

UNIVERSITI TUNKU ABDUL RAHMAN

JANUARY 2020 TRIMESTER

INDIVIDUAL FINAL ASSESSMENT

**UECS1643/UECS1653/UCCS1633/UGCS1633 FUNDAMENTALS OF  
PROGRAMMING**  
**UECS1003 PROGRAMMING CONCEPTS AND DESIGN**

WEDNESDAY, 6<sup>th</sup> MAY 2020

DURATION: 2 DAYS

BACHELOR OF ENGINEERING (HONS) BIOMEDICAL ENGINEERING  
BACHELOR OF ENGINEERING (HONS) CHEMICAL ENGINEERING  
BACHELOR OF ENGINEERING (HONS) CIVIL ENGINEERING  
BACHELOR OF ENGINEERING (HONS) ELECTRICAL AND ELECTRONIC  
ENGINEERING  
BACHELOR OF ENGINEERING (HONS) ELECTRONIC ENGINEERING  
BACHELOR OF ENGINEERING (HONS) ELECTRONIC AND COMMUNICATIONS  
ENGINEERING  
BACHELOR OF ENGINEERING (HONS) ELECTRONICS (COMPUTER  
NETWORKING)  
BACHELOR OF ENGINEERING (HONS) ENVIRONMENTAL ENGINEERING  
BACHELOR OF ENGINEERING (HONS) INDUSTRIAL ENGINEERING  
BACHELOR OF ENGINEERING (HONS) MECHANICAL ENGINEERING  
BACHELOR OF ENGINEERING (HONS) MATERIALS AND MANUFACTURING  
ENGINEERING  
BACHELOR OF ENGINEERING (HONS) MECHATRONICS ENGINEERING  
BACHELOR OF ENGINEERING (HONS) PETROCHEMICAL ENGINEERING  
BACHELOR OF SCIENCE (HONS) ACTUARIAL SCIENCE  
BACHELOR OF SCIENCE (HONS) FINANCIAL MATHEMATICS  
BACHELOR OF SCIENCE (HONS) GAME DEVELOPMENT  
BACHELOR OF SCIENCE (HONS) PHYSICS  
BACHELOR OF SCIENCE (HONS) STATISTICAL COMPUTING AND OPERATIONS  
RESEARCH

**DEADLINE: 09:00AM, 8<sup>th</sup> MAY 2020**

**Instruction to Students:**

1. This assessment consists of FOUR (4) parts, and each part carries 25 marks.
2. In order to gain full or higher marks, you are required to fulfill the requirements described in **ALL FOUR (4) parts**. Submit the **ANSWER SCRIPT** together with a **C++ SOURCE** and an **INPUT TEXT FILE** by **09:00AM, 8<sup>th</sup> May 2020**.

**UCCS1633/UGCS1633 FUNDAMENTALS OF PROGRAMMING**

3. You may submit your answer via Web-Based Learning Environment (**WBLE**) There will be two (2) submission links (separate links for different programmes) created in WBLE for you to submit softcopy of your works. **Ensure your submission is captured by WBLE.** Alternatively, you may email your answer to [phanky@utar.edu.my](mailto:phanky@utar.edu.my) if encounter any problem of uploading. Late submission will not be entertained.
4. Please read the instructions carefully before you answer the questions.
5. The answer script **MUST** be either a Microsoft Word file or a PDF file, in A4 size format.
6. All texts **MUST** be typed using Times New Roman characters with font size not less than 12, except for the diagram drawings.
7. For the diagram drawings, you can draw them on a paper, and then take pictures and include the pictures in the Word document as part of your answers. You may also use drawing tools in Word Processing Software to draw the diagrams.
8. On the first page of your answer script, you need to clearly state:
  - (a). Your Name
  - (b). Your Student ID
  - (c). Your Examination Index Number
  - (d). Faculty/Centre
  - (e). You Degree Programme
  - (f). Submission Date
9. On the second page of your answer script, you need to sign a Declaration Form to indicate your authenticity of submitted work without plagiarism.
10. Please name your answer script using the following file name for submission:  
**[Your Course Code]\_Answer\_[Your Exam Index Number]**  
  
For example, if your Course Code is UGCS1633 and Exam Index Number is A01234CBABF, then your answer script should be named as  
**UGCS1633\_Answer\_A01234CBABF.doc**  
**OR**  
**UGCS1633\_Answer\_A01234CBABF.pdf**
11. Please name your C++ source using the following file name for submission:  
**[Your Course Code]\_Program\_[Your Exam Index Number]**  
  
For example, if your Course Code is UGCS1633 and Exam Index Number is A01234CBABF, then your C++ source should be named as  
**UGCS1633\_Program\_A01234CBABF.cpp**
12. Save your works (*C++ source file, input text file(s) and answer script*) into a folder and compress the folder into a **ZIP** file. Name your ZIP file with your Course Code and Examination Index Number.

**UCCS1633/UGCS1633 FUNDAMENTALS OF PROGRAMMING**

For example, if your course code is UGCS1633 and Exam Index Number is A01234CBABF, then your ZIP file should be named as **UGCS1633\_A01234CBABF.zip**

- 13. In the case of plagiarism being suspected, the evidences will be submitted to the University Examination Disciplinary Committee for further investigation and trial. If found guilty, serious disciplinary action will be taken against the students.**

**UCCS1633/UGCS1633 FUNDAMENTALS OF PROGRAMMING****Introduction – Simulation of an ATM**

An automatic teller machine (ATM) is used by the customers of a bank. Each customer has an *account number*, a *customer name* and a *personal identification number (PIN)*; *account number* and *PIN* are required to gain access to the account. In a real ATM, the account number would be recorded on the magnetic strip of the ATM card. In this simulation, the customer will need to type it in. With the ATM, customers can log in to their account and make transactions. This process is repeated until the customer chooses to exit.

