

```
import java.io.Serializable;

public abstract class Vehicle implements Serializable{

    private int ID;
    protected boolean rented;
    protected int nbHours;
    protected double rentalAmount;

    public Vehicle(int id){
        this.ID = id;
        nbHours = 10;
    }

    public abstract double computeRentalAmount();

    public void display(){
        System.out.println("Vehicle ID: " + ID);
        System.out.println("Is rented?: " + rented);
        System.out.println("Number of hours: " + nbHours);
        System.out.println("Rental Amount: " +
rentalAmount);
    }

    public int getNbHours(){
        return nbHours;
    }

}
```

```
public class Car extends Vehicle{

    private double dailyRate;
    private int mileage;

    public Car(int ID, double rate, int mileage){
        super(ID);
        this.dailyRate = rate;
        this.mileage = mileage;
    }

    public double getDailyRate() {
        return dailyRate;
    }

    public int getMileage() {
        return mileage;
    }

    public double computeRentalAmount(){
        int nbDays = nbHours/24;
        if(nbHours % 24 > 0) nbDays++;
        if (rented) rentalAmount = dailyRate * nbDays;
        return rentalAmount;
    }

}
```

```
public class Truck extends Vehicle{

    private double hourlyRate;

    public Truck(int ID, double rate){
        super(ID);
        this.hourlyRate = rate;
    }

    public double getHourlyRate() {
        return hourlyRate;
    }

    public double computeRentalAmount(){
        if(rented) rentalAmount = nbHours * hourlyRate;
        return rentalAmount;
    }

}
```

```

public class Branch {

    private String name;
    private Vehicle arrVeh[];
    private int nbVeh;

    public Branch(String name, int size){
        this.name = name;
        arrVeh = new Vehicle[size];
        nbVeh = 0;
    }

    public boolean addVehicle(Vehicle v){
        if(nbVeh >= arrVeh.length)
            return false;
        if(v instanceof Car)
            arrVeh[nbVeh++] = (Car) v;
        else
            arrVeh[nbVeh++] = (Truck) v;
        return true;
    }

    public double sumRentedCars(int mil){
        double sum = 0;
        for(int i = 0; i < nbVeh; i++){
            if(arrVeh[i] instanceof Car &&
((Car)arrVeh[i]).getMileage() < mil)
                sum += arrVeh[i].computeRentalAmount();
        }
        return sum;
    }
}

```

```

public void saveToFile(int nbH, double dailyR)
    throws IOException{

    File f =
new File("C:/Users/akalshememry/Desktop/cars.data");
    FileOutputStream outputStream =
new FileOutputStream(f);
    ObjectOutputStream outCar =
new ObjectOutputStream(outputStream);

    File f2 =
new File("C:/Users/akalshememry/Desktop/Trucks.data");
    FileOutputStream outputStream2 =
new FileOutputStream(f2);
    ObjectOutputStream outTruck =
new ObjectOutputStream(outputStream2);

    for(int i = 0; i < nbVeh; i++){
        if(arrVeh[i] instanceof Car
&& ((Car)arrVeh[i]).getDailyRate() == dailyR)
            outCar.writeObject(arrVeh[i]);
        else if(arrVeh[i] instanceof Truck
&& arrVeh[i].getNbHours() > nbH)
            outTruck.writeObject(arrVeh[i]);
    }
}

```

```
import java.io.IOException;

public class testBranch {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Branch b = new Branch("Test", 4);

        b.addVehicle(new Car(1111, 30.0, 50000));
        b.addVehicle(new Car(2222, 30.0, 30000));
        b.addVehicle(new Truck(3333, 50));
        b.addVehicle(new Truck(4444, 100));

        try {
            b.saveToFile(5, 30.0);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        System.out.println("Bye!");
    }
}
```