Class Vehicle

```
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;
	}
	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;
	}
}
```

Class Car

```
public class Car extends Vehicle{
	private double dailyRate;
	private int milage;

	public Car(int id, double dailyRate, int milage) {
		super(id);
		this.dailyRate = dailyRate;
		this.milage = milage;
	}
	public double getDailyRate() {
		return dailyRate;
	}
	public int getMilage() {
		return milage;
	}
	public double computeRentalAmount(){
		int nbDays = nbHours/24;
		if(nbHours%24 > 0) nbDays++;
		if(rented) rentalAmount = nbDays * dailyRate;
		return rentalAmount;
	}
}
```

Class Truck

```
public class Truck extends Vehicle{
	private int hourlyRate;
	public Truck(int id, int hourlyRate) {
		super(id);
		this.hourlyRate = hourlyRate;
	}
	public int getHourlyRate() {
		return hourlyRate;
	}
	public double computeRentalAmount(){
		rentalAmount = hourlyRate * nbHours;
		return rentalAmount;
	}
}
```

Class Branch

```
import java.io.*;
public class Branch {
	private String name;
	private Vehicle vehicles[];
	private int nbV;
	public Branch(String name, int size) throws Exception{
		if(size <= 0) throw new Exception("Invalid Size");
		this.name = name;
		vehicles = new Vehicle[size];
		nbV = 0;
	}
	public boolean addVehicle(Vehicle v){
		if(nbV >= vehicles.length) return false;
		vehicles[nbV++] = v;
		return true;
	}
	public double sumRentedCars(int mil){
		int sum = 0;
		for(int i = 0; i < nbV; i++)
	if(vehicles[i] instanceof Car && ((Car)vehicles[i]).getMilage() < mil)
				sum += vehicles[i].computeRentalAmount();
		return sum;
	}
	public void saveToFile(int nbH, double dailyR) throws IOException{

	File f = new File("Cars.data");
	FileOutputStream outStream = new FileOutputStream(f);
	ObjectOutputStream outCars = new ObjectOutputStream(outStream);
```

```
      File f2 = new File("Trucks.data");
      FileOutputStream outStream2 = new FileOutputStream(f2);
      ObjectOutputStream outTrucks = new ObjectOutputStream(outStream2);

            for(int i = 0; i < nbV; i++){
if(vehicles[i] instanceof Car && ((Car)vehicles[i]).getDailyRate() == dailyR)
                        outCars.writeObject(vehicles[i]);
else if(vehicles[i] instanceof Truck && vehicles[i].getNbHours() > nbH)
                        outTrucks.writeObject(vehicles[i]);
            }
            outCars.close();
            outTrucks.close();
            outStream.close();
            outStream2.close();
      }
}
```

Class Test

```
import java.io.IOException;
public class test {
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            try{
                  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) {
                        System.out.println(e);
                  }

            } catch(Exception e){
                  System.out.println(e);
            }
            System.out.println("Bye!");
      }

}
```