Lab 4 PDF

Name: Haichuan Wei

Lab: Lab 4

Date: 10/10/2021

Program Description:

This program uses two class structures to calculate the total balance of a bank account. The data is inputted by the user, and it is passed to the class to be calculated. The user will either use “deposit” to add money, “withdraw” to remove money, or “Transfer” to move money from one account to another. The user will input a double value for the amount they want to add or remove and a string value to point to which ever account they want to affect.

Program Source Code:

BankMain.cpp:

BankMain.cpp

```cpp
/*
Purpose- This program uses class structures to calculate the total balance of a bank account based on user defined inputs.
@author Haichuan Wei
*/
#include <iostream>
#include "printMeFirst.h"
#include "Bank.h"
#include "Account.h"
using namespace std;
int main()
{
    printMeFirst("Haichuan Wei", "CS-116 - 2021 Spring");
    Bank my_bank;
    cout << "\nInital bank balances: \n";
    my_bank.printBalances(); /* set up empty accounts */
    cout << "\nAdding some money to accounts: \n";
    cout << "\nAdding $1000 to saving \n";
    cout << "Adding $2000 to checking \n";
    my_bank.deposit(1000, "S"); /* deposit $1000 to savings */
    my_bank.deposit(2000, "C"); /* deposit $2000 to checking */
    my_bank.printBalances();
    cout << "\nTaking out $1500 from checking,and moving $200 from";
    cout << " savings to checking.\n";
    my_bank.withdraw(1500, "C"); /* withdraw $1500 from checking */
    my_bank.transfer(200, "S");  /* transfer $200 from savings */
    my_bank.printBalances();
    cout << "\ntrying to withdraw $900 from Savings.\n";
    my_bank.withdraw(900, "S");
    my_bank.printBalances();
    cout << "\ntrying to withdraw $400 from Checking.\n";
    my_bank.withdraw(400, "C");
    my_bank.printBalances();
    return 0;
}

```

Lab 4 PDF

Bank.cpp

```cpp
#include <iostream>
#include "Bank.h"
#include <iomanip>
void Bank::deposit(double amount, string account)
{
    if (account == "C")
    {
        checking.deposit(amount);
    }
    if (account == "S")
    {
        savings.deposit(amount);
    }
}
Bank::Bank()
{
}
Bank::Bank(double checkingAmount, double savingsAmount)
{
    checking.deposit(checkingAmount);
    savings.deposit(savingsAmount);
}
void Bank::withdraw(double amount, string account)
{
    if (account == "C")
    {
        checking.withdraw(amount);
    }
    if (account == "S")
    {
        savings.withdraw(amount);
    }
}
void Bank::transfer(double amount, string account)
{
    if (account == "C")
    {
        checking.withdraw(amount);
        savings.deposit(amount);
    }
    if (account == "S")
    {
        savings.withdraw(amount);
        checking.deposit(amount);
    }
}
void Bank::printBalances() const
{
    cout << fixed << setprecision(2);
    cout << "Savings Balance: " << savings.getBalance() << endl;
    cout << "Checking Balance: " << checking.getBalance() << endl;
}

```

Bank.h

```h
/*
Purpose- This is the header file for the Bank class. It's purpose is to call the account class to calculate the balance of the account then prints the balance.
@author Haichuan Wei
@version 1.0 10/10/21
@class function Bank() - the deafult constructor.
@class function Bank(double , double) - the constructor with paramenters of checkingAmount and savingsAmount.
@param checkingAmount - the amount of money in the checking account.
@param savingsAmount - the amount of money in the savings account.
@class function deposit(double, string) - this function will take in the amount and account type specified by the user and call the Account class deposit function witch will add the amount to the total balance.
@class function withdraw(double, string) - this function will take in the amount and account type specified by the user and call the Account class withdraw function witch will subtract the amount from the total balance.
@class transaction(double, string) - this function will take in the amount and account type specified by the user and then it add money to oppsite account specified by the user and remove it from the current account.
@param amount - the amount of money to be added or removed from the account.
@param account - the account type.
@class function printBalances() - this function will print the total balance of each account everytime it is called.
@return-none
*/

#ifndef _Bank_h_
#define _Bank_h_
using namespace std;
#include <string>
#include "Account.h"
class Bank
{
public:
    Bank();
    Bank(double checkingAmount, double savingsAmount);
    //deposit member function will call Account member function deposit
    void deposit(double amount, string account);
    void withdraw(double amount, string account);
    void transfer(double amount, string account);
    void printBalances() const;

private:
    Account checking; //   change ????? with correct data type for variable checking
    Account savings;  // use correct data type for variable savings
};

#endif //!_Bank_h_
```

