Credit Name: Inheritance and Polymorphism
Assignment Name: Vehicle, Car, Truck, Minivan

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

```
private double fuelEconomyCity, fuelEconomyHwy, cargoVolume, maxSpeed;
private int seatingCapacity;
private boolean hasHeatedSeats, hasChargingPorts;
```

I started out by creating the 4 instance variables for Vehicle that were provided as well as adding a couple of my own.

```
public Vehicle(double FEC, double FEH, int SC, double CV, boolean HHS,boolean HCS, double MS)
{
    fuelEconomyCity = FEC;
    fuelEconomyHwy = FEH;
    seatingCapacity = SC;
    cargoVolume = CV;
    hasHeatedSeats = HHS;
    hasChargingPorts = HCS;
    maxSpeed = MS;
}
```

Next I created the constructor to initialize all of the variables in the Vehicle class.

```
public void setFuelEconomyCity(double FEC)
{
    fuelEconomyCity = FEC;
}
public void setFuelEconomyHwy(double FEH)
{
    fuelEconomyHwy = FEH;
}
public void setSeatingCapacity(int SC)
{
    seatingCapacity = SC;
}
public void setCargoVolume(double CV)
{
    cargoVolume = CV;
}
public void setHasHeatedSeats(boolean HHS)
{
    hasHeatedSeats = HHS;
}
public void setHasChargingPorts(boolean HCS)
{
    hasChargingPorts = HCS;
}
public void setMaxSpeed(double MS)
{
    maxSpeed = MS;
}
```

Next I created the modifier methods for all of the variables.

```java
//Accessor methods
public double getFuelEconomyCity()
{
    return fuelEconomyCity;
}
public double getFuelEconomyHwy()
{
    return fuelEconomyHwy;
}
public int getSeatingCapacity()
{
    return seatingCapacity;
}
public double getCargoVolume()
{
    return cargoVolume;
}
public boolean getHasHeatedSeats()
{
    return hasHeatedSeats;
}
public boolean getHasChargingPorts()
{
    return hasChargingPorts;
}
public double getMaxSpeed()
{
    return maxSpeed;
}
```

Next I created the accessor methods for the variables.

```java
//Return object as string
public String toString()
{
    String veh = "City fuel economy: " + fuelEconomyCity
            + "\nHighway fuel economy: " + fuelEconomyHwy
            + "\nSeating capacity: " + seatingCapacity
            + "\nCargo volume: " + cargoVolume
            + "\nHas heated seats: " + hasHeatedSeats
            + "\nHas charging ports: " + hasChargingPorts
            + "\nMax speed: " + maxSpeed;

    return veh;

}
```

Finally I created the toString method so I could display the entire object later in the test code. I then moved on to working on the other 3 subclasses.

```
double bedSize;
double towingCapacity;


//Constructor
public Truck(double FEC, double FEH, int SC, double CV, boolean HHS, boolean HCS, double MS, double BS, double TC)
{
    super(FEC, FEH, SC, CV, HHS, HCS, MS);
    bedSize = BS;
    towingCapacity = TC;
}
```

I moved onto the truck and created 2 variables that are specific to a truck, its bed size and its towing capacity. I also created the constructor for this class, setting the values of every variable.

```
//Modifier methods
public void setBedSize(double BS)
{
    bedSize = BS;
}
public void setTowingCapacity(double TC)
{
    towingCapacity = TC;
}


//Accessor methods
public double getBedSize()
{
    return bedSize;
}
public double getTowingCapacity()
{
    return towingCapacity;
}
```

Then I created the accessor and modifier methods for both of the new variables I added.

```
//Return object as string
public String toString()
{
    return (super.toString()
            + "\nBed size: " + bedSize
            + "\nTowing Capacity: " + towingCapacity);
}
```

