### lmu_blakCourse Submission Cover Sheet Module: CS4001 Programming Engineering

### Assignment no: 002

### Weighting: 30% of module mark

### Deadline: 28th of May 2021

**Module Leader: Sandra Fernando Student ID:**

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

(Signature:) Date:

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations on*

Cheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples.

1. **Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.**
2. **Falsifying data in experimental results.**
3. Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
4. **Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.**
5. Collusion to present joint work as the work solely of one individual.
6. Plagiarism, where the work or ideas of another are presented as the candidate's own.
7. Other conduct calculated to secure an advantage on assessment.
8. Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.

Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Stating an author’s argument and attributing it, perhaps by quoting, achieve quoting ideas

1

# Assignment

This assignment carries 30% of the overall module weighting and the grading scheme is available at the end of this document.

**Your .java files and report for this part must be uploaded to WebLearn and submitted by 3pm on Wednesday 28th of May 2021.** The assignment must be carried out individually so you must not obtain help from anyone other than the module teaching staff. You must not copy code from any source apart from the module core text and the module materials in WebLearn. Collusion, plagiarism (unreferenced copying) and other forms of cheating constitute Academic Misconduct, which can lead to failure of the module and suspension from the University.

# Aim

The aim of this assignment is to add a class to the project that you developed for the first part of the coursework to make a graphical user interface (GUI) for a system that stores details of cars in a car company. The class will contain a main method and will be tested using the command prompt. A learning aid about the command prompt is linked from the main index page of the learning materials. You will also need to write a report about your program. You are advised to study the learning aid on reflective practice, which is also linked from the module in WebLearn. Your report should be no more than 1200 words in length (excluding the cover page, table of contents, class diagram, pseudocode and code listing).

# Deliverables

Create a new class within the project called CarCompany. When you are ready to submit your solution, upload your CarCompany.java file, together with the Car.java, CarToBuy.java and CarToRent.java files from the first part of the coursework (not any other files from the project) to WebLearn, together with your report in Microsoft Word format and, when you are absolutely sure that you have uploaded the correct files, press the submit button.

# Program

An example GUI is shown below:



Your GUI should contain the same components but you are free to use a different layout if you feel that it improves the aesthetics, ease of use etc. The CarCompany class should store an array list (***not*** an array) of type Car to hold the cars tobuy and the cars to rent.

# Text field input methods (11 marks)

For each text field, write a method to read its contents and return the value. For the description, customer name, rental date and return date, the value should simply be returned as a string but, for the other text fields, the input value should be converted into its correct integer data type and then that numerical value should be returned.

There should be text fields for entering:

1. the price
2. the year of registration
3. the mileage
4. the description
5. the admin fee
6. the daily rate
7. the customer name
8. the rental date
9. the return date
10. the number of days
11. the car number (its position in an array list of cars)

The input method for the car number is different from the input methods for the other text fields. The method first initialises the car number to -1. If the input from the user via the GUI is valid, the car number will be changed from -1 to a value that corresponds to a car in the array list. The input from the car number text field must therefore be tested using a try/catch statement to ensure that the car number entered is an integer and, if it is, that it is in the correct range. If the value

entered is an integer but is not in the correct range, a suitable error message is displayed using a message dialog box. If the car number entered is not an integer then an alternative error message is displayed using a message dialog box. The method will therefore either return –1 or a value that corresponds to a car in the array list, depending on whether or not there was an error in the input. Any method that gets the car number should check its value and only use it if its value is not equal to –1. **(3 marks)**

# Buttons (14 marks)

The GUI should have the following buttons with its corresponding action performance:

# Add Car To Buy

The description, price, year and mileage are input via the GUI. When this button is pressed, the methods to get the description, price, year and mileage are called, and these values are used to create a new object of type CarToBuy, which isadded to the array list of cars.

# Add Car To Rent

The description, admin fee and daily rate are input via the GUI. When this button is pressed, the methods to get the description, admin fee and daily rate are called, and these values are used to create a new object of type CarToRent, whichis added to the array list of cars.

