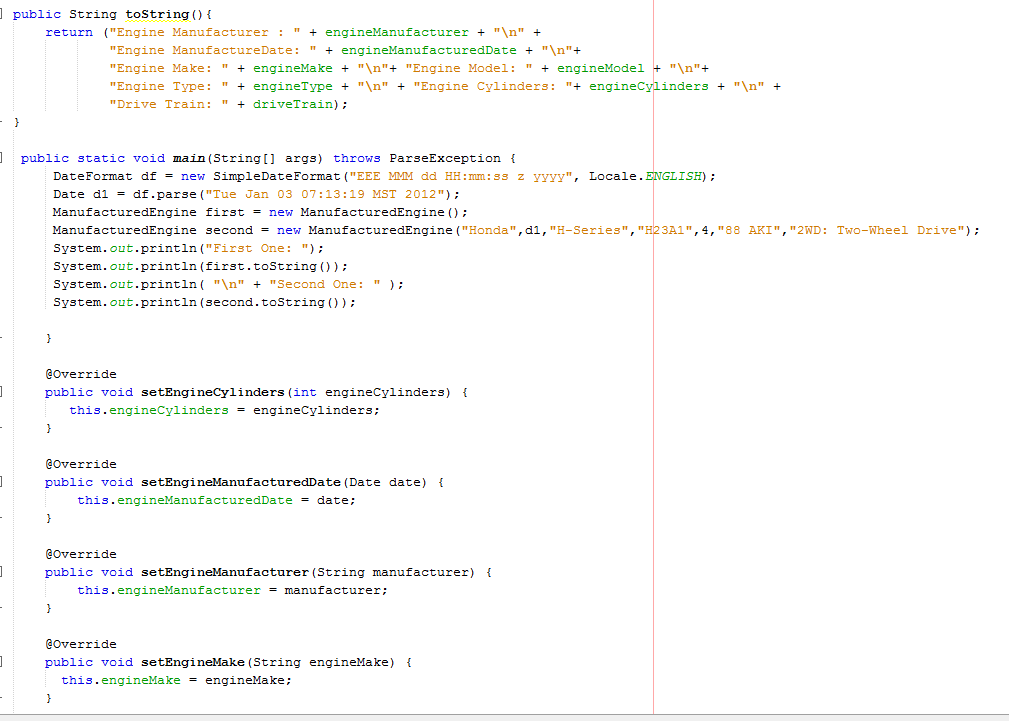
2.One that prints a value set by a single string value.

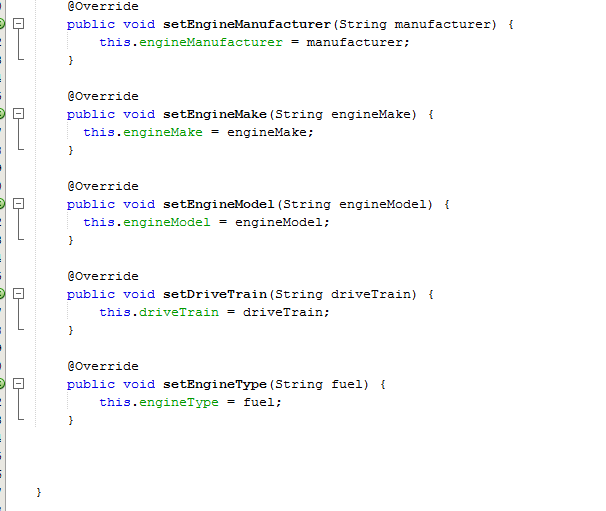
Vehicle Frame : Ladder Frame



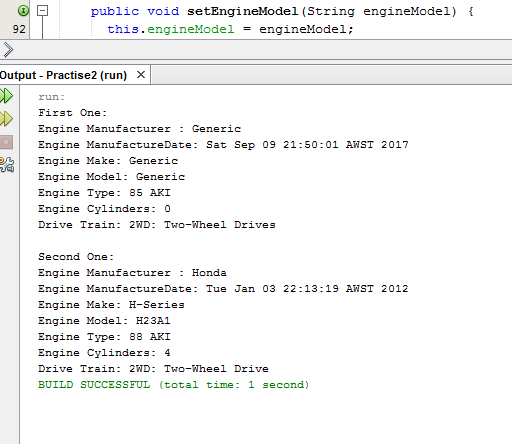
5. Create a concrete class named ManufacturedEngine that implements the interface Engine







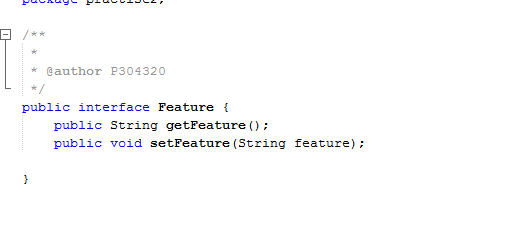
Output:



6. Create an interface Feature with the following method definitions:

public String getFeature();

public void setFeature(String feature);



7. Create a concrete class named InteriorFeature that implements the interface Feature with the following:

• Create a String named interiorFeature as an instance variable.

