HOMEWORK 03

THE ROAD OF MONEY MANAGEMENT.

GOAL (100PT)

- Given the base class Account, you are asked to develop two derived classes SavingAcount and CheckingAcount respectively for saving and checking accounts using public inheritance.
- You should not modify the main() function.

PROGRAM FOR YOU TO MODIFY(25PT)

- Base Class Definition: Account

```
class Account
{
public:
   Account(double = 0.0, double =0.0);
   void credit(double =0.0);  // Deposit money >0
   bool debit(double = 0.0);  // Withdraw money>0
   double getBalance(); // Get balance
   double calculateInterest(); // Return interest and add
the interest to the balance
   void print(); // print balance and interest rate
private:
   double balance;  // Account balance >=0
    double interestRate; // Interest rate >=0
};
```

- You can not add other functions.

MAIN() FUNCTION

```
double money , rate;
int main()
{
   cin >> money >> rate;
   cout << "Create a saving account." << endl;
   SavingAccount sAcnt(money, rate);
   sAcnt.print();
   sAcnt.debit(50.0);
   cout << "    New balance after withdrawing $50 from the saving account:" << sAcnt.getBalance() <<endl;
   sAcnt.credit(150.0);
   cout << "    New balance after depositing $150 to the saving account:" << sAcnt.getBalance() << endl;
   sAcnt.print();
   cout << "    Interest of the saving account:" << sAcnt.calculateInterest()<<endl;
   cout << "    New balance after adding interest:" <<sAcnt.getBalance()<<endl;
   cout << "    Withdrawing 800 from the saving account:" << endl;
   sAcnt.debit(800);

   cin >> money >> rate;
   cout << "\nCreate a checking account." <<endl;
   CheckingAccount cAcnt(money, rate);
   cAcnt.print();
   cAcnt.debit(200.0);
   cout <<" New balance after withdrawing $200 from the checking account:" << cAcnt.getBalance() << endl;
   cAcnt.credit(150.0);
   cout << " New balance after depositing $150 to the checking account:" << cAcnt.getBalance() << endl;

   cout << endl;
   cAcnt.print();
   sAcnt.print();

   cout << "\nAfter transfer $600 from cAcnt to sAcnt:" <<endl;
   CheckingToSaving(cAcnt,sAcnt,600.0);
   cout << "New balance of cAcnt:"<<cAcnt.getBalance()<<" New balance of sAcnt " <<sAcnt.getBalance() << endl;
   cout << "\nAfter transfer $800 from sAcnt to sAcnt" <<endl;
   SavingToChecking(sAcnt,cAcnt,800.0);
   cout << "New balance of cAcnt:"<<cAcnt.getBalance()<<" New balance of sAcnt " <<sAcnt.getBalance() << endl;
   CheckingToSaving(cAcnt,sAcnt,50.0);
   cout << "\nAfter transfer $50 from cAcnt to sAcnt" <<endl;
   cout << "New balance of cAcnt:"<<cAcnt.getBalance()<<" New balance of sAcnt " <<sAcnt.getBalance() << endl;
   SavingToChecking(sAcnt,cAcnt,50.0);
   cout << "\nAfter transfer $50 from sAcnt to cAcnt" <<endl;
   cout << "New balance of cAcnt:"<<cAcnt.getBalance()<<" New balance of sAcnt " <<sAcnt.getBalance() << endl;
}
```

SAVINGACCOUNT CLASS(25PT)

For your convenience, a summary of SavingAcount class is given below.

```
class SavingAccount {
public:
        SavingAccount(double = 0.0, double = 0.0, double = 3.0); // parameters: balance, interest rate, transaction fee.
        bool debit(double =0.0);
        void print();
private:
        double transactFee; // transaction fee for withdrawing
};
```

- transactFee is an amount of money paid to the bank by a saving account if a withdraw transaction is made on a saving account. No transaction fee is charged for deposition.
- debit() can only be done if balance remains positive after withdrawing. There is no transaction fee if a transaction fails.
- You can add other functions to complete the GOAL ,but you must at least have the functions shown in the picture.

