Khoi Duong

Prof. Yang

CS360L

8/8/2022

LAB#8

1.

Source code:

Account.h

```
#ifndef ACCOUNT_H
#define ACCOUNT_H

#include <iostream>
using namespace std;

class Account{
    public:
    double account_balance;
    Account(double x) {
        if (x > 0) {
            account_balance = x;
        }
        else {
            account_balance = 0;
            cout << "The initial balance is invalid. Set account balance to zero." << endl;
        }
    };
    virtual void credit(double x) {
        account_balance += x;
    };
    virtual void debit(double x) {
        if (account_balance >= x) {
            account_balance -= x;
        }
        else {
            cout << "Debit amount exceeded account balance. Transaction cancelled."
<< endl;
        }
    };
    virtual double getBalance() {
        return account_balance;
    };
};

#endif // ACCOUNT_H
```

SavingsAccount.h

```
#ifndef SAVINGSACCOUNT_H
#define SAVINGSACCOUNT_H
#include "Account.h"
#include <iostream>
using namespace std;

class SavingsAccount: public Account {
    public:
    double interest_rate;
    SavingsAccount(double x, double y) : Account(x) {
        if (y > 0) {
            interest_rate = y;
        }
        else {
            interest_rate = 0;
            cout << "The interest rate is invalid. Set interest rate to zero." <<
endl;
        }
    };
    double calculateInterest(){
        return (account_balance * interest_rate);
    }
};

#endif // SAVINGSACCOUNT_H
```

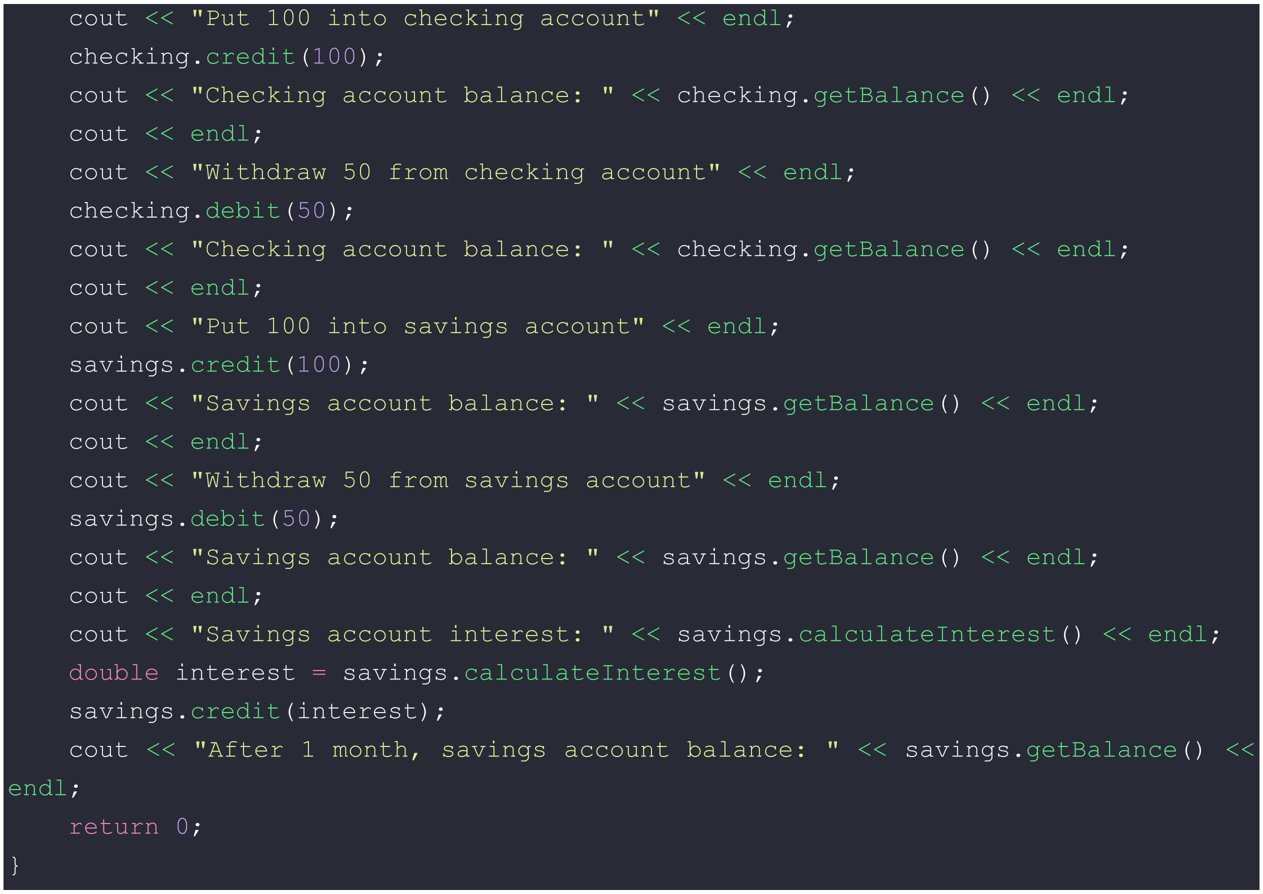
CheckingAccount.h

```
#ifndef CHECKINGACCOUNT_H
#define CHECKINGACCOUNT_H
```

```
#include "Account.h"
#include <iostream>
using namespace std;

class CheckingAccount: public Account {
    public:
    double transaction_fee;
    CheckingAccount(double x, double y) : Account(x) {
        if (y > 0) {
            transaction_fee = y;
        }
        else {
            transaction_fee = 0;
            cout << "The transaction fee is invalid. Set transaction fee to zero."
<< endl;
        }
    };
    void credit(double x) {
        Account::credit(x - transaction_fee);
    }
    void debit(double x) {
        if (account_balance >= x) {
            account_balance = account_balance - x - transaction_fee;
        }
        else {
            cout << "Debit amount exceeded account balance. Transaction cancelled."
<< endl;
        }
    };
};

#endif // CHECKINGACCOUNT_H
```

