

Credit Name: CSE 3130 - Object Oriented Programming 2

Assignment: Vehicle

How has your program changed from planning to coding to now? Please Explain

```
abstract class Vehicle
{
    private double fuelEconomyCity;
    private double fuelEconomyHwy;
    private int seatCapacity;
    private double cargoVolume;

    Vehicle(double fuelCity, double fuelHwy, int seatCap, double cargoVol)
    {
        fuelEconomyCity = fuelCity;
        fuelEconomyHwy = fuelHwy;
        seatCapacity = seatCap;
        cargoVolume = cargoVol;
    }
}
```

We make an abstract superclass Vehicle because a vehicle on its own is not a real object but a type of object, so we make it with general parameters and variables like its fuel economy in the city and highway as well as its seat capacity and how much cargo space there is.

We set up the constructor with variables initialized with the relevant parameters.

```

public double getFuelEconomyCity()
{
    return fuelEconomyCity;
}

public double getFuelEconomyHwy()
{
    return fuelEconomyHwy;
}

public int getSeatCapacity()
{
    return seatCapacity;
}

public double getCargoVolume()
{
    return cargoVolume;
}

```

We create a series of methods each of which return the respective component of the vehicle's information.

```

public String toString() {
    return "\nFuel Economy (City): " + fuelEconomyCity + " mpg" +
        "\nFuel Economy (Highway): " + fuelEconomyHwy + " mpg" +
        "\nSeating Capacity: " + seatCapacity +
        "\nCargo Volume: " + cargoVolume + " cubic ft";
}

```

We create a toString method and structure and how the different variables will display by concatenating our variables with strings.

```

abstract String getVehicleType();

```

Lastly, I made a abstract method which returns the type of vehicle something is. Every type of vehicle will have a completely different return statement, inspiring the use of the abstract method.

```
public class Car extends Vehicle
{
    private boolean isSportsCar;
    private boolean isConvertible;
    private int horsepower;

    public Car(boolean sportsCar,boolean convertible,int hp,double fuelCity, double fuelHwy, int seatCap, double cargoVol)
    {
        super(fuelCity,fuelHwy,seatCap,cargoVol);

        isSportsCar = sportsCar;
        isConvertible = convertible;
        horsepower = hp;
    }
}
```

Our first subclass is car and it inherits the vehicle class. In addition to all of vehicle's parameters,this class has 3 additional variables which check if the car is a sports car, if its convertible and its horsepower, 2 boolean and one int variable. We initialize these variables using the parameters in the constructor.

```
public boolean checkSportsCar()
{
    return isSportsCar;
}

public boolean checkConvertible()
{
    return isConvertible;
}

public int getHP()
{
    return horsepower;
}
```

We create methods which check if the car is a sports car or convertible returning either true or false. Then we have a method which returns the horsepower of the car.

```
public String getVehicleType()
{
    String str = "\nVehicle Type: Car";
    return str;
}
```

We then defined the abstract method and return that the vehicle is a car.

```
public String toString()
{
    String str = "\nHorsepower: " + horsepower + "hp";

    if(checkSportsCar())
    {
        str += "\nThe car IS a Sports Car";
    }
    else
    {
        str += "\nThe car is NOT a Sports Car";
    }

    if(checkConvertible())
    {
        str += "\nThe car IS a Convertible";
    }
    else
    {
        str += "\nThe car is NOT a Convertible";
    }

    str += super.toString();

    return str;
}
```

We override the vehicle's toString method and add the horsepower and we highlight whether or not the car is a sports car or convertible using 2 if-else statements.

```
public class Truck extends Vehicle
{
    private boolean has4WD;
    private double towCapacity;
    private double bedLength;

    public Truck(boolean fourWD,double towCap,double bLength,double fuelCity, double fuelHwy, int seatCap, double cargoVol)
    {
        super(fuelCity,fuelHwy,seatCap,cargoVol);

        has4WD = fourWD;
        towCapacity = towCap;
        bedLength = bLength;
    }
}
```

Our next subclass is the truck subclass, which has 3 extra variables which determine if it has four wheel drive, determine its tow capacity and lastly the length of its bed. All are initialized by the parameters in the constructor.

```
public boolean check4WD()
{
    return has4WD;
}

public double getTowCapacity()
{
    return towCapacity;
}

public double getBedLength()
{
    return bedLength;
}

public String getVehicleType()
{
    String str = "\nVehicle Type: Truck";
    return str;
}
```

