

MQF Object Oriented Programming I Fall 2019
Hw5 Due 12/7/2019 before midnight

Specifications

Create a system which will allow user to create bank account, delete account, perform various transactions, and print all accounts' information.

Requirements

Create a class name it **Account** which is the base account of checking account and saving account. In order to practice polymorphism, some of the member functions of Account are virtual. Your main testing program shall use a variable of type **vector<Account*>** to hold these accounts.

Testing

The following are some of the tests. Make sure that you test all the features of your system. You can assume user will enter valid input for this assignment if you are not working for extra credits.

Test creating accounts

```
Choose an Option
1. Transaction
2. Create New Account
3. Delete Account
4. Print All
5. Exit

Enter your choice -->> 2
Creating Account...
Please enter owner name: Peter
Initial amount: 123
Type of Account c (checking), s (saving): c
Your account number is 0

Choose an Option
1. Transaction
2. Create New Account
3. Delete Account
4. Print All
5. Exit

Enter your choice -->> 2
Creating Account...
Please enter owner name: Kimble
Initial amount: 222
Type of Account c (checking), s (saving): s
Your account number is 1
```

Create another account for Peter

```
Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 2
Creating Account...
Please enter owner name: Peter
Initial amount: 555
Type of Account c (checking), s (saving): s
Your account number is 2

Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit
```

Now print every account

```
Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 4
Print all Accounts

Account number -- 0
Account holder -- Peter
Checking account balance:      123.00

Account number -- 1
Account holder -- Kimble
Savings account balance:      222.00

Account number -- 2
Account holder -- Peter
Savings account balance:      555.00
```

Now test delete account

```

Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 3
Delete Account:
Please enter your account number: 1

```

See if really deleted by list all account information.

```

Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 4
Print all Accounts

Account number -- 0
Account holder -- Peter
Checking account balance:      123.00

Account number -- 2
Account holder -- Peter
Savings account balance:      555.00

```

Now test transaction (deposit). Note a submenu is showing up.

```

Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 1
In Transaction....

Choose an Option
  1. Deposit
  2. Withdraw
  3. Transfer from Checking to Saving
  4. Transfer from Saving to Checking
  5. Back to Main Menu

Enter your choice -->> 1
Enter Account Number: 0
Enter deposit amount: 100
    credit to account      100.00
Checking account balance:  223.00

```

Check if really do it right by listing all accounts

```
Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 4
Print all Accounts

Account number -- 0
Account holder -- Peter
Checking account balance:      223.00

Account number -- 2
Account holder -- Peter
Savings account balance:      555.00
```

Perform transaction (withdraw)

```
Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 1
In Transaction....

Choose an Option
  1. Deposit
  2. Withdraw
  3. Transfer from Checking to Saving
  4. Transfer from Saving to Checking
  5. Back to Main Menu

Enter your choice -->> 2
Enter Account Number: 2
Enter wthdraw amount: 55
    debit to account      55.00
Savings account balance:  500.00
```

List all and check the results

```
Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 4
Print all Accounts

Account number -- 0
Account holder -- Peter
Checking account balance:      223.00

Account number -- 2
Account holder -- Peter
Savings account balance:      500.00
```

Do transfer transaction. Move fund from checking to saving

```
Choose an Option
  1. Transaction
  2. Create New Account
  3. Delete Account
  4. Print All
  5. Exit

Enter your choice -->> 1
In Transaction....

Choose an Option
  1. Deposit
  2. Withdraw
  3. Transfer from Checking to Saving
  4. Transfer from Saving to Checking
  5. Back to Main Menu

Enter your choice -->> 3
Enter Checking Account Number: 0
Enter Saving Account Number: 2
Enter transfer amount: 23
  debit to account      23.00
Checking account balance:      200.00
  credit to account      23.00
Savings account balance:      523.00
```

Do transfer from saving to checking

```
Choose an Option
 1. Transaction
 2. Create New Account
 3. Delete Account
 4. Print All
 5. Exit

Enter your choice -->> 1
In Transaction....

Choose an Option
 1. Deposit
 2. Withdraw
 3. Transfer from Checking to Saving
 4. Transfer from Saving to Checking
 5. Back to Main Menu

Enter your choice -->> 4
Enter Checking Account Number: 0
Enter Saving Account Number: 2
Enter transfer amount: 100
    debit to account      100.00
Savings account balance: 423.00
    credit to account     100.00
Checking account balance: 300.00
```

See if we can exit.

```
Choose an Option
 1. Transaction
 2. Create New Account
 3. Delete Account
 4. Print All
 5. Exit

Enter your choice -->> 5
Bye!!
```

Grading rubrics

(1) Programming style	4
(2) Use of inheritance	4
(3) Use of polymorphism	4
(4) Appropriate testing	6
(5) Correct results	12

Submission:

- (1) Run your program and record the output by copy the screen shots (as detail as possible, it is very important for TA to reference). Paste all your screen shots into that document. On that document, before all the screen shots, put all team members' names on the first page of that document. If you do extra credits, mention on the second page of that file. Convert the document file into a pdf file so it is easier for TA to grade.
- (2) Put all your source code file or files (.cpp file, .h) into a folder. Use *yourLastName-FirstName-Hw5* as the name of the folder. E.g. if your name is Kathy Smith then *Smith-Kathy-hw5* is the folder name
- (3) Put the pdf file from (1) into the folder above
- (4) Zip the folder and Use *yourLastName-FirstName-Hw5* as the zip file name.
- (5) Submit the file to blackboard.

Extra Credit one: 2 points

Do error checking. For example amount transferred cannot be negative. Deposit amount cannot be negative, withdraw amount cannot be negative. User name cannot be null. Menu input cannot out of range.

Extra Credit two: 2 points

Friendly interface. Extend extra credit one, if user enter any illegal input, continue to ask the user to enter valid input until your system receive valid input.

Extra credit three: 1 points

Add interest. Extend the functionality of your system by adding interest rate and allow the user to set interest rate and compute interest then update balance. To simplify the computation of interest, you may use the current balance times the interest rate to compute the interest.

Extra credit four: 2 points

Minimum balance and service charge. Extend the functionality of your system by creating a *Student* account which has **gain no interest** but required **minimum balance**. Each transaction which causes the balance below the minimum balance, a service charge will impose on this account. Of course the service charge is set when account was created but it can be changed later by setting to a new value.

References

Self-check of chapter 12 polymorphism. Note you can modify the Part III question to fit your need here.

Chapter 10 the *new* operator.