 Use exactly same class names as mentioned

 Use exactly same method signature (method name, return type, method parameter type, position of each method parameter)

 Define attributes with same name and data type as given in class outline.

 Define constructors and getter setters as given in the class outline.

 Ensure attributes are private and other methods which will be called from main method, getter-setter methods and constructor are public.

 Use main method only for input, output, object creation and method invocation.

 Within the given time, mail it.

**As mentioned above, any logic which may be 100% correct is not valid if above points are not taken care. Hence, simply building logic does not certify us as project ready. Building exact and complete solution does.**

**Problem Statement:**

People of a Residential Apartment planned to monitor the vehicle movement details of that apartment. Develop a Parking Management system based on the given requirements.

1. Create a class **Vehicle** with the following attributes.

**regNumber** - String, **ownerName** - String, **mobileNumber** – long

Generate constructors, getters and setters

2. Create a class **ResidentVehicle** with following attributes. Inherit this class from Vehicle class

**flatNumber** - int, **parkingStatus** – boolean (Parked / Not parked) Generate constructors, getters and setters

3. Create a class **VisitorVehicle** with following attributes. Inherit this class from Vehicle class.

**visitingFlatNumber** - int, **inTime** – int, **outTime –** int

Generate constructors**(without outTime)**, getters and setters

4. Create an exception class “VehicleNotFoundException”

5. Create an exception class “ParkingSlotNotAvailableException”

6. Create a class ParkingManagement with following members

**allVehicles** - array list of objects of Vehicle class

**addVehicle(Vehicle vehicle)** method - This method accepts either ResidentVehicle object or VisitorVehicle object as argument, and stores it in allVehicles arraylist and return the message in the format “Vehicle parked at Parking Slot no : 2”. Only 10 Vehicles can be parked. If no parking slots are free throw ParkingSlotNotAvailableException with message “No Parking Slots Available”.

**setVisitorVehicleOutTime(String regNumber, int outTime)** - This method accepts registration number of a vehicle and out time as input values. The method searches allVehicles arraylist, for the vehicle with the given register number and if it is VisitorVehicle type, then updates its out time and return the message “Vehicle with Reg Number : TN40A1212 updated successfully” .

If the vehicle not found, this method throws VehicleNotFoundException with message

“Searched Vehicle Not Found: TN40A1212”

**getParkedResidentVehicleCount()** - This method searches allVehicles array and counts the number of ResidentVehicle parked(parked status is true).

**displayAllVehicles()** - This method displays regNumber, owenerName and mobileNumber of all the vehicles.

**main() -** In the main method of the program,

1. Create objects for the ResidentVehicle and VisitorVehicle with the given sample data.

2. Test all the methods and display the correct output.

**Sample Input: Resident Vehicles**

Registration

Number Owner Name Mobile Number Flat Number Parking Status

TN49A2222 Senthil 1212121212 201 true

TN49B4444 Ragavan 1414141414 202 false

TN49C6666 Srimathi 1616161616 204 true

**Visitor Vehicles**

Registration

Number Owner Name Mobile Number

Visiting Flat

Number In Time

TN49A2222 Senthil 1212121212 201 9

TN49B4444 Ragavan 1414141414 202 8

TN49C6666 Srimathi 1616161616 204 15

**The overview of the project is given below.**

