

Name: Haichuan Wei

Lab: Lab 7

Date: 11/3/2021

## Program Description:

This program was made to practice using C++ class. There is a main BankAccont.cpp file that the other files inherit from. The user will define data that will be plugged into each function and whichever account they want to change will be changed accordingly.

## Program Source Code:

BankAccount.cpp

```
C++ BankAccount.cpp > ...
1  #include "BankAccount.h"
2  BankAccount::BankAccount()
3  {
4      balance = 0;
5      accountNum = "NA";
6  }
7
8  BankAccount::BankAccount(string acct, double a)
9  {
10     accountNum = acct;
11     balance = a;
12 }
13 void BankAccount::deposit(double amount)
14 {
15     balance += amount;
16 }
17 void BankAccount::withdraw(double amount)
18 {
19     if (amount > balance)
20     {
21         cout << "Insufficient funds! Trying to remove $" << amount << " But only $" << balance << "available! "
22         << "Not removing anything!" << endl;
23     }
24     else
25     {
26         balance -= amount;
27     }
28 }
29
30 double BankAccount::get_balance()
31 {
32     return balance;
33 }
34 void BankAccount::display_balance()
35 {
36     cout << accountNum << "'s balance is $" << balance << endl;
37 }
38 void BankAccount::set_account_number(string n)
39 {
40     accountNum = n;
41 }
42 string BankAccount::get_account_number()
43 {
44     return accountNum;
45 }
46
```

## Lab 7 PDF

### BankAccount.h

```
h BankAccount.h > BankAccount
1  #ifndef _BankAccount_h
2  #define _BankAccount_h
3  #include <iostream>
4  #include <string>
5  using namespace std;
6
7  class BankAccount
8  {
9  private:
10     double balance;
11     string accountNum;
12
13 public:
14     BankAccount();
15     BankAccount(string acct, double a);
16     void deposit(double amount);
17     void withdraw(double amount);
18     double get_balance();
19     void display_balance();
20     void set_account_number(string n);
21     string get_account_number();
22 };
23 #endif // _BankAccount_h
24
```

## SavingsAccount.cpp

```
SavingsAccount.cpp > month_end()
1  #include "SavingsAccount.h"
2  SavingsAccount::SavingsAccount()
3  {
4      interest_rate = 0;
5      min_balance = 0;
6  }
7  SavingsAccount::SavingsAccount(string acctNum, double b, double i)
8  {
9      BankAccount(acctNum, b);
10     interest_rate = i;
11     min_balance = b;
12 }
13
14 void SavingsAccount::withdraw(double amounts)
15 {
16     BankAccount::withdraw(amounts);
17     if (BankAccount::get_balance() < min_balance)
18     {
19         min_balance = BankAccount::get_balance();
20     }
21 }
22
23 void SavingsAccount::set_interest_rate(int rate)
24 {
25     rate = interest_rate;
26 }
27 double SavingsAccount::get_interest_rate()
28 {
29     return interest_rate;
30 }
31 void SavingsAccount::display_balance()
32 {
33     cout << "Account:" << BankAccount::get_account_number() << "'s interest-
34     rate is " << interest_rate << endl;
35     cout << BankAccount::get_account_number() << "'s Balance is $" <<
36     min_balance << endl;
37 }
38 void SavingsAccount::month_end()
39 {
40     BankAccount::deposit(min_balance);
41     double interest = min_balance * interest_rate / 100;
42     BankAccount::deposit(interest);
43     min_balance = BankAccount::get_balance();
44 }
```

SavingsAccount.h

```
h SavingsAccount.h >...
1 | #ifndef _SavingsAccount_h
2 | #define _SavingsAccount_h
3 | #include "BankAccount.h"
4 | #include "CheckingAccount.h"
5 | class SavingsAccount : public BankAccount
6 | {
7 | private:
8 |     double interest_rate;
9 |     double min_balance;
10 |
11 | public:
12 |     SavingsAccount();
13 |     SavingsAccount(string acctNum, double b, double i);
14 |     void withdraw(double amount);
15 |     void set_interest_rate(int rate);
16 |     double get_interest_rate();
17 |     void display_balance();
18 |     void month_end();
19 | };
20 | #endif // _SavingsAccount_h
```

## CheckingAccount.cpp

```
C> CheckingAccount.cpp > check_for_fee()
1  #include "CheckingAccount.h"
2  void CheckingAccount::check_for_fee()
3  {
4      transaction++;
5      const int FREE_TRANSACTIONS = 3;
6      const int TRANSACTION_FEE = 1;
7
8      if (transaction > FREE_TRANSACTIONS)
9      {
10         cout << "Number of Transactions: " << transaction << " is over free number of
11         transactions of: " << FREE_TRANSACTIONS << endl;
12         cout << "Deduct $" << TRANSACTION_FEE << " From account \n";
13         BankAccount::withdraw(TRANSACTION_FEE);
14         transaction = 0;
15         cout << "Number of Transactions:" << transaction << endl;
16     }
17     else
18     {
19         cout << "Number of Transactions:" << transaction << endl;
20     }
21 }
22 CheckingAccount::CheckingAccount()
23 {
24     transaction = 0;
25 }
26 CheckingAccount::CheckingAccount(string acctNum, double b)
27 {
28     BankAccount(acctNum, b);
29 }
30 void CheckingAccount::withdraw(double amount)
31 {
32     cout << "Checking Account Balance:" << endl
33     << endl;
34     BankAccount::withdraw(amount);
35     check_for_fee();
36 }
37 void CheckingAccount::deposit(double amount)
38 {
39     cout << "Checking Account Balance:" << endl
40     << endl;
41     BankAccount::deposit(amount);
42     check_for_fee();
43 }
44 void CheckingAccount::month_end()
45 {
46     transaction = 0;
47 }
48
49 }
```