Finally I finished off Truck by creating the toString method, using super.toString() to add the other variables from the Vehicle class.

```java
public class Car extends Vehicle
{
    boolean hasTrunkSensor, hasRoofRack;


    //Constructor
    public Car(double FEC, double FEH, int SC, double CV, boolean HHS, boolean HCS, double MS, boolean HTS, boolean HRR)
    {
        super(FEC, FEH, SC, CV, HHS, HCS, MS);
        hasTrunkSensor = HTS;
        hasRoofRack = HRR;
    }




    //Modifier methods
    public void setHasTrunkSensor(boolean HTS)
    {
        hasTrunkSensor = HTS;
    }
    public void setHasRoofRack(boolean HRR)
    {
        hasRoofRack = HRR;
    }


    //Accessor methods
    public boolean getHasTrunkSensor()
    {
        return hasTrunkSensor;
    }
    public boolean getHasRoofRack()
    {
        return hasRoofRack;
    }



    //Return object as string
    public String toString()
    {
        return (super.toString()
                + "\nBed size: " + hasTrunkSensor
                + "\nTowing Capacity: " + hasRoofRack);
    }
```

I then moved onto the Car class, copying the same routine of creating the variables, constructor, accessor and modifier methods, and finally the toString method.

```java
public class Minivan extends Vehicle
{
    private boolean hasSlidingDoors;
    private boolean hasEntertainmentScreen;

    public Minivan(double FEC, double FEH, int SC, double CV, boolean HHS, boolean HCS, double MS, boolean HSD, boolean HES)
    {
        super(FEC, FEH, SC, CV, HHS, HCS, MS);
        hasSlidingDoors = HSD;
        hasEntertainmentScreen = HES;
    }



    //Modifier methods
    public void setHasSlidingDoors(boolean HSD)
    {
        hasSlidingDoors = HSD;
    }
    public void setHasEntertainmentScreen(boolean HES)
    {
        hasEntertainmentScreen = HES;
    }


    //Accessor methods
    public boolean getHasSlidingDoors()
    {
        return hasSlidingDoors;
    }
    public boolean getHasEntertainmentScreen()
    {
        return hasEntertainmentScreen;
    }



    //Return object as string
    public String toString()
    {
        return (super.toString()
                + "\nHas Sliding Doors: " + hasSlidingDoors
                + "\n Has Entertainment Screen: " + hasEntertainmentScreen);
    }
```

Finally I did the same thing for Minivan adding variables for sliding doors and for if there's an entertainment screen. I then moved on to working on the client code.

My idea for testing this code was to allow the user to enter statistics of their personal vehicles and the program would store and display all of them. To test both the toString and accessor methods I would also allow them to add a second vehicle that would test a different way of displaying the data.

```java
public static void main(String[] args)
{
    int vehicleType;
    boolean otherVehicle = false;
    Scanner input = new Scanner(System.in);


    //Loop if there's a second vehicle
    do
    {
        //Menu
        System.out.println("Which Vehicle do you have?"
                + "\n1. Car"
                + "\n2. Truck"
                + "\n3. Minivan");

        //Get vehicle type
        System.out.println("Please enter the vehicle number: ");
        vehicleType = input.nextInt();

    } while(otherVehicle);
}
```

I started out the test code by creating the 3 variables I knew I would need for the rest of the program and followed it with a do while loop. Inside the loop would be where the menu, choices, and processing would occur. I started with the menu for the type of vehicle and the code that would let the user pick one.

```java
//Check for vehicle
switch (vehicleType)
{
//User has a car
case 1:

    break;



//User has a truck
case 2:

    break;



//User has a minivan
case 3:

    break;
}
```