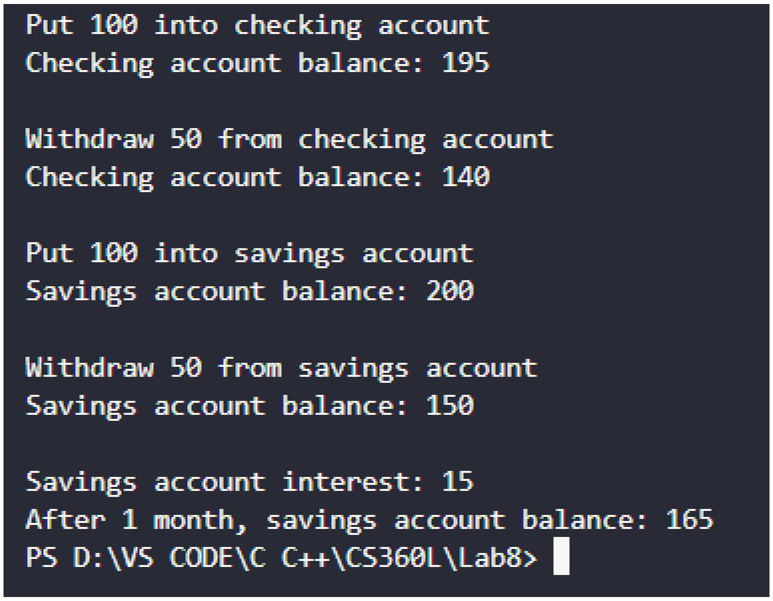
main.cpp

```
#include <iostream>
using namespace std;
#include "Account.h"
#include "CheckingAccount.h"
#include "SavingsAccount.h"

int main(){
    CheckingAccount checking(100, 5);
    SavingsAccount savings(100, 0.1);
```



```
    cout << "Put 100 into checking account" << endl;
    checking.credit(100);
    cout << "Checking account balance: " << checking.getBalance() << endl;
    cout << endl;
    cout << "Withdraw 50 from checking account" << endl;
    checking.debit(50);
    cout << "Checking account balance: " << checking.getBalance() << endl;
    cout << endl;
    cout << "Put 100 into savings account" << endl;
    savings.credit(100);
    cout << "Savings account balance: " << savings.getBalance() << endl;
    cout << endl;
    cout << "Withdraw 50 from savings account" << endl;
    savings.debit(50);
    cout << "Savings account balance: " << savings.getBalance() << endl;
    cout << endl;
    cout << "Savings account interest: " << savings.calculateInterest() << endl;
    double interest = savings.calculateInterest();
    savings.credit(interest);
    cout << "After 1 month, savings account balance: " << savings.getBalance() <<
endl;
    return 0;
}
```

Run program & result:



```
Put 100 into checking account
Checking account balance: 195

Withdraw 50 from checking account
Checking account balance: 140

Put 100 into savings account
Savings account balance: 200

Withdraw 50 from savings account
Savings account balance: 150

Savings account interest: 15
After 1 month, savings account balance: 165
PS D:\VS CODE\C C++\CS360L\Lab8>
```