Similarly we have methods which return the tow capacity, the bed length, check if there is four wheel drive. Then we also have the vehicle type method which indicates the vehicle is a truck.

```

public String toString()
{
    String str = "\nTowing Capacity: " + towCapacity+ "lbs\nBed Length: " + bedLength + "ft";
    if(check4WD())
    {
        str += "\nThe Truck HAS Four-Wheel-Drive";
    }
    else
    {
        str += "\nThe Truck does NOT have Four-Wheel-Drive";
    }

    str += super.toString();

    return str;
}

```

Again the toString method is overridden and the four wheel drive feature's presence is indicated by using an if else statement. The other truck features are added to the string and the rest is generated by the superclass's toString method.

```

public class Minivan extends Vehicle
{
    private boolean hasDVDSysyem;

    public Minivan(boolean DVDSysyem,double fuelCity, double fuelHwy, int seatCap, double cargoVol)
    {
        super(fuelCity,fuelHwy,seatCap,cargoVol);
        hasDVDSysyem = DVDSysyem;
    }
}

```

In our last subclass we are creating a minivan class and the only additional variable we are adding is a boolean variable which indicates if there is a DVD entertainment system.

```

public boolean checkDVDSysyem()
{
    return hasDVDSysyem;
}

public String getVehicleType()
{
    String str = "\nVehicle Type: Minivan";
    return str;
}

```

We have a method which returns true or false depending on if there is a DVD system. Then we have the method which indicates that our vehicle is a minivan.

```
public String toString()
{
    String str = "";
    if(checkDVDSystem())
    {
        str += "\nThe Minivan HAS a DVD Entertainment System, ";
    }
    else
    {
        str += "\nThe Minivan does NOT have a DVD Entertainment System, ";
    }

    str += super.toString();

    return str;
}
```

The toString() method is done in the same way as the other subclasses.

The createVehicle is our tester class and it creates and initializes 3 objects, a car, truck and a minivan and displays their features indicating the differences between the vehicle types.


```

public static void main(String[] args)
{
    Car coolCar = new Car(true,false,250,25,35,2,15);

    Truck coolTruck = new Truck(true,1000,5.5,20,25,6,20);

    Minivan coolMinivan = new Minivan(false,17,22,8,30);

    for(int i = 0;i<200;i++)
    {
        System.out.print("=");
    }

    System.out.println(coolCar.getVehicleType());
    System.out.println(coolCar);

    for(int i = 0;i<200;i++)
    {
        System.out.print("=");
    }

    System.out.println(coolTruck.getVehicleType());
    System.out.println(coolTruck);

    for(int i = 0;i<200;i++)
    {
        System.out.print("=");
    }

    System.out.println(coolMinivan.getVehicleType());
    System.out.println(coolMinivan);
}

```

The for statements highlight the separation between the 3 vehicles details. We print what type of vehicle each one is and then display the toString() method for each object. We do not need to explicitly state coolCar.toString() for that class or any other class because when an object is passed through a print statement it automatically calls the toString method.

```
=====
Vehicle Type: Car
```

```
Horsepower: 250hp
```

```
The car IS a Sports Car
```

```
The car is NOT a Convertible
```

```
Fuel Economy (City): 25.0 mpg
```

```
Fuel Economy (Highway): 35.0 mpg
```

```
Seating Capacity: 2
```

```
Cargo Volume: 15.0 cubic ft
=====
```

```
Vehicle Type: Truck
```

```
Towing Capacity: 1000.0lbs
```

```
Bed Length: 5.5ft
```

```
The Truck HAS Four-Wheel-Drive
```

```
Fuel Economy (City): 20.0 mpg
```

```
Fuel Economy (Highway): 25.0 mpg
```

```
Seating Capacity: 6
```

```
Cargo Volume: 20.0 cubic ft
=====
```

```
Vehicle Type: Minivan
```

```
The Minivan does NOT have a DVD Entertainment System,
```

```
Fuel Economy (City): 17.0 mpg
```

```
Fuel Economy (Highway): 22.0 mpg
```

```
Seating Capacity: 8
```

```
Cargo Volume: 30.0 cubic ft
```

This is a visual of the output.