**Programming Assignment 2**

**Purpose**: Demonstrate the use of classes

**Program definition:**

Design a **class bankAccount** that defines a bank account as an ADT and implements the basic properties of a bank account.

The program will be an interactive, menu-driven program.

a.Each object of the **class bankAccount** will hold the following information about an account:

account holder’s name

account number

balance

interest rate

**The data members MUST be private.**

Create an array of the bankAccount class that can hold up to 20 class objects.

b.Include the member functions to perform the various operations on objects of type **bankAccount**:

Include all appropriate constructors

Include any get and set member functions for member variables.

For the account number, include a separate, private, static member (integer) in the class to automatically assign and keep track of account numbers. Each customer should have a unique account number updated using the static member.

Add member functions to:

Set bank account data

Make a deposit into an account

Withdraw from an account. Be sure to include balance error checking, i.e., cannot withdraw more than is in the account.

Update balance with interest

Print **all** customer data

ALL member variables must be private and accessed through member functions. The main client program CANNOT access the variables directly.

c. Have the program provide the following capabilities (by user request):

Through a **menu** system provide the capability to:

(1) Add a customer. Ensure that the addition of a new customer **does not** cause an array overflow. The capability of entering the customer name, initial deposit amount and interest should be provided for a new customer. The customer’s account number must be generated from the private static data member and not entered during the ‘add a customer’ action.

(2) Print **all** customer data for **all** customers

(3) Update customer data as follows:

Display the customers already in the banking system.

**Request the user account number to update** (you may need to display customer names and numbers for the user to choose) and though a submenu:

(a) Make a deposit for the customer number entered

(b) Make a withdrawal (with error checking) for the customer number entered

(c) Print balance of the customer number entered

(d) Update balance with interest for the customer number entered

(e) Exit the submenu

(4) Exit the program

d.Write the definitions of the member functions of the **class bankAccount**.

c.Write an interactive program that uses the **class bankAccount** and tests various operations on the objects of the **class bankAccount**. Declare an array of 25 components of type **bankAccount**.

f. Include error checking (invalid menu choices, subtraction errors, etc.)

g. Include menus and submenus indicating all the menu options available for (1)‑(4) above.

All classes must have a class definition header file (\*.h) and a class implementation file (\*.cpp). You must have a main program that exercises these classes (\*.cpp). You must create a project in your IDE. **You must submit all header files, source files (\*.cpp) and the project file (\*.dev, \*.cpb).** Submit a **README** file that contains: The names of all your files including all the files in the project; how to compile; and how to run.

**All assignments are to be submitted to the appropriate Assignment posting in Canvas.**

**Each solution is to be uniquely your own; student collaboration not allowed.**

**Submit your code files (\*.cpp, \*.h, \*.dev/\*.cpb), and readme files**. Program **must** compile and all files must be submitted to receive credit. DO NOT submit the executable (\*.exe).

**A program that does not compile will receive a zero (0).**

You may zip all your files and submit the zip file. A zipped submission will receive 1 point extra credit.

See Canvas and Syllabus for due date

The code file **must** contain the following as documentation. If the following is not included, **10% may be deducted**:

The name of your C++ file

Your name

Some kind of date, either the due date or the date you finished

The type of input

The type of output

A brief description of the algorithm or purpose of the program

For example:

/ \* Program name: assignment1.cpp

Author: Pam Smith

Date: 8/25/18

Input: requests an input and output file name from the user (inputfile.txt provided).

Output is name of your choice

Output: displays output on the console and writes output to file name provided

Description: This program translates a word or phrase using the ROT13 cipher

\*/

**All** procedures and functions should be documented with the following information:

Precondition(s) as applicable

Postcondition(s) as applicable

For example:

/\*

Precondition: the integer value must contain a valid number representing the ordinal number of the letter contained in the character parameter; the character parameter must contain a value representing a letter of the alphabet

Postcondition: there are no post conditions

\*/

void myfunction (int a, char b) {}

Examples of documentation and readme files available in assignment documentation in Canvas.