The transactions can be made under the account as follows:

1. Check balance – This is to allow customer to check balance amount in the account.
2. Deposit amount – This is to allow customer deposit amount into his/her account. Ask user for *deposit amount*; new balance should be updated in the account.
3. Withdraw amount – This is to allow customer withdraw amount from his/her account. Ask user for *withdrawal amount*; new balance should be updated in the account. You are required to make sure the customer has sufficient balance for withdrawer.
4. Transfer amount (to other customer) – This is to allow customer transfer amount to other customers. Ask user for *account numbers* of transferee and *transfer amount*. Then search for the customer records using the input number; new balance should be updated in both accounts (transferor and transferee) if the transferee record is found. If the transferee record is not found, display error message.
5. Check history – This is to allow customer to check history of transactions. Display the list of transactions forward according to transaction date. Each display should contain an appropriate heading and column captions. Note that this is an *advance question*. You are required to modify the structure definitions from *array of structures* to *nested arrays of structures* for this task. Make sure the first four processes (check balance, deposit amount, withdraw amount and transfer amount) work well before attempting this question. Your final mark will be affected if you fail to implement the first four processes when attempting to implement this process (check history).

Write an application in C++ to process customer transactions. Because this is a simulation, the ATM does not actually communicate with a bank. It simply loads a set of account numbers, PINs and balance amounts from a text file. Gather data on at least 5 customer records and prepare them in the text file(s). As each customer record is read, insert it into a list by using an *array of structures*. After building the list (array of structures), display a menu allowing customer to make transactions. When user chooses to exit the application, the program will write the updated data in the list (array of structures) to the text file.

Prepare the necessary data file(s) for building the data structure(s) needed in your application. You may give additional assumptions for your application. To make your program more robust and avoid problems at run time, do as much status/error checking as you can in your program. You may also add more features and/or record more details of customers in your program for enhancement.

**UCCS1633/UGCS1633 FUNDAMENTALS OF PROGRAMMING****Assessment****Part 1 - Fundamental Program Structures**

- i. You are required to maintain a good program style when developing your C++ program. Program style refers to things that make your program easy to read and understand but do not affect the execution of the program. Follow the advice given for placing whitespaces to improve readability, choosing good user-defined identifiers and giving comments that are relevant and liberally, but not to excess. (10 marks)
- ii. Your program structure will be assessed. To make your program runs well, the overall form of your program, with particular emphasis on the individual components of the program and the interrelationships between these components should be well-structured. (10 marks)
- iii. Arithmetic expressions used in deposit, withdraw and transfer processes will be examined to ensure they produce accurate results. (5 marks)  
[25 marks]

**Part 2 – Program Design Elements**

- i. Construct a *structure chart* with *data flow* and *execution of conditional and loops* for your solution design. (12.5 marks)
- ii. Construct a *flowchart* for *Menu Module* to illustrate the execution flow of the process. (12.5 marks)  
[25 marks]

**Part 3 - Use of Selection and Loop Control Structures**

The following processes/modules will be assessed:

- i. Log in process (7 marks)
- ii. Menu module (7 marks)
- iii. Withdraw transaction (5 marks)
- iv. Search process in the transaction of transfer amount (6 marks)  
[25 marks]

**Part 4 - Advanced Program Features**

The following components/processes/modules will be assessed:

- i. Design of structure definition (5 marks)
- ii. Manipulation of array(s) (5 marks)
- iii. Manipulation of file input and output processes (5 marks)
- iv. Design of the functions (10 marks)  
[25 marks]

**UCCS1633/UGCS1633 FUNDAMENTALS OF PROGRAMMING****Submission**

This is an *individual* assessment. Your **SUBMISSION SHOULD INCLUDE** the following:

1. C++ program ( .cpp )
2. Sample of input data ( text file(s) – .txt )
3. An answer script in a portable document format ( .pdf ) that contains:
  - 3.1. Structure chart for your solution design.
  - 3.2. Flowchart for *Menu Module only*.
  - 3.3. Sample outputs ( Alt+PrntScrn ) of the transactions in your program:
    - 3.3.1. Customer log-in process (successful and unsuccessful log-in)
    - 3.3.2. Menu
    - 3.3.3. Check balance
    - 3.3.4. Deposit amount
    - 3.3.5. Withdraw amount (successful and unsuccessful withdrawers)
    - 3.3.6. Transfer amount (successful and unsuccessful transfers)
    - 3.3.7. Check history (if any)

This final assessment will contribute 40% or 50% of your final mark depends on your course structure. The answer script will be marked for *correctness, completeness, presentation style, and relevant use of diagrams*, etc. And the C++ program together with the input text file(s) will be marked for *correctness, completeness, program style, adequate testing and appropriate documentation/comments*. It's your responsibility to understand the requirements of the tasks and prepare well for your submission.

**Plagiarism**

It is important that your solution to the final assessment be **your own work**. It is perfectly acceptable to seek advice for the requirements of the tasks when completing the final assessment, but **this must not be taken to the point where what is submitted is in part someone else's work**. Any violation of these rules will result in disciplinary actions.