ONLINE CAR RENTAL SYSTEM (OCRS)

SUPER CAR RENTAL SERVICES (SCRS) is one of the fast-growing Online Car Rental Service in Malaysia which help customer to save their time by booking rental cars online from their place. SCRS decided to enhance their online Car Rental booking services by allowing customers to book cars for rent, online.

SCRS requires you to develop Python program for the OCRS which have 3 type of users and should contains features stated below:

Functionalities of Admin

i. Login to Access System.
ii. Add Cars to be rented out.
iii. Modify Car Details
iv. Display All records of
    a. Cars Rented Out
    b. Cars available for Rent
    c. Customer Bookings
    d. Customer Payment for a specific time duration
v. Search Specific record of
    a. Customer Booking
    b. Customer Payment
vi. Return a Rented Car.
vii. Exit

Functionalities of All Customers (Registered / Not-Registered)

i. View all cars available for rent.
ii. New customer Register to Access other Details
iii. Exit

Functionalities of Registered Customer

i. Login to Access System
ii. Modify Personal Details.
iii. View Personal Rental History.
iv. View Detail of Cars to be Rented Out.
v. Select and Book a car for a specific duration.
vi. Do payment to confirm Booking.
vii. Exit

1.0 REQUIREMENTS

i. You are required to carry out extra research for your system and document any logical assumptions you made after the research.

ii. Your program should use symbolic constants where appropriate. Validations need to be included to ensure the accuracy of the system. State any assumptions that you make under each function.

iii. You are required to store all data in text files. There is no limit on the number of text files that can be used but they should be kept minimum.

iv. You are expected to use list and functions in your program. Your program must embrace modular programming technique and should be menu-driven.

v. You may include any extra features which you may feel relevant and that add value to the system.

vi. There should be no need for graphics in your program, as what is being assessed, is your programming skill not the interface design. The marking scheme for the assignment has been provided so that you clearly know how the assessment for this assignment would be done.

vii. You should include the good programming practice such as comments, variable naming conventions and indentation**.**

viii. In a situation where a student:

- ***Failed to attempt the assignment demonstration, overall marks awarded for the assignment will be adjusted to 50% of the overall existing marks.***
- ***Found to be involved plagiarism, the offence and will be dealt in accordance to APU regulations on plagiarism.***

ix. You are required to use Python programming language to implement the solution. Use of any other language like C/C++/Java is not allowed.

x. Global variables, build in functions like min, max, sort, and search are not allowed.

2.0 DELIVERABLES

You are required to submit a softcopy of:

i. Program coded in Python – submitted as .py file.
   - Name the file under your name and TP number (e.g. KATHY_SIERRA_TP123456.py)
   - Start the first two lines in your program by typing your name and TP number (e.g. as follows):
     #KATHY SIERRA
     #TP123456

ii. Text files created through test data – submitted as .txt files.

iii. A documentation of the system – submitted as NAME_TPNUMBER.pdf file - that incorporates basic documentation standards such as header and footer, page numbering and includes:
   - Cover page
   - Table of contents
   - Introduction and assumptions
   - Design of the program – using pseudocode **and** flowcharts – which adheres to the requirements provided above
   - Program source code with explanation
   - Additional features source code with explanation
   - Screenshots of sample input/output with explanation
   - Conclusion
   - References using Harvard Name Referencing

3.0 ASSESSMENT CRITERIA

i. Design (Pseudocode and Flowchart) 20%

Detailed, logical and accurate design of programmable solution.

ii. Coding / Implementation (Python code) 20%

Application of Python programming techniques (from basic to advance); good programming practices in implementing the solution as per design; and adequate validation meeting all system requirements with all possible additional features.

iii. Documentation 20%

Adherence to document standard format and structure; screen captures of input/output with explanation; and inclusion of generated text files.

iv. Group Work 10%

Contribution toward group work (design, coding and documentation).

v. Q & A 10%

Ability to trace code and answer questions.

vi. Demonstration 20%

Ability to run and explain work done.

	Design (%)	**Coding (%)**	**Documentation (%)**	**Group Work (%)**	**Q&A (%)**	**Demo (%)**
Group	10	20	20			
Individual	10			10	10	20

4.0 PERFORMANCE CRITERIA

Distinction (80% and above)

This grade will be assigned to work which meets all of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to advanced level. The program solution is unique with excellent coding styles and validation. The program implemented maps completely against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with hardly any errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented have clear explanation. Student must be able to provide excellent explanation of the codes and work done, show additional concepts / new ideas used in the solution, able to answer all questions posed with accurate / logical answers / explanation provided with sound arguments and clear discussion. Overall an excellent piece of work submitted.

Credit (65%-74%)

This grade will be assigned to work which is considered to be of good standard and meets most of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to at least intermediate level. The program solution is unique with good coding styles and validation. The program implemented maps well against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with minor errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented with some explanation. Student must be able to provide good explanation of the codes and work done, answer most questions posed with mostly accurate / logical answers / explanation. Overall a good assignment submitted.

Pass (50%-64%)

This grade will be assigned to work which meets at least half of the basic requirements (approximately 50%) stated in the questions. The program runs smoothly when executed. There is clear evidence and application of Python concepts at basic level. The program solution is common with basic coding styles and validation. The program implemented somewhat maps with the design (pseudocode and flowchart) as seen in the documentation. The design of the solution is average in terms of logic and style with some errors / omissions. The documentation

has some missing components. Sample inputs/outputs documented but without any explanation. Student must be able to explain some codes and work done and able to answer some questions posed with some accurate / logical answers / explanation. Overall an average piece of work submitted.

Fail (Below 50%)

This grade will be assigned to work which achieved less than half of the requirements stated in the question. The program is able to compile but not able to execute or with major errors. The program solution has only basic coding styles with no validation. The program solution has little or no mapping with the design. The design of the solution has major / obvious errors / omissions. The documentation has some missing essential components. Student is barely able to explain the codes / work done and answer given on the questions posed but with mostly inaccurate / illogical answers / explanation. Overall a poor piece of work submitted.