CST 8219 – W20 – Theory Assignment 2

Due Time: 23.59, Thursday April 9th, 2020 **Earnings:** 10% of your final grade

Submitted to Professor Surbhi Bahri

NOTE: Plan to finish a few days early to avoid last minute hardware/software holdups for which no allowance is given.

NOTE: The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.

Purpose: You are to write the code in Visual Studio 2019. This assignment will give you an opportunity to review material that has already been taught in an earlier C course and get up to speed for programming that is used in this course.

Implementation: Consider a bank that offers its customers the following account types:

- A 'Savings Account' that earns interest. The interest compounds monthly and is computed on the minimum monthly balance.
- A 'Chequing Account' that has no interest, gives you 4 free withdrawals per month, and charges a $2 transaction fee for each additional withdrawal.

The program should manage a set of accounts of both types and it should be structured in a way so that other account types can be added without affecting the main processing loop. Menu should be like:

D) eposit W) ithdraw M) onth Q) uit

For Deposit and Withdrawals, query the account number and amount. Print the balance of account after each transaction.
For the month ending command, accumulate interests or clear the transaction counter so that new transaction details can establish (depends on the type of account). Print the balance of all accounts.

Assignment Structure to be followed:

I. List all the classes that are part of hierarchy
II. Organize the classes into an Inheritance hierarchy
III. Determine the common responsibilities to carry out certain tasks at hand.
IV. Decide which functions are going to get overridden in the Derived Classes.
V. Define the public interface of each derived class
VI. Identify all the relevant data members and member functions along with the implementation of required constructors
VII. Process all the objects allocated on heap.

Explain the Inheritance relationship you'll be using with relevant UML Diagram.