

## CS 2024 – ASSIGNMENT #4

**Assigned:** 9/19/2019

**Due:** 9/25/2019

### PROBLEM:

This assignment is going to “build” on our Menu assignment from last time (A3). You will be able to re-use parts of your A3 Menu class and a little bit of your main.cpp. In short, you are going to create another Menu based system but this time your menu items will be commands to interact with a `BankAccount` class that you also create. Let’s start again with an example run of the program:

I’ll start by showing a sample run of what this should look like:

```
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 1
BALANCE: 0
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 2

Setting balance
ENTER NEW BALANCE> 10
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 1
BALANCE: 10
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 3

Deposit
HOW MUCH DO YOU WANT TO DEPOSIT> 5
```

```

---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 1
BALANCE: 15
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 4

Withdrawal
HOW MUCH DO YOU WANT TO WITHDRAWAL> 55
Withdrawal failed
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 1
BALANCE: 15
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 6
ERROR, unknown item selected
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 1
BALANCE: 15
---- MENU ----
(1) Balance Inquiry
(2) Set Balance
(3) Deposit
(4) Withdrawal
(5) Quit
Enter Choice> 5
Quitting!

```

How do we do this?

First, re-use your `Menu` class from the last assignment. You will need a new enumeration to define the following “constants” :

- `cUnknownItem`
- `cBalanceInquiry`
- `cSetBalance`
- `cDeposit`
- `cWithdrawal`
- `cQuitItem`

You can put these in an old style C/C++ enumeration or a new C++11 scoped enumeration.

The `Menu` class definition will be the same as it was for A3:

```
class Menu
{
public:
    void showMenu();
    MenuItemCode promptUser();
};
```

See the A3 description for information on what these methods do!

The “new” class needed for this assignment is a `BankAccount` class to manage a single instance of a bank account (used with this menu). Its class definition should look like this:

```
class BankAccount {
public:
    BankAccount();
    BankAccount(int initialBalance);

    int getBalance();
    void setBalance(int newBalance);
    bool withdrawal(int amount);
    void deposit(int amount);

private:
    int mBalance;
};
```

Obviously, this class treats the balance as an integer (dollars, no cents!) and only allows methods that adjust the balance in integer quantities. The `withdrawal()` method takes an amount to withdrawal as a parameter and only deducts it from the balance in the account IF there is enough “money” in the account to allow the withdrawal. It also makes sure that the withdrawal amount is a positive integer. If either of these cases is NOT true, it will return FALSE. Otherwise it returns TRUE.

Note also that I've asked you to provide two constructors for the `BankAccount` class. The first just sets the `mBalance` member variable to 0 while the second takes an initial value to store in the `mBalance` member variable. You actually will NOT be using the second constructor for this assignment, but we will use it in A5 next week.

Keep in mind that the implementation of the `BankAccount` class does not involve any user interaction. User interaction is handled directly in `main.cpp`.

For this assignment you will be adding two new files, `BankAccount.cpp` and `BankAccountH.h`.

Your `main.cpp` file will be a bit more complicated than it was in A3. You will need to implement the following functions:

1. A global function named `setBalance` that will prompt the user for a new balance and then use the `setBalance` member function of the `BankAccount` class to set the new balance.
2. A global function named `withdrawal` that will prompt the user for an amount to withdrawal and then call the `widthdrawal` member function of the `BankAccount` class to perform the withdrawal. This function should pay attention to what the `withdrawal` member function returns (true or false) and let the user know if the withdrawal was successful or not.
3. A global function named `deposit` that will prompt the user for an amount to deposit and then use the `deposit` member function of the `BankAccount` class to perform the deposit.

Your `main()` function will need to do the following:

1. Declare an instance of `Menu`
2. Declare an instance of `BankAccount`
3. Enter into a loop that does the following:
  - a. Shows the menu by calling the `showMenu` method in the `Menu` class
  - b. Prompts the user for a menu choice by calling the `promptUser` method in the `Menu` class
  - c. Enters into a switch statement that switches on the value returned by `promptUser` and use the global functions specified above to carry out the proper operations (NOTE: you don't need a global function for Balance Inquiry as it doesn't interact with the user... just call the `getBalance` member function of the `BankAccount` class directly)
  - d. The loop terminates when the user enters Quit

## **HINTS/DISCUSSION:**

Pay attention to the sample printout above. It will help remind you what you should be doing!

You will be declaring a single instance of `BankAccount` in your `main()` function. However you will need to be able to reference that instance of `BankAccount` in the global functions you will be implementing in `main.cpp`. How do you do that? Consider utilizing a pass-by-reference instance of `BankAccount` to the global functions. Review Lecture #6 for information about pass-by-reference.

Review the Assignment #3 description hints as many of them apply to this assignment as well.

Remember to not include a `.cpp` file in another `.cpp` file. While it is legal, it is not usually done.

There are parts of this assignment I have purposely not “spelled out”. Good luck and come to office hours with questions!

## **FILES TO SUBMIT:**

1. `main.cpp`
2. `MenuH.h`
3. `Menu.cpp`
4. `BankAccountH.h`
5. `BankAccount.cpp`
6. Writeup (.doc, .docx, .pdf, .txt accepted)