# Buy Car

The car number and the customer name are input via the GUI. When this button is pressed, the method to get the car number is called and its value is checked. If it is not equal to –1, the method to get the customer name is called, the method to get the car from the array list is called and the car is cast to CarToBuy, then the method from the CarToBuy class to buy the car is called.

# Rent Car

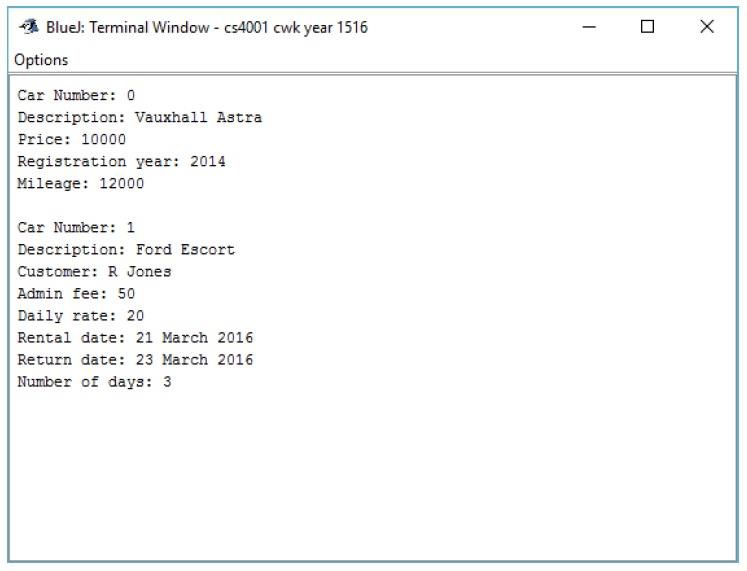
The car number, customer name, rental date, return date and number of days are input via the GUI. When this button is pressed, the method to get the car number is called and its value is checked. If it is not equal to –1, the methods to get the customer name, rental date, return date and number of days are called, the method to get the car from the array list is called and the car is cast to CarToRent, then the method from the CarToRent class to rent the car is called.

# Return Car

The car number is input via the GUI. When this button is pressed, the input value of the car number is checked. If a suitable value has been entered, the method to get the car number is called and its value is checked. If it is not equal to –1, the method to get the car from the array list is called and the car is cast to CarToRent, then the method from the CarToRent class to return the car is called.

# Display All

When this button is pressed, the car number and the information relating to that car is displayed for each car, for example:



# Clear

When this button is pressed, the text is cleared from all of the eleven text fields.

# Report

For the report, marks will be awarded as follows:

A class diagram of all three class of CarCompany showing the name of the class, the data types and names of the fields, and the return types and signatures of all of the methods. **(5 marks)**

A short description of each method in the CarCompany class. **(9 marks)**

Pseudocode for the following button-handling methods in the CarCompany class: **(18 marks)**

1. Add Car To Buy
2. Add Car To Rent
3. Buy Car
4. Rent Car
5. Return Car
6. Display All

You should give evidence through appropriate screenshots of the following testing that you carried out on your program:

1. Test that the program can be compiled and run using the command prompt, including a screenshot similar to Figure 1 from the command prompt learning aid. **(5 marks)**
2. Test the following: adding a car to buy, adding a car to rent, selling a car, renting a car, returning a car, displaying all of the cars **(7 marks)**
3. Test that appropriate dialog boxes appear when unsuitable values are entered for the car number, (include a screenshot of the dialog box, together with a corresponding screenshot of the GUI, showing the values that were entered). **(7 marks)**

The report should contain a section on error detection and error correction where you give examples and evidence of three errors encountered in your implementation. The errors (syntax and/or runtime) should be distinctive and not of the same type. **(5 marks)**

The report should contain a conclusion, where you evaluate your work, reflecting on what you learnt from the assignment. You are advised to study the learning aid on reflective practice, which is linked from the module in WebLearn**. ( 5 marks)**

The report should include a title page (including your name and ID number), a table of contents (with page numbers), and a listing of the code (in an appendix). Marks will also be awarded for the quality of writing and the presentation of the report**. (7 marks)**

Additional features and advanced coding **(4 marks)**