**CIS247C Lab Week 1**

**CODE:**

/\*

Leah Rieger

CIS247C

ATM Application

10/30/2019

\*/

// BRING IN OUR LIBRARIES

#include <iostream>

#include <conio.h>

#include <string>

#include <fstream> // read/write to files

#include <ctime> // time(0)

#include <iomanip> // setprecision()

using namespace std;

//Prototypes

void deposit(double\* ptrBalance);

void withdrawal(double\* ptrBalance, float dailyLimit);

//overloaded method - ^ this version doesn't take withdrawal amount ^

void withdrawal(double\* ptrBalance, float dailyLimit, float amount);

//overloaded method that ^DOES take withdrawal amount^

// Entry point to the application

int main()

{

// create constant values -- cannot be changed

const int EXIT\_VALUE = 5;

const float DAILY\_LIMIT = 400.0f;

const string FILENAME = "Account.txt";

// create balance variable

double balance = 0.0;

// look for the starting balance; otherwise generate a random starting balance

ifstream iFile(FILENAME.c\_str());

if (iFile.is\_open())

{

// did the file open? if so, read the balance

iFile >> balance;

iFile.close();

}

else

{

// if the file did not open or does not exist,

// create a random number for the starting balance

srand(time(0));

const int MIN = 1000;

const int MAX = 10000;

balance = rand() % (MAX - MIN + 1) + MIN;

}

cout << fixed << setprecision(2) << "Starting Balance: $" << balance << endl;

//creating a pointer & set to balance variable location

double\* ptrBalance = &balance; //& means "address of"

//pause before clearing screen

cout << "\nPress any key to continue...";

\_getch();

// create loop variable BEFORE the loop

short choice = 0;

// start the app loop

do

{

// show the menu

system("cls"); // clears the console screen

cout << "Menu\n" << endl;

cout << "1) Deposit " << endl;

cout << "2) Withdrawal " << endl;

cout << "3) Check Balance " << endl;

cout << "4) Quick $40" << endl;

cout << "5) Exit" << endl;

// get user input

cout << "\nEnter your choice: ";

cin >> choice;

//run code based on the user's choice

switch (choice)

{

case 1:

deposit(ptrBalance); // passing a pointer so only 4 bytes have to go across sys bus.

break;

case 2:

withdrawal(ptrBalance, DAILY\_LIMIT); //passing 4 byte ptr!

break;

case 3: //show the balance

cout << fixed << setprecision(2) << "\nCurrent Balance: $" << balance << endl;

break;

case 4: //grab quick $40

withdrawal(ptrBalance, DAILY\_LIMIT, 40.0f);

break;

case 5:

cout << "\nGoodbye" << endl;

break;

default:

cout << "\nError. Please select from the Menu." << endl;

break;

}

//pause

cout << "\nPress any key to continue...";

\_getch();

} while (choice != EXIT\_VALUE);

//Now that app is over, need to write to file:

ofstream oFile(FILENAME.c\_str());

oFile << balance << endl;

oFile.close();

return 0;

}

//Make a deposit

void deposit(double\* ptrBalance)

{

//Get deposit & validate it

float deposit = 0.0f;

do

{

cout << "\nEnter deposit amount: ";

cin >> deposit;

if (cin.fail()) // did they dive us a char instead of an int

{

cin.clear(); //clears fail state

cin.ignore(INT16\_MAX, '\n'); //clears keyboard buffer

cout << "\nError. Please use numbers only.\n" << endl;

deposit = -1; //set deposit to a "bad" number

continue; //restarts loop

}

else if (deposit < 0.0f) //checks for neg num

cout << "\nError. Invalid deposit amount.\n" << endl;

} while (deposit < 0.0f);

//how do we get the double value located at the pointer?

//DEREFERENCE it using an asterisk!!

\*ptrBalance += deposit; //same as: \*ptrBalance = \*ptrBalance + deposit;

cout << fixed << setprecision(2) << "\nCurrent ptrBalance: $" << \*ptrBalance << endl;

//notice the asterisk!

}

//Make a withdrawal

void withdrawal(double\* ptrBalance, float dailyLimit)

{

//get the withdrawal(you should validate this input)

float amount = 0.0f;

cout << "\nEnter withdrawal amount: ";

cin >> amount;

//call the overloaded method version that takes

//balance, dailyLimit, amount

withdrawal(ptrBalance, dailyLimit, amount);

}

//Make a withdrawal - this overload accepts balance, dailyLimit & amount

void withdrawal(double\* ptrBalance, float dailyLimit, float amount)

{

//take away money from the account & show the balance

if (amount > dailyLimit)

{

cout << "\nError. Amount exceeds daily limit." << endl;

}

else if (amount > \* ptrBalance) //NOTICE the astrisk to dereference the pointer!!!!!!

{

cout << "\nError. Insufficient funds." << endl;

}

else

{

\*ptrBalance - +amount; //same as: \*ptrBalance = \*ptrBalance - amount

cout << "\nHere is your cash: $" << amount << endl;

}

cout << fixed << setprecision(2) << "\nCurrent Balance: $" << \*ptrBalance << endl;

}

**SCREENSHOT:**



