

Coursework Description

Objectives:

The aim of this coursework is to assess the knowledge and skills that you have acquired about object-oriented programming during the module. You are asked to implement a program in which objects interact in order to fulfil with a set of functional requirements.

Analyse the problem statement:

An important skill that you will start to develop in this module is analysing a problem statement in order to identify the details needed to develop a solution.

In this assignment, the first task you should perform is a careful analysis of the problem statement in order to make sure you have all the information to elaborate a solution.

Design a solution

The design of your system should be consistent with the Object Oriented principles and easy to understand by an independent programmer.

You are required to design your program using UML diagrams. In particular you have to draw:

- A use cases for the system (6 marks).
- A class diagram (6 marks)

Problem description and requirement statement

You are required to develop a program that implements a basic vehicle rental system.

You should implement a **console system** from where the manager can add new vehicles, delete if needed, edit the vehicle information, etc. (as described in detailed below).

You should implement a **Graphical User Interface** (GUI) from where a customer can see the list of vehicles and book available vehicles for specific dates.

In this assignment, you will be required to implement the following functionality:

1. According to the *Inheritance* principle you have to design and implement a super class **Vehicle** (6 marks) and the subclasses **Car** and **Motorbike**. The classes should include appropriate methods in order to comply with the *encapsulation principle* and hold information about the *Plate number*, and the *Make* (You can add any other information that you consider appropriate for a rental vehicle system and you can implement additional classes with justification to make the code more robust or user friendly).

In particular:

- The **Car** class should hold specific information and methods. You should add at least 2 instance variables (attributes that you believe are important to be held in this class) and the relative get/set methods (5 marks).
 - The **Motorbike** class should hold specific information and methods. You should add at least 2 instance variables (attributes) and the relative get/set methods (5 marks).
 - You should implement a class **Schedule** to represent the time slot when a vehicle has been booked. The class should have a pick-up date and a drop-off date (to represent the date you can use either the class provided during tutorials or you can use any java API). (5 marks).
2. Design and implement a class called **WestminsterRentalVehicleManager**, which implements the interface **RentalVehicleManager** (2 marks). WestminsterRentalVehicleManager maintains the list of the vehicle to rent and provides all the methods for the system manager.

The class should display **in the console a menu** containing the following management actions from which the manager can select one.

- **Add a new vehicle** in the rental system. It should be possible to add a new vehicle, either a car or a motorbike with all the relevant information. You should consider that in the rental car park there are max 50 parking lots (5 marks).