

Course Learning Outcomes:

Upon completion of this assignment you should be able to:

CLO1	Translate simple problem statements into programmable solutions using flow chart/pseudo code (C3, PLO2).
CLO2	Comprehend knowledge of basic and advanced programming concepts (C2, PLO1).
CLO3	Show the ability to write computer programs for a given problem statement (P4, PLO3).

1.0 INDIVIDUAL ASSIGNMENT DESCRIPTION**UNIVERSITY STUDENT APARTMENT MANAGEMENT SYSTEM**

The University Student Services provides apartment for rental to the university female students at affordable rates. The types of apartment and its rental details are given in the Table 1 below:

Table 1: Apartment Type and Rental Details

Type	No. of Rooms per Apartment	Room Type	Monthly Rental per Room (RM)	Facilities		
				Kitchen	Laundry	Internet (Optional) Monthly Subscription Fee per User (RM)
A	2	2 Single Rooms	400.00	√	√	50.00
B	3	1 Master Bedroom	500.00		√	40.00
		2 Single Rooms	300.00	√		

There are 20 apartments available for rental under each type. Each apartment can house exactly one student per room. Students are required to check out from the apartment at the end of every semester. One semester is equivalent to 5 months.

You are required to write a Python program with following features to manage the apartment rental:

1. Student Registration and Apartment Room Allocation

Before an apartment room is allocated, the program should register the student and allow the selection of apartment type (i.e. Type A or B) and Internet subscription. The program should then check for the number of occupants in selected apartment and recommend a room in the next available apartment unit. Upon room allocation, the program should calculate rental charges for one semester based on the apartment selection and Internet subscription. In addition to this amount, students will be required to pay a deposit of RM100 for utility and room key. After receiving the total bill (rental charges added with deposit), a student should be given an option to pay full or partial amount. Students who opt to pay partial amount will be required to pay at least 50% of the rental charges and the RM100 deposit as their first installment.

Important Note:

- i. When testing the program, you should house at least 10 students in apartment Type A and 15 students in apartment Type B. Apartment Type B should house at least 5 students in Master Bedroom.
- ii. Student registration, their room allocation and payment details should be recorded in one or more text files.
- iii. The test data should be having atleast 10 students opting for partial payment. Students who opt for partial payment should be informed on the minimum payment amount for their first installment which is 50% of the rental charges added with deposit. They should be allowed to pay any amount that are more than or equals to this minimum amount. Your test data is expected to have different amounts (a combination of some students paying exact minimum amount while others paying more than the minimum amount) paid as first installments. The program also should be tested/validated with an input amount lower than the minimum payable amount.
- iv. Some students should be subscribed with Internet service during room allocation to diversify the test data.

2. Early (or mid-way) Check-Out

The program should have an option for students to check-out before the end of semester. In an event where a student checked-out early, the billing should be re-calculated based on the number of months stayed. The program should then indicate whether the student has a balance amount to settle or refund to receive.

Important Note:

- i. Calculation of balance to settle or refund to receive should include the deposit paid during check-in.
- ii. When testing the program, you should have at least 3 students who had made early check-out.
- iii. Records of all checked-out students should be removed from its original text file and stored in a separate text file. Only one text file is allowed to store all checked-out students' records.
- iv. The rooms vacated by these students should be made available first for next allocation. You are required to provide test data on the room re-allocation in a checked-out room.

3. Accounts Information

The program should have options to print:

- i. Total deposit collected.
- ii. Total amount collected excluding the deposit. This amount should include amount received from the checked-out students.
- iii. Total amount receivable from the students.

4. Searching Functionalities

The program should have options to search for a particular student's record when searched by:

- i. TP number or name.
- ii. Apartment type, apartment number and room number.

2.0 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. You may include some analytics in your program as additional features like percentage of occupancy by apartment type; percentage of occupants who stay with/without Internet subscription; tabulation of occupied and available apartment rooms; or any other feature(s).
- 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. Global variable is not allowed.
- x. Results of a comprehensive testing is to be included in your document. The tests conducted shall take into consideration of all valid inputs and negative test cases.

3.0 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_S[REDACTED].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 and explanation
 - Screenshots of sample input/output and explanation
 - Conclusion
 - References (if any) using Harvard Name Referencing

4.0 ASSESSMENT CRITERIA

- | | | |
|------|---|-----|
| i. | <u>Design (Pseudocode and Flowchart)</u> | 30% |
| | Detailed, logical and accurate design of programmable solution. | |
| ii. | <u>Coding / Implementation (Python code)</u> | 30% |
| | 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. | <u>Documentation</u> | 25% |
| | Adherence to document standard format and structure; screen captures of input/output with explanation; and inclusion of generated text files. | |
| iv. | <u>Demonstration</u> | 15% |
| | 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.