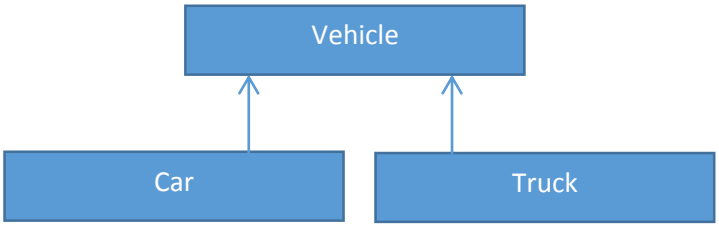
Object-Oriented Programming - Further Studies

Topic 11 – Exercise 2 – Constructors



Create a Java program to model the class relationships as depicted above.

The **Vehicle** class should have the following instance variables (private scope):

- Manufacturer (String)
- Engine Size (float)
- Registration Number (String)
- NCT (boolean)

Setter and Getter methods should be created.

The Vehicle class should contain the following constructors.

public Vehicle(String manufacturer, float engineSize, String registrationNo, boolean NCT)
public Vehicle(String manufacturer, float engineSize, String registrationNo)
public Vehicle(String manufacturer, float engineSize)
public Vehicle(String manufacturer)
public Vehicle()

The Vehicle class should also contain the following instance methods:

- accelerate(): to return a String, "Accelerating….".
- brake(): to return a String, "Vehicle braking…".

Please turn over

The **Car** class does not have any unique instance variables of its own.

You will need to create the following constructors however.

public Car(String manufacturer, float engineSize, String registrationNo, boolean NCT)
public Car(String manufacturer, float engineSize, String registrationNo)
public Car(String manufacturer, float engineSize)
public Car(String manufacturer)
public Car()

We would like to keep track of the number of Car objects created from the Car class. Include the appropriate functionality within the Car class to achieve this.

The **Car** class should override the brake() method and return the String, "Car braking".

The **Truck** class should extend the Vehicle class and include an instance variable named isTrailer (a boolean – private scope). This variable determines whether the Truck has a trailer.

Include a setter and getter method for this variable.

Include the following constructors in the Truck class.

public Truck(String manufacturer, float engineSize, String registrationNo, boolean NCT)
public Truck(String manufacturer, float engineSize, String registrationNo)
public Truck(String manufacturer, float engineSize)
public Truck(String manufacturer)
public Truck()

Override the brake() method in the Truck class to return the following String, "Truck braking…".

Include a toString() method in the Truck class which:

- Calls the superclass version of toString().
- Includes the state of the boolean variable (isTrailer).

Please turn over

- Create a folder named **JFT11Ex2** with the following classes.
    - Vehicle
    - Car
    - Truck
    - Main

- In the **Main** class, create the following objects.

1: Car Object (use the four argument constructor)

- Call the following methods:
    - toString()
    - brake()
    - accelerate()

```
Manufacturer=Subaru Hatchback,
EngineSize=53.45,
RegistrationNo=89C4545,
NCT=true
Car braking...
Accelerating..
```

2: Car Object (Use the no-arg constructor and the setter and getter methods)

- Call the following methods:
    - toString()
    - brake()
    - accelerate()

```
Manufacturer=Toyota Carola,
EngineSize=45.44,
RegistrationNo=90C4343,
NCT=true
Car braking...
Accelerating..
```

3: Output to the console the number of Car objects produced

```
2 cars have been produced.
```

Please turn over

4: Truck Object (use the one argument constructor)

- Call the following methods:
  - toString()
  - brake()
  - accelerate()

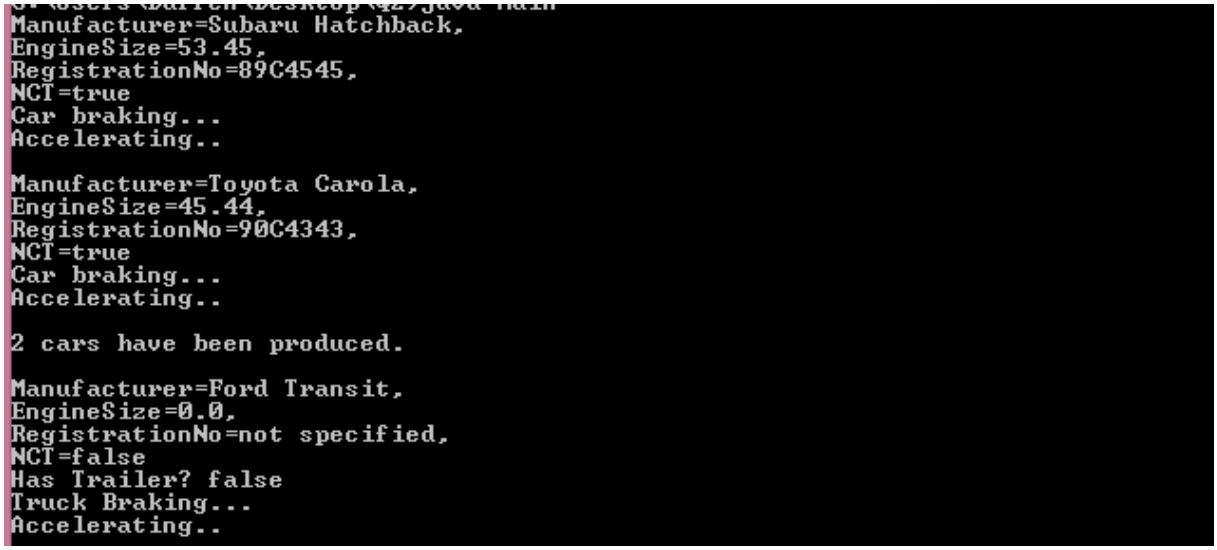
```
Manufacturer=Ford Transit,
EngineSize=0.0,
RegistrationNo=not specified,
NCT=false
Has Trailer? false
Truck Braking...
Accelerating..
```

5: Truck Object (Use the no-arg constructor and the setter and getter methods)

- Call the following methods:
  - toString()
  - brake()
  - accelerate()

```
Manufacturer=Hyundai Pick-Up,
EngineSize=89.45,
RegistrationNo=99C5344,
NCT=true
Has Trailer? false
Truck Braking...
Accelerating..
```

The full program output is shown overleaf.



End of Exercise