CHECKINGACCOUNT CLASS(25PT)

For your convenience, a summary of CheckingAcount class is given below.

```
class CheckingAccount {
public:
        CheckingAccount(double = 0.0, double =0.0, double =3.0, double = 2.0); // Parameters: balance,
interest rate, transaction fee for withdraw, transaction fee for deposition
        bool debit(double =0.0); // return true if it can be done successfully.
        void credit(double =0.0);
        void print();
private:
        double transactFeeW; // withdraw double transactFeeD; // Deposit
};
```

- There is a transaction fee respectively for withdrawing and depositing if a transaction succeeds. Otherwise, no transaction fee is applied.
- debit() and credit()can only be done if their balance remains positive after transaction.
- You can add other functions to complete the GOAL ,but you must at least have the functions shown in the picture.

EXAMPLE

```
void SavingAccount::credit(double cre)
{
        Account::credit(cre);
}
```

- Both accounts must use inheritance, as shown above.

EXTRA GLOBAL FUNCTIONS(25PT)

bool CheckingToSaving(CheckingAccount&, SavingAccount&, const double);

- This function should transfer an amount of money from a checking account to a saving account. The checking account should pay a transaction fee for withdrawing. Return true when the transaction is successful.

bool SavingToChecking(SavingAccount&, CheckingAccount&, const double);

- This function should transfer an amount of money from a saving account to a checking account. The saving account should pay a transaction fee for withdrawing and the checking account should pay a transaction fee for deposition. Return true when the transaction is successful.

These two functions should make friend to CheckingAccount class and SavingAccount class.

EXAMPLE OUTPUT

```
300 0.05
Create a saving account.
SavingAccount account:
    Balance: 300
    Interest rate: 0.05
    Transaction fee of withdraw:3
    New balance after withdrawing $50 from the saving account:247
    New balance after depositing $150 to the saving account:397
SavingAccount account:
    Balance: 397
    Interest rate: 0.05
    Transaction fee of withdraw:3
    Interest of the saving account:19.85
    New balance after adding interest:416.85
    Withdrawing 800 from the saving account:
Debit amount exceeded account balance.
400 0.02

Create a checking account.
Checking account:
    Balance: 400
    Interest rate: 0.02
    Transaction fee of withdraw:3
    Transaction fee of deposition:2
 New balance after withdrawing $200 from the checking account:197
 New balance after depositing $150 to the checking account:345

Checking account:
    Balance: 345
    Interest rate: 0.02
    Transaction fee of withdraw:3
    Transaction fee of deposition:2
SavingAccount account:
    Balance: 416.85
    Interest rate: 0.05
    Transaction fee of withdraw:3

After transfer $600 from cAcnt to sAcnt:
Transfer transaction fails.
New balance of cAcnt:345 New balance of sAcnt 416.85

After transfer $800 from sAcnt to sAcnt
Transfer transaction fails.
New balance of cAcnt:345 New balance of sAcnt 416.85

After transfer $50 from cAcnt to sAcnt
New balance of cAcnt:292 New balance of sAcnt 466.85

After transfer $50 from sAcnt to cAcnt
New balance of cAcnt:340 New balance of sAcnt 413.85
```

SUBMIT YOUR HOMEWORK

- Please make sure again that your code is ready to be compiled.
  - If it can't be compiled, it will not be corrected this time (0 points)
  - TA will try to inform you and ask you to make up for it.
  - After uploading, you still need to complete the DEMO within the specified time.
- Please use s1234567.h 、 s1234567.cpp 、 and s1234567_Main.cpp as your file names
  - Replace s1234567 by your own student ID.
  - Pack all the file into a folder named "s1234567_hw3".
  - Zip it and Upload to Portal.
  - Attach a s1234567_hw3.txt if u want to add some additional information to TA.

Warning : 10 points will be deducted if one of them is violated.

FIGHTING~~~