Next I started working on a switch statement to write code for specific vehicle features.

```java
//Get fuelEconomyCity
System.out.println("Please enter the vehicle's city fuel economy: ");
fuelEconomyCity = input.nextDouble();

//Get fuelEconomyHwy
System.out.println("Please enter the vehicle's highway fuel economy: ");
fuelEconomyHwy = input.nextDouble();

//Get seatingCapacity
System.out.println("Please enter the vehicle's seating capacity: ");
seatingCapacity = input.nextInt();

//Get cargoVolume
System.out.println("Please enter the vehicle's cargo volume: ");
cargoVolume = input.nextDouble();

//Get hasHeatedSeats
System.out.println("Does the vehicle have heated seats? (true or false): ");
hasHeatedSeats = input.nextBoolean();

//Get hasChargingPorts
System.out.println("Does the vehicle have charging ports? (true or false): ");
hasChargingPorts = input.nextBoolean();

//Get maxSpeed
System.out.println("Please enter the vehicle's maximum speed: ");
maxSpeed = input.nextDouble();


//Check for vehicle
switch (vehicleType)
```

Above the switch statement I also added code to ask the user about the variables that apply to every vehicle so I wouldn't have to add repeating code.

```java
//User has a car
case 1:

    //Get hasTrunkSensor
    System.out.println("Does the vehicle have an automatic trunk opening sensor? (true or false): ");
    boolean hasTrunkSensor = input.nextBoolean();

    //Get hasRoofRack
    System.out.println("Does the vehicle have a roof rack? (true or false): ");
    boolean hasRoofRack = input.nextBoolean();


    //Create Car object
    Car userCar = new Car(fuelEconomyCity, fuelEconomyHwy, seatingCapacity, cargoVolume, hasHeatedSeats, hasChargingPorts, maxSpeed, hasTrunkSensor, hasRoofRack);
    break;
```

For each case I added 2 more questions and variables to store answers depending on which vehicle they picked and created an object using all of the information that the user has entered.

```
//Check for which way to display and store the vehicle's data
if(otherVehicle == false)
{
    userCar = new Car(fuelEconomyCity, fuelEconomyHwy, seatingCapacity, cargoVolume, hasHeated$
    System.out.println("\nYour truck's data:\n" + userCar);
}
else
{
    //Add data to new truck object
    userCar = new Car(0, 0, 0, 0, false, false, 0, false, false);
    userCar.setFuelEconomyCity(fuelEconomyCity);
    userCar.setFuelEconomyHwy(fuelEconomyHwy);
    userCar.setSeatingCapacity(seatingCapacity);
    userCar.setCargoVolume(cargoVolume);
    userCar.setHasHeatedSeats(hasHeatedSeats);
    userCar.setHasChargingPorts(hasChargingPorts);
    userCar.setMaxSpeed(maxSpeed);
    userCar.setHasTrunkSensor(hasTrunkSensor);
    userCar.setHasRoofRack(hasRoofRack);

    //Display data
    System.out.println("\nYour truck's data:"
            + "\nFuel Economy (City): " + userCar.getFuelEconomyCity()
            + "\nFuel Economy (Highway): " + userCar.getFuelEconomyHwy()
            + "\nSeating capacity: " + userCar.getSeatingCapacity()
            + "\nCargo volume: " + userCar.getCargoVolume()
            + "\nHas heated seats: " + userCar.getHasHeatedSeats()
            + "\nHes charging ports: " + userCar.getHasChargingPorts()
            + "\nMax speed: " + userCar.getMaxSpeed()
            + "\nHas trunk sensor: " + userCar.getHasTrunkSensor()
            + "\nHas roof rack: " + userCar.getHasRoofRack());
}
break;
```

I decided to change how I was going to store and display the data by creating an if statement to check if the vehicle is the first one or not. If it is the first vehicle then the program uses the parameters to store the data and the toString method to display the data. If it's the second vehicle then the accessor and modifier methods are tested. I then applied this to all 3 vehicle options.

```
//Ask the user if they have any other cars
System.out.println("Is there another vehicle you would like to enter data for? (true or false): ");
otherVehicle = input.nextBoolean();

while(otherVehicle);
```

Finally at the end of the loop I wrote code to ask the user if they have another vehicle they would like to enter data for and if they don't then the program ends.