## Lab 7 PDF

### CheckingAccounts.h

```
h CheckingAccount.h > ...
1  #ifndef _CheckingAccount_h
2  #define _CheckingAccount_h
3  #include "BankAccount.h"
4  class CheckingAccount : public BankAccount
5  {
6  private:
7      int transaction;
8      void check_for_fee();
9
10 public:
11     CheckingAccount();
12     CheckingAccount(string acctNum, double b);
13     void withdraw(double amount);
14     void deposit(double amount);
15     void month_end();
16 };
17 #endif // _CheckingAccount_h
```

## Lab 7 PDF

### bankTest.cpp

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 /*
6  * You need to use this test program for this lab
7  *
8  * You need to create SavingsAccount.h, CheckingAccount.h and-
9  * BankAccount.h files. The c++ codes for these header files
10  * are stored in a separate cpp file.
11  */
12
13 #include "SavingsAccount.h"
14 #include "CheckingAccount.h"
15 #include "BankAccount.h"
16 #include <vector>
17 #include <iomanip>
18 #include "printMeFirst.h"
19
20 int main ()
21 {
22     printMeFirst("Haichuan Wei", "CS-116 - Bank Inheritance Lab");
23     double amount;
24
25     // Create accounts
26
27     vector < BankAccount > myAccount;
28     BankAccount acct1;
29     BankAccount acct2 ("51002", 3000);
30     CheckingAccount myChecking;
31     SavingsAccount mySaving ("51001", 2500, 1.25);
32
33     cout << fixed << setprecision (2);
34     mySaving.set_account_number ("51001");
35     mySaving.display_balance ();
36     acct1.set_account_number ("51000");
37     acct2.set_account_number ("51002");
38     acct1.deposit (2000);
39     myAccount.push_back (acct1);
40     myAccount.push_back (acct2);
41     for (unsigned int n = 0; n < myAccount.size (); n++)
42     {
43         myAccount[n].display_balance ();
44     }
45
46     // withdraw from account acctNum
47     string acctNum = "51002";
48     double withdrawlet = 200;
49     for (unsigned int n = 0; n < myAccount.size (); n++)
50     {
51         if (myAccount[n].get_account_number () == acctNum)
52         {
53             cout << "Withdraw $" << withdrawlet
54                 << " from account " << acctNum << endl;
55             myAccount[n].withdraw (withdrawlet);
56             myAccount[n].display_balance ();
57         }
58     }
59
60     myChecking.set_account_number ("C1001");
61     myChecking.set_account_number ("C1001");
62     amount = 4500;
63     cout << "Account: " << myChecking.get_account_number ()
64         << " Deposit " << amount << endl;
65     myChecking.deposit (amount);
66     myChecking.display_balance ();
67
68     amount = 150;
69     cout << "Account: " << myChecking.get_account_number ()
70         << " Deposit " << amount << endl;
71     myChecking.deposit (amount);
72     myChecking.display_balance ();
73
74     amount = 100;
75     cout << "Account: " << myChecking.get_account_number ()
76         << " Withdraw " << amount << endl;
77     myChecking.withdraw (amount);
78     myChecking.display_balance ();
79
80     amount = 550;
81     cout << "Account: " << myChecking.get_account_number ()
82         << " Withdraw " << amount << endl;
83     myChecking.withdraw (amount);
84     myChecking.display_balance ();
85
86     cout << "\nSaving account month end\n";
87     mySaving.display_balance ();
88     mySaving.month_end ();
89     mySaving.display_balance ();
90
91     amount = 4000;
92     cout << "Account: " << mySaving.get_account_number ()
93         << " Withdraw " << amount << endl;
94     mySaving.withdraw (amount);
95     mySaving.display_balance ();
96
97     return 0;
98 }
99
```

## Code Output:

All data displayed matched the example

```

arthur@DESKTOP-UP5LF24:/ent/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 7$ make
g++ -fsanitize=address -o Bank BankAccount.cpp BankTest.cpp CheckingAccount.cpp SavingsAccount.cpp printMeFirst.cpp
arthur@DESKTOP-UP5LF24:/ent/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 7$ make run
./Bank
Program written by: Haichuan Wei
Course Info: CS-116 - Bank Inheritance Lab
Date: Wed Nov 3 22:36:44 2021

Account:S1001's interest rate is 1.25
S1001's Balance is $2500.00
S1000's balance is $2000.00
S1002's balance is $3000.00
Withdraw $200.00 from account S1002
S1002's balance is $2800.00
Account: C1001: Deposit 4500.00
Checking Account Balance:

Number of Transactions:1
C1001's balance is $4500.00
Account: C1001: Deposit 150.00
Checking Account Balance:

Number of Transactions:2
C1001's balance is $4650.00
Account: C1001: Withdraw 100.00
Checking Account Balance:

Number of Transactions:3
C1001's balance is $4550.00
Account: C1001: Withdraw 550.00
Checking Account Balance:

Number of Transactions: 4 is over free number of transactions of: 3
Deduct $1 From account
Number of Transactions:0
C1001's balance is $3999.00

Saving account month end
Account:S1001's interest rate is 1.25
S1001's Balance is $2500.00
Account:S1001's interest rate is 1.25
S1001's Balance is $2531.25
Account: S1001: Withdraw 4000.00
Insufficient funds! Trying to remove $4000.00But only $2531.25available!Not removing anything!
Account:S1001's interest rate is 1.25
S1001's Balance is $2531.25
arthur@DESKTOP-UP5LF24:/ent/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 7$

```