Lab 4 PDF

Account.cpp

Account.cpp > ...

```cpp
#include <iostream>
#include "Account.h"

void Account::deposit(double amount)
{
    balance += amount;
}
void Account::withdraw(double amount)
{

    if (balance > amount)
    {
        balance -= amount;
    }
    else if (balance < amount)
    {
        cout << "Only " << balance << "$ available. But trying to withdraw " << amount << "$ removing 5$" << endl;
        balance -= PENALTY;
    }
}
double Account::getBalance() const
{
    return balance;
}
Account::Account()
{
    balance = 0;
}
```

Account.h

Account.h > ...

```cpp
/*
Purpose- The account class is used to calculate the balance of the user provided numbers and return the balance result.
@author Haichuan Wei
@version 1.0 10/10/21
@class function Account(): This is the deafult constructor of the class Account.
@class function deposit(double): This function is used to add the numbers to the balance varable.
@class function withdraw(double): This function is used to subtract the user defined number from the balance varable.
@class function getBalance(): This function is used to return the balance varable.
@param amount - the amount of money to be added or removed from the account.
@return-balance
*/

#ifndef _Account_h_
#define _Account_h_
using namespace std;
#include <string>
const double PENALTY = 5.0;

class Account
{
public:
    Account();
    void deposit(double amount);
    void withdraw(double amount);
    double getBalance() const;
    // void penalty(double amount) const;

private:
    double balance;
};
#endif //!_Account_h_
```

Lab 4 PDF

printMeFIrst.cpp

printMeFirst.cpp > ...

```cpp
#include "printMeFirst.h"
#include <string>
#include <iostream>
#include <iomanip>
#include <ctime>
using namespace std;

/*
Purpose- Prints the information of the developer.
@author Haichuan Wei
@version 1.0 9/2/21
@using  CLion
@param name - none
@param courseInfo - CS-116 OOP C++
@return-
*/

void printMeFirst(string name, string courseInfo)
{
    cout << "Program written by: " << name << endl;
    cout << "Course Info: " << courseInfo << endl;
    time_t now = time(0);
    char *dt = ctime(&now);
    cout << "Date: " << dt << endl;
}

```

Program Output

All values matched the example

```
arthur@DESKTOP-UP5LF24:/mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 6$ make
g++ -fsanitize=address -o BankAccount Account.cpp Bank.cpp BankMain.cpp printMeFirst.cpp
arthur@DESKTOP-UP5LF24:/mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 6$ make run
./BankAccount
Program written by: Haichuan Wei
Course Info: CS-116 - 2021 Spring
Date: Sun Oct 10 22:14:12 2021


Inital bank balances:
Savings Balance: 0.00
Checking Balance: 0.00

Adding some money to accounts:

Adding $1000 to saving
Adding $2000 to checking
Savings Balance: 1000.00
Checking Balance: 2000.00

Taking out $1500 from checking,and moving $200 from savings to checking.
Savings Balance: 800.00
Checking Balance: 700.00

trying to withdraw $900 from Savings.
Only 800.00$ available. But trying to withdraw 900.00$ removing 5$
Savings Balance: 795.00
Checking Balance: 700.00

trying to withdraw $400 from Checking.
Savings Balance: 795.00
Checking Balance: 300.00
arthur@DESKTOP-UP5LF24:/mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 6$ make clean
rm -rf BankAccount
arthur@DESKTOP-UP5LF24:/mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 6$
```