COIT11134 Object Oriented Programming (T2 2020) Assignment 1

Due date: Friday of Week 6 (26th August 2020)
Weighting: 15%
Length: NA

1. Objectives

This assignment is designed to test your ability in applying object-oriented programming techniques. By completing this assignment, you demonstrate that you can:

- Implement object-oriented programs using a modern programming language
- Build interactive software applications using Graphical User Interface components

2. Assessment Task

Your task in this assessment is to write a GUI application named MVR to perform Motor Vehicle Registration (MVR) for Queensland Road and Transport Authority who is your client. You must provide your client with a complete GUI desktop application that satisfies all the requirements described in the System Specification section in this document.

3. System Specifications

Queensland Road and Transport Authority (QRTA), as part of Queensland Government currently offers road and transport services to the road users. This service is dedicated to collecting, storing, manipulating, and making available Motor Vehicle Registration information important to the delivery of services to Queensland road users. Each motor vehicle can only have one owner and will have a unique identifier (plate number) assigned by the system when registered for the first time. The system is used by the QRTA staff to Create, Edit, and Search records for all vehicle in QLD. The system should have the ability to add update/change the vehicle and owner information.

3.1 Owner Information:

There are 2 types of vehicle owners (private, corporate). Each owner must have all or some of the following information according to its type.

- Id (licence number),
- Australian Business Number (only for corporate owners)
- Date of birth (only for private owners)
- First Name,
- Last Name,
- Address,
- Phone number.

3.2 Vehicle Information:

There are three types of vehicles (Motorcycle, Light vehicle, heavy vehicle).
Each vehicle must have all or some of the following information according to its type.

- Plate number,
- Load capacity, (only for heavy vehicles)
- Number of seats, (only for light vehicles)
- Engine capacity,
- Make,
- Model,
- Year,
- owner Id.

The system should be developed using good practices in Object-oriented Java programming (commenting your code, proper variable/method/class naming, using inheritance, encapsulation, etc). the system will be a **Java desktop application** which runs on a windows operating system with a graphical user interface.

3.3 System Functionality

The system should enable the user to complete the following tasks:

1. **Create new owner record;** Get the data entry about a vehicle owner, create an object from the data and store the object in an ArrayList.
2. **Create a new vehicle record;** Get the data entry about a vehicle, create an object from the data and store it in another ArrayList.
3. **Search and display existing owner;** Get a licence number from the user and search the owner ArrayList for that that owner, display the owner information if found, display error message if not found.
4. **Edit Existing owner information:** if the owner is found and displayed, user must be able to change the information and save it back to the owner ArrayList
5. **Search and display existing vehicle;** Get a plat number from the user and search the vehicle ArrayList for that vehicle, display the vehicle information if found, display error message if

not found.

6. **Edit Existing vehicle information;** if the vehicle is found and displayed, user must be able to change the information and save it back to the vehicle ArrayList
7. **Clear button;** there must be a "Clear" Button in every data entry screen that can clear all the information from the entry fields on the screen.
8. **Exit button;** there must be an exit button always present that allow the user to gracefully exit the system (ask the user if they really want to exit)
9. **All input data must be validated;** Error messages are displayed in the event of improper data type, length, or range.

4. Recommended Program Guidelines

You should design and build an interactive graphical user interface which is user friendly and easy to use.

4.1 Object-Oriented Design:

Your program must use inheritance by using the following classes:
A generic Vehicle class
Two derived classes for Light Vehicle and Heavy Vehicle
A generic Owner class
Two derived classes for Individual and Corporate owner.
Your program can use other classes should your design require them.

User should be able to enter more than one vehicle or owner details at a time. These entries should be stored in appropriate Array List. The data is not stored in a database, and therefore, each time fresh data should be entered to test the system.

4.2 Data validation:

You should validate data entered by the user using the following guidelines:
Names should only contain alphabets (words separated by space)
Phone number (Alpha numeric)
Date of birth (Alpha numeric) (no need to use Date class in this assignment)
Plate number (Alpha numeric)
You should include default, parameterised, and copy constructors, get, set methods, and properly written toString() method in all your classes.

4.3 Source Code Quality

Your source code should be written following quality coding styles, with file header comments, comments for variables, and methods, meaningful variable names, and appropriate indentation.

5 Submission

You should submit the following:

One zip/archive file containing complete **NetBeans** project files (**you must use NetBeans IDE, version 8.2 or above**), including any additional external library or jar files used by your application, if any.

Important Note:

- If your program fails to run, partial marks will be given to all functional criteria items:(3 to 9),
- If you submit an interactive Java application without any GUI controls, you will be given only partial marks.
- In order to avoid plagiarism, complete list of classes, fields, methods and GUI layout have not been provided specifically in this assignment so that you are encouraged to develop your own classes appropriately.

6 Marking Guide

COIT11134	Assignment1	**Important Notice**		
StudentId:		**If your program fails to compile or run, partial marks will be given to all functional criteria items:(3 to 9)**		
Student Name:				
Item	**Description**	**Maximum marks**	**Your marks**	**Comments**
1	Used Inheritance and/or aggregation	1		
2	Displays necessary/appropriate GUI controls for data entry (good layout, user friendly)	1.5		
3	Creates and stores Owner details in an ArrayList	1		
4	Creates and stores Vehicle details in an ArrayList	1		
5	Search and display existing owner	1		
6	Search and display existing vehicle	1		
7	Edit Existing owner information	1		
8	Edit Existing vehicle information	1		
9	Clear button works (0.5) Exit button works (0.5)	1		
10	Validated inputs (5.0) and displayed error messages (5.0)	1		
11	Classes' implementation complete and adequate: Suitable fields and appropriate data types (1) Appropriate constructors (1) Appropriate methods (1)	3		
12	Quality: Code quality (meaningful names, indentation, comments, etc.,)	1.5		
13	Less late penalty (5% per day)			
14	Less Plagiarism			
Total		15		