Course Learning Outcomes:

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

CLO1	Explain the fundamental of software development and programming concepts (C2, PLO1) – Final Exam
CLO2	Construct a programmable solution using appropriate problem solving methods and programming concepts to given scenario. (C3, PLO2) – Group Assignment

1.0 GROUP ASSIGNMENT DESCRIPTION

BANKING SERVICE SYSTEM

You as a team (group) of 2 members, are required to develop a Banking Service Application to handle all the primary information required to maintain the customer account in a bank. Once the system is opened, all users (including admin staff and customers) have to pass through a login page to enter the main screen. Upon login, the system should display the user's name on the system's user interface. Customers are allowed to make transaction for deposit and withdrawal of the money. In Addition, customers could also reset their own password and print a Statement of Account for a specified duration.

You are required to work in a group of two members. The following characteristics are important to be included in your system:

Group Member 1 Tasks:

- A default Super User Account should be created in the system that could be used to create administrative staff accounts. This Account should be used to create other users account (i.e., login credentials).
- In order to register with the bank, the customer is required to fill a registration form to provide his/her personal details. The staff then enters the customer details into the system, and thereafter provides him with the unique account number and a default password. A unique customer account number must be autogenerated and should in a sequence. The account can be **Savings** account or **Current** account.
- Admin staff may update or edit the staff or customer's details. However, customer's ID and name should not be available for update. Customer may only perform Deposits or Withdrawals.
- Customer's Statement of Account Report may be generated by both Admin Staff and Customer's Statement of Account Report must be printed for a specified period (i.e. a start and an end date must be entered). A total of all deposits and withdrawals should be displayed for the specified duration at the end of the report.

Group Member 2 Tasks:

- The customer can login into the system using account number and the default password. In addition to modifying the password, the customer can perform deposit and withdrawal transactions using this account number and password.

- After login into the system, the customer can have the following options:
    - deposit any amount of money in his/her account and the deposited amount should be added into the existing balance. The new balance should be updated in the data file.
    - Withdraw any amount. It should always checked whether the withdrawal of money will affect the minimum balance or not. If yes, the transaction will not be permitted. Minimum balance for saving account and current is RM100 and RM500 respectively.
- Unique User Interface (UI) using menus should be done for all interactions between users and the system.
- Customer's Statement of Account Report may be generated by the customer, and Customer's Statement of Account Report must be printed for a specified period (i.e. a start and an end date must be entered). A total of all deposits and withdrawals should be displayed for the specified duration at the end of the report.

All details must be saved in files – text files.
You are allowed to import only two modules as mentioned below:

1. os
2. datetime

The program submitted should compile and be executed without errors. Besides, validation should be done for each entry from the users in order to avoid logical errors.

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. 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.

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 (user interface) in your program, as what is being assessed, is your programming skill not the interface design.

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 in plagiarism, the offence 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_SIERRA_TP123456_DAVID_LEE_TP04321.py)
   - Start the first two lines in your program by typing your name and TP number (e.g. as follows):
     #KATHY SIERRA , DAVID LEE
     #TP123456, TP04321

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

iii. A documentation of the system (1000 words) – submitted as NAME1_TPNUMBER1_NAME2_TPNUMBER2.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
- Explanations of Programing Concepts with source code for explanation
- Screenshots of sample input/output and explanation
- Conclusion
- References (if any) using APA Referencing

4.0 ASSESSMENT CRITERIA

i. Design (Pseudocode and Flowchart) 30%

Detailed, logical and accurate design of programmable solution.

ii. Coding / Implementation (Python code) 30%

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

iii. Documentation 25%

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

iv. Demonstration 15%

Ability to run, trace code, explain work done and answer questions.

The marking scheme for the assignment has been provided so that you clearly know how the assessment for this assignment would be done.

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.