

Course Learning Outcomes

Upon completion of this assignment, you should be able to:

CLO2	Construct a programmable solution using appropriate problem-solving methods and programming concepts to the given scenario. (C3, PLO2)
------	--

1.0 GROUP ASSIGNMENT DESCRIPTION

Title: Online Hall Event Renting System

Section A: System Specification

The CEO of Hall Symphony Inc. has chosen your team to create a system for renting their halls, making it easy for customers to book them at their preferred times. The business operates daily from 8:00 AM to 6:00 PM. Hall Symphony offers three types of halls: auditorium (seating 1000 people), banquet hall (seating 300 people), and meeting rooms (seating 30 people). Customers can reserve these halls for their events. Table 1 shows the pay rate for each of the hall type

Hall Type	Rent Rate per hour (in RM)
Auditorium	RM 300.00
Banquet Hall	RM 100.00
Meeting Room	RM 50.00

Functionality of The System

There are 2 end users for this system that includes:

- **ADMINISTRATOR**
 1. Requires login to access the system using username and password
 2. Performs hall management:
 - i. Enters the hall information:
 1. Hall ID
 2. Hall Name
 3. Hall Description
 4. Hall Pax

5. Hall availability status (based on date & time)
6. Rate Price per day
- ii. View all the hall information
- iii. Search the hall information
- iv. Edit the hall information
- v. Delete the hall information
3. Performs booking management:
 - i. View all booking information
 - ii. Search the booking information using the username or email
 - iii. Edit booking information
 - iv. Delete/Cancel the booking information
4. Performs user management:
 - i. View all the user information
 - ii. Search user information using the first or last name
 - iii. Edit the user information
 - iv. Delete/Block the user from login
5. Logout the system

- **USER**

1. Requires registration before proceeding to perform booking. The user needs to enter the following details for registration:
 - i. Username
 - ii. Password
 - iii. First Name
 - iv. Last Name
 - v. Date of birth
 - vi. Contact Number
 - vii. Email Address
2. Requires login using username and password to proceed booking
3. Perform Booking:
 - i. Event Name
 - ii. Event Description
 - iii. Hall ID (based on availability status)
 - iv. Number of Pax
 - v. Date and time of renting
 - vi. Payment Price

4. View their booking information after entering all booking information
5. Delete/Cancel their booking information
6. Edit their booking information
7. Search the booking information
8. Update their profile information
9. Logout the system

The application must be developed using **python** programming language. Data must be stored in text files. The program submitted should be free of syntax and runtime errors. Besides that, validation should be done for each entry from the users to avoid **logical errors**.

IMPORTANT INSTRUCTIONS:

This is a group assignment. Upon submission of your assignment, you are required to present your assignment at a date and time specified by your module lecturer.

Each team member is required to contribute towards some features in the system and documentation, present and explain his or her work accordingly. Each team member should be able to answer questions during presentation with regards to the overall systems project and or specific question(s) related to the codes used in the development. Responses to these questions such as “I don’t know because I didn't work on that part of the assignment” are not acceptable and will result in a penalty for either the entire group or specific individual(s)

2.0 REQUIREMENTS

- i. This is a group assignment, which comprises of **3 to 4** members.
- ii. You are required to carry out extra research for your system and document for any logical assumptions you made after the research.
- iii. 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.
- iv. 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 think of efficiency as well since too many text files also can affect the system performance.
- v. You are expected to use any type of data storing and functions in your program. Your program must embrace modular programming technique and should be menu driven.
- vi. You may include any extra features which you may feel relevant and that add value to the system.
- vii. 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.
- viii. You should include the good programming practice such as comments, variable naming conventions and indentation.
- ix. 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.*
- x. You are required to use **Python** programming language to implement the solution. Use of any other language like C/C++/Java is not allowed.
- xi. **Global variables, build in functions** like min, max, sort, etc... are **not allowed**.

3.0 DELIVERABLES

You are required to submit a softcopy of:

- i. A softcopy of the report (*in Word Document format*), in addition to the **python files** of the programs. The organization of files and folders **must adhere to the following instructions precisely**:
 - The report should be named using format
 - “<GroupNo>_<student ID-leader>_<student ID-member1>_<student ID-member2>_<student ID-member3>_<student ID-member3>.docx”.
 - For example “G1_TP012345_TP012344_TP012123.docx”
 - All the source codes (**.py and .txt**) should be zipped into one file and named following the above format. Make sure to **DELETE** all non-source-code files, including executables (*.exe).
- ii. A documentation of the system - 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 **or** flowcharts – which adheres to the requirements provided above
 - Program source code and explanation
 - Screenshots of sample input/output and explanation
 - Conclusion
 - References using APA Name Referencing
- iii. It is compulsory to submit the documentation and the application via Moodle.
- iv. Submit a Workload Matrix given by lecturer through Moodle.
- v. Each group member is required to participate in all tasks / discussions together.
- vi. Presentation schedules will be published at a later date through Ms.TEAMS. Starting Week 12.
- vii. Last submission due date: **Week 12 on 22nd December 2023 (Friday), before 11:59 p.m.**

- a. Late submission will result in marks deduction for the individual project marks, without warning.

4.0 ASSESSMENT CRITERIA

- i. Design (Pseudocode or Flowchart) 30%
Detailed, logical and accurate design of programmable solution.
- ii. Coding / Implementation (Python code) 40%
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. Demonstration 10%
Ability to run, trace code, explain work done and answer questions.

5.0 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.