- **Delete a vehicle** from the system, selecting the plate number. Display a message with the information of the vehicle that has been deleted (if it is a car or motorbike) and the number of available parking lots in the garage *(5 marks)*.

- **Print the list of the vehicles** in the system. For each vehicle, print the plate number, the type of vehicle (if is a car, a van or a motorbike). The list should be ordered alphabetically according to the vehicle make *(5 marks)*.

- **Write/Save** the vehicle stock list in a file after any changes (5 marks).

Create a Graphical User Interface (GUI) that can be opened selecting an option from the menu console *(? marks).*
You should implement the GUI according the following specification:
- The user can visualise the list of vehicles in WestminsterRentalVehicle system *(8 marks)*.
- The user can filter the vehicles according at least one parameter (e.g. the type of the vehicle, or the engine size, or the make, etc.) *(4 marks).*
- The user can check the availability of the vehicle in specific dates and can book make a reservation if it is available *(8 marks)*.

Note: You can choose how the GUI should look like and how to meet at the best these specifications.

3. Testing and system validation:

- Write a test plan designed to ensure that the coded solution works as expected. The test plan will include specific instructions about the data and conditions the program will be tested with *(5marks).*
- Implement an automated testing (you can use JUnit or feel free to use any other tool or scripts for unit testing) that runs scenarios of each of the use cases you implemented in the console menu *(10 marks)*.
- The following will be evaluated:
  - o The robustness of the code through the use of error handling and input validation *(5marks).*
  - o The quality of the code and the adherence to coding standards and conventions *(5 marks).*

Demonstration
You will be expected to be familiar with your code and required to make any ad-hoc changes during the demonstration
Please note that no marks will be given to a task if there is no understanding of the code.
Note that the maximum mark for work marked without a viva/demonstration shall range from 1-30% only.

Coursework Guidance

The implementation goals are split into 4 phases:

Phase I This involves an initial design of your system through UML diagrams and the implementation of the classes Vehicle, Car, Motorbike and Schedule (plus any accessory class you might need), and the Interface RentalVehicleManager. The marking scheme allows for students to score up to 35%. I would suggest concluding this Phase 1 by Week 05.

Phase II This is for implementing the system functionalities through a console menu. You should provide the implementation (in the class WestminsterRentalVehicleManager) of methods to add, delete, print, sort vehicles and generate report. The marking scheme allows for students to score up to 55%. I would suggest concluding this phase 2 by Week 07.

Phase III This phase involves the GUI implementation and the possibility for a user to visualise and book a vehicle. The marking scheme allows for students to score up to 75%. I would suggest concluding this phase 3 by Week 09.

Phase IV This phase is dedicated to the testing and evaluation of your system and the marking scheme allows for students to score up 100% for completing all the phases. This Phase 4 needs to be concluded by week 11 (week of final submission).