• A public default constructor without parameters that sets the interiorFeature instance variable to “Generic”.

• An overloaded constructor with the following value:

A String with a parameter value of interiorFeature

• Set the interiorFeature instance variable to the parameter interiorFeature.

• A public method named getFeature that doesnt have a formal parameter and that returns an instance of String.

• A public method setFeature that takes a String parameter interiorFeature and that returns a void. It should set the instance variable interiorFeature.

• A public toString method that returns the following:

Interior [Generic]

• Write a static main method that tests two scenarios:

• One that prints all fuel grade values, like:

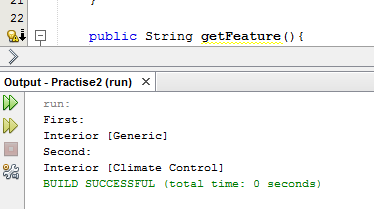
Interior [Generic]

Another like

Interior [Climate Control]



Output:



8.Create a concrete class named ExteriorFeature that implements the interface Feature with the following:

• Create a String named exteriorFeature as an instance variable.

• A public default constructor without parameters that sets the exteriorFeature instance variable.

• An overloaded constructor with the following value:

A String with a parameter value of exteriorFeature

• Set the exteriorFeature instance variable in both, use features as String values.

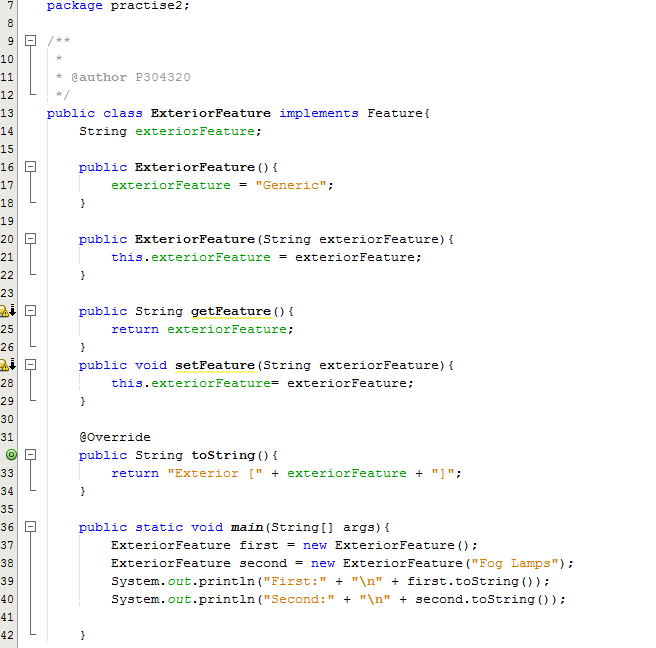
• A public method named getFeature that doesnt have a formal parameter and that returns an instance of String.

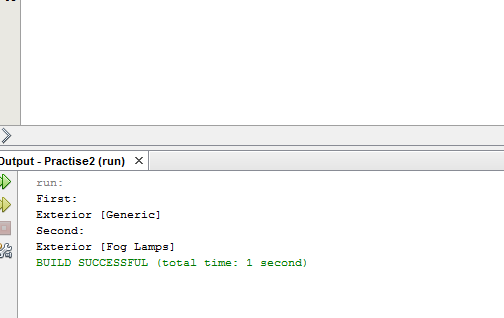
• A public method setFeature that takes a String parameter exteriorFeature and that returns a void. It should set the instance variable exteriorFeature.

