**(SECJ1013) PROGRAMMING TECHNIQUE 1**

**SEM 1, SESSION 2023/2024**

**ASSIGNMENT 2**

**INSTRUCTIONS TO THE STUDENTS**

* This exercise must be done **in a group of two**.
* Any form of plagiarism is **NOT ALLOWED**. Students who copied other students' assignments will get **ZERO** marks (both parties, students who copied, and students who shared their work).
* Please put your **name and matric number** and your member’s **name and matric number** **AS COMMENTS IN THE BEGINNING OF THE CODE**.

**SUBMISSION PROCEDURE**

* Please submit this exercise no later than **December 15, 2023, Friday (00:00 MYT)**.
* Only TWO SOURCE CODE files (**.CPP**) is required for the submission (**NOT ALLOWED .ZIP** **FILE**).
* Only **ONE** submission per pair (group).
* Submit the assignment via the UTM's e-learning system (<https://elearning.utm.my/23241/>).

# SET 1

Write a complete C++ program for a clear demonstration of the banking system's functionality, which involves several tasks, including displaying the initial account information, performing a deposit of RM 500.00, performing a withdrawal of RM 200.00, and then displaying the updated account information.

The program should perform the following tasks:

Task 1:

Write a function named **displayAccountInfo**,

* which displays the initial account information.
* It includes the account holder's name, account number, and the current balance.

Task 2:

Write a function named **deposit**,

* which adds the specified amount to the current balance and outputs a success message.
* It performs a deposit transaction with the parameter **500.00**.

Task 3:

Write a function named **withdraw**,

* which checks if the specified amount is less than or equal to the current balance.
* If the condition is met, the function subtracts the specified amount from the balance and outputs a success message to the console.
* Otherwise, it displays an "**Insufficient funds for withdrawal**" message.
* It performs a withdrawal transaction with the parameter **200.00**.

Task 4:

* After performing the deposit and withdrawal transactions, the program again calls the **displayAccountInfo** function to output the updated account information to the console.
* This includes the modified balance after the deposit and withdrawal transactions.

Task 5:

Write a **main** function to perform the following tasks:

* The function may need to call the functions that are defined in Task 1, Task 2, and Task 3.
* You need to use an appropriate LOOP to perform the process in this function. The loop will be terminated when the user press ‘N’. It prompts the user if they want to perform another transaction and continues based on the user's input.
* The program should produce the output as in the sample execution given below.
* You are NOT ALLOWED to use arrays except an array of characters.

**Note: The values in bold are input by the user.**

Task 6:

* List all function prototypes.

Task 7:

You must ensure your program fulfil the following criteria:

* The program is able to run.
* The program uses an appropriate structure for the program (e.g. all required header files are included, the program is properly written, proper indentation, etc.).

**SAMPLE PROGRAM EXECUTION**

|  |
| --- |
| <<<<< My Accounts Overview >>>>>  Account Holder Name: User 1  Account Number: 1013202341  Balance: RM 200  <<<<< Deposit Transaction >>>>>  Deposit of RM 500 successful.  <<<<< Withdrawal Transaction >>>>>  Withdrawal of RM 200 successful.  <<<<< My Accounts Overview >>>>>  Account Holder Name: User 1  Account Number: 1013202341  Balance: RM 500  Do you want to perform another transaction? (Y/N): y  <<<<< My Accounts Overview >>>>>  Account Holder Name: User 1  Account Number: 1013202341  Balance: RM 500  <<<<< Deposit Transaction >>>>>  Deposit of RM 500 successful.  <<<<< Withdrawal Transaction >>>>>  Withdrawal of RM 200 successful.  <<<<< My Accounts Overview >>>>>  Account Holder Name: User 1  Account Number: 1013202341  Balance: RM 800  Do you want to perform another transaction? (Y/N): n  --------------------------------  Process exited after 7.191 seconds with return value 0  Press any key to continue . . . |

# SET 2

Write a complete C++ program for an example of demonstration of the food ordering system's functionality

The program should perform the following tasks:

1. Begin by including the necessary header files for input/output operations in C++.
2. Create the main function to start the program.
3. Display a welcome message and the menu of food items to the user.
4. Prompt the user to input the number corresponding to the item they want to order.
5. Take the user's input for food selection and store it in a variable.
6. Use conditional statements to calculate the total bill based on the user's choice of food item.
7. Display the total bill to the user.
8. End the program.

**Note: The values in bold are input by the user.**

**SAMPLE PROGRAM EXECUTION**

1. Example of select Pizza

|  |
| --- |
| Welcome to the Food Ordering System  1. Pizza - $10  2. Burger - $5  3. Sandwich - $7  Enter the number of the item you want to order: **1**  Your total bill is: $10  --------------------------------  Process exited after 2.818 seconds with return value 0  Press any key to continue . . . |

1. Example of select Burger

|  |
| --- |
| Welcome to the Food Ordering System  1. Pizza - $10  2. Burger - $5  3. Sandwich - $7  Enter the number of the item you want to order: **2**  Your total bill is: $5  --------------------------------  Process exited after 1.562 seconds with return value 0  Press any key to continue . . . |

1. Example of select Sandwich

|  |
| --- |
| Welcome to the Food Ordering System  1. Pizza - $10  2. Burger - $5  3. Sandwich - $7  Enter the number of the item you want to order: **3**  Your total bill is: $7  --------------------------------  Process exited after 1.684 seconds with return value 0  Press any key to continue . . . |