CMP 142 - Object Oriented Programming Fall 2019 **LAB-05**

Issue Date: October 19, 2020

The objective of this lab is to:

- 1. Practice array of objects, const data members and functions.
- 2. Practice good coding conventions e.g commenting, meaningful variable and functions names, properly indented and modular code.

Instructions!

- 1. This is a graded lab, you are strictly NOT allowed to discuss your solutions with your fellow colleagues, even not allowed asking how is he/she is doing, it may result in negative marking. You can **ONLY** discuss with your TAs or with me.
- 3. Strictly follow good coding conventions (commenting, meaningful variable and functions names, properly indented and modular code.
- 4. Save your work frequently. Make a habit of pressing CTRL+S after every line of code you write.

Task 01: [20 Marks]

You have created account class in last lab. Now again you have been assigned to develop a program that can manage a basic account with a little different functionality. This time you should also keep track of transactions. Your class should perform following tasks:

- Save the account balance.
- Save the number of transactions performed on the account.
- Allow deposits to be made to the account.
- Allow withdrawals from the account.
- Calculate and add interest for the period.
- Report the current account balance at any time.
- Report the current number of transactions at any time.

To implement above functionality you may need following private data members:

- double balance; //to store the current account balance.
- const double interestRate; //to store the interest rate for the period.
- double interest; //to store the interest earned for the current period.
 int transactions; //to store the current number of transactions.

Transactions variable should be incremented on every deposit and withdrawal. Show appropriate message in case of overdrawn but remember this message should not be displayed from withdraw function. Interest can be calculated by following formulae:

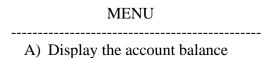
 $interest = balance \times interestRate$

To add interest of the period:

balance = balance + interest

Also implement constructors, destructor and getter setters for private data members. Make *const* functions where appropriate. Don't write setter for *ineterstRate*, assume it to be 0.045 by default. To test your class functionality, write a driver program. Create a function displayMenu() to display a menu on console. Keep running program until user enters G.

Sample Execution:



CMP 142 - Object Oriented Programming **Fall 2019 LAB-05**

Issue Date: October 19, 2020

- B) Display the number of transactions
- C) Display interest earned for this period
- D) Make a deposit
- E) Make a withdrawal
- F) Add interest for this period
- G) Exit the program

Enter your choice: **D**

Enter the amount of the deposit: 500

MENU

- A) Display the account balance
- B) Display the number of transactions
- C) Display interest earned for this period
- D) Make a deposit
- E) Make a withdrawal
- F) Add interest for this period
- G) Exit the program

Enter your choice: A

The current balance is PKR. 500.00

MENU

- A) Display the account balance
- B) Display the number of transactions
- C) Display interest earned for this period
- D) Make a deposit
- E) Make a withdrawal
- F) Add interest for this period
- G) Exit the program

Enter your choice: E

Enter the amount of the withdrawal: 700 ERROR! Withdrawal amount too large.

MENU

A) Display the account balance

- B) Display the number of transactions
- C) Display interest earned for this period
- D) Make a deposit
- E) Make a withdrawal
- F) Add interest for this period
- G) Exit the program

Enter your choice: E

Enter the amount of the withdrawal: 200

MENU

Issue Date: October 19, 2020

- A) Display the account balance
- B) Display the number of transactions
- C) Display interest earned for this period
- D) Make a deposit
- E) Make a withdrawal
- F) Add interest for this period
- G) Exit the program

Enter your choice: **F** Interest Added!

MENU

- A) Display the account balance
- B) Display the number of transactions
- C) Display interest earned for this period
- D) Make a deposit
- E) Make a withdrawal
- F) Add interest for this period
- G) Exit the program

Enter your choice: **B** Number of transactions: 2

MENU

- A) Display the account balance
- B) Display the number of transactions
- C) Display interest earned for this period
- D) Make a deposit
- E) Make a withdrawal
- F) Add interest for this period
- G) Exit the program

Enter your choice: G

Task 02: [10 Marks]

Create a class *Customer* to store name, NIC number and phone number of a customer. Implement following class specification:

CMP 142 - Object Oriented Programming Fall 2019 LAB-05

Issue Date: October 19, 2020

```
char* getName() const;
char* getCNIC() const;
long getPhone()const;
void displayCustomerInfo();
};
```

In *main()* function create an array of Customers to store data of 5 customers. Populate the array and display the information of customer in following table format:

| Customer Name | NIC Number | Contact Info |
|---------------|-----------------|--------------|
| Ali Ahmed | 35205-7563252-0 | 03254121515 |
| Hunnain Raza | 35201-8541236-1 | 03256896541 |
| Aisha Khalid | 35202-6354125-0 | 03334264549 |
| Amna Faheem | 35201-4545485-2 | 03121551515 |
| Rida Fatime | 32520-1514841-1 | 03004854845 |

There is no ELEVATOR to success, You have to take the STAIRS!

-ZIG ZIGLAR