• A public toString method that returns the following:

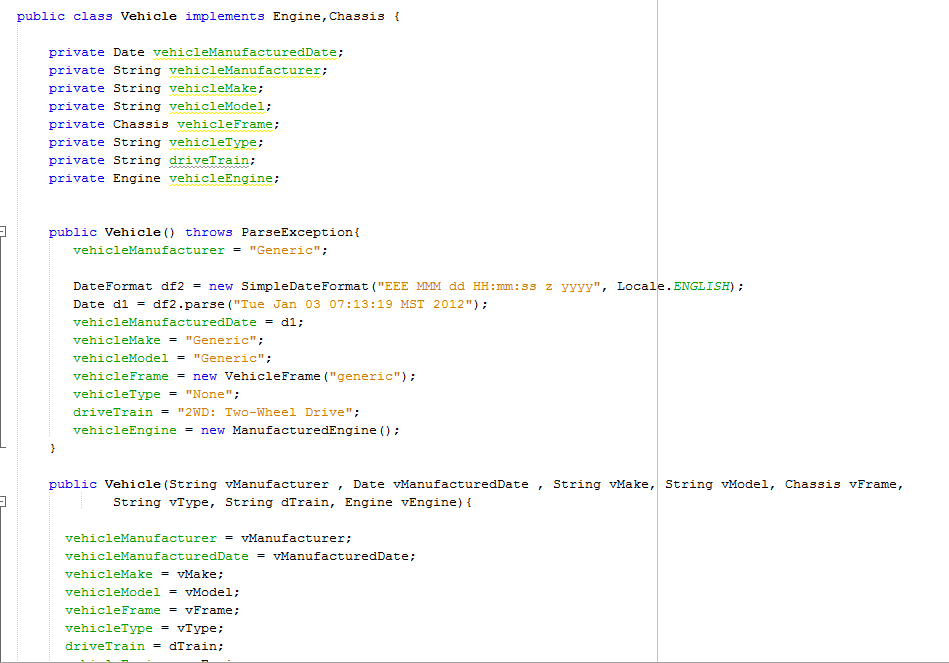
Exterior [Generic]

• Write a static main method that tests two scenarios

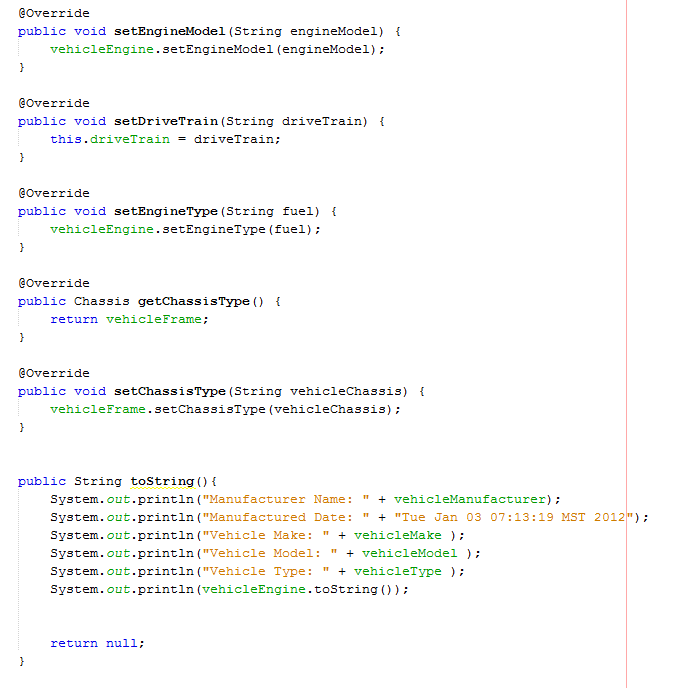


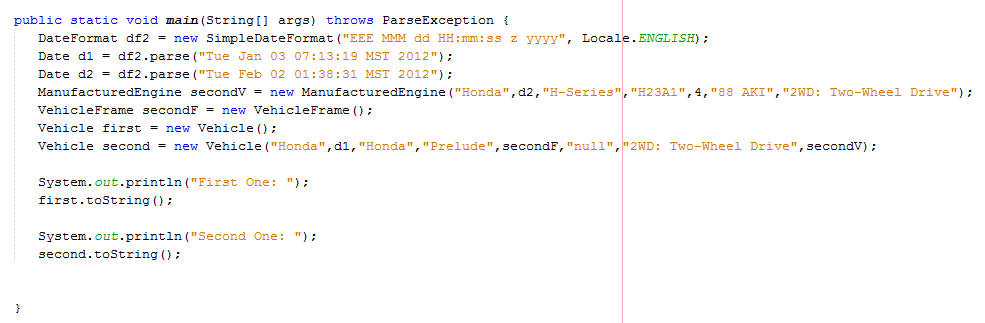


9. Create a concrete class named Vehicle that implements the Engine and Chassis interfaces









Output:

