C++ Group Project

```
DDL: will release on iSpace
At most 3 members for each team
Chooce your team on ispace
```

Description

Customer account

Note: the account number 1000 is reserved for bank manager. Customer account number starts from 1001.

In a bank, there are many types of accounts. To make it simple for this course project, there are only two types of accounts: `personal account` or `enterprise account`.

Account information is stored into a file in the following format (account.txt):

account	name	password	withdrawable	category	loan_amount	loan_quota_left	net_value
1001	Alita	123456	100000.23	personal	0	20000	100000.23
1002	Naruto	654321	101000.23	enterprise	1000	49000	100000.23

`account` - customer acount number starts from 1001, which contains 4-digit

`name` - customer name

`password` - login password which must be a 6-digit only

`withdrawable` - the amount of money customer can withdraw from the bank

`category` - personal or enterprise

`loan_amount` - record the amount that customer loans from the bank

`loan_quota_left` - the loan quota left for the customer. over the entire course, personal account can loan at most 20,000 and enterprise account can loan at most 50,000

`net_value` - net value of the account (withdrawable - loan_amount)

Account number (account) contains 4-digit. Password must be a 6-digit number only. For every account, the client can inquiry, withdraw, save, transfer, loan and repay.

1. `Inquiry`
    - When a personal or enterprise account is inquired, the balance for his/her account is displayed;
2. `Deposit`
    - A customer can deposit as much as possible. After deposits, customer's `withdrawable`, `net_value` and bank's `total_deposits` (see introduction in bank manager section below) should be updated.
3. `Withdraw`
    - For personal account, the customer can withdraw at most 2000 each time. For enterprise account, the customer can withdraw 5000 each time. If a customer wants to get money that exceeds this limit, a warning should be given.

- When a customer withdraws money from an account and the amount exceeds the budget in that account, the withdrawal fails. A warning must be given.
- Once withdraw successfully, bank's `total_deposits` should be updated.

4. `Transfer`
   - When transferring money, enterprise account can transfer at most 20,000 while personal account can transfer at most 10,000 each time. The withdrawable should be updated in the files if transaction success.
5. `Loan`
   - Bank offers loan service to customers, and the borrowable amount of the bank is 100000. Customers can loan from bank. If the customer wants to borrow the amount exceeding bank's borrowable amount, a warning message should be given.
   - Once money is loaned, customer's `withdrawable`, `loan_amount`, `loan_quota_left` and bank's `borrowable_amount` should be deducted accordingly, and data file should be updated accordingly as well.
6. `Repayment`
   - Customer is responsible for repayment. And he/she has to pay the loan interest according bank's interest rate, where personal account is 5% and enterprise account is 3%. For example, if a personal account loans 100 from bank, the customer should pay 105 to bank. Once finishing repayment, customer's `loan_amount`, `loan_quota_left`, bank's `borrowable_amount` and `total_deposits` should be updated accordingly once repayment is done.
7. `Change password`
   - Customer can change his/her password.

Use Case Demo

When the program starts, it requires user to input his/her account number and password to login.

```
Please input your account number:
```

If a valid account can be found in account.txt, the user can process to next step. Otherwise, a warning will be given and back to the account input.

```
Please input you account password:
```

Verify input with the password stored in the account.txt. If failed, a warning will be given and back to the account input.

Once log into successfully, it should display the function list:

```
1. Inquiry
2. Deposit
3. Withdraw
4. Transfer
5. Loan
6. Repay
7. Change password
8. Quit
Please Select:
```

Function list

After customer log into system, a menu list should be displayed as following:

a) `Inquiry`

Output the balance as following

```
Dear Alita, your balance is 100000.23
```

b) `Deposit`

Input how much money you want to deposit

```
Please input the amount to save: 10000
```

Verify if it is a reasonable amount, and output the updated balance

```
Dear Alita, your balance has been updated to 110000.23
```

c) `Withdraw`

Input the amount you want to withdraw

```
Please input the amount to withdraw: 10000
```

Verify if it is a reasonable amount, and output the updated balance. Otherwise warning must be given.

```
Dear Alita, your balance has been updated to 100000.23
```

d) `Transfer`

Input the target account number

```
Please input the target account number: 1002
```

Verify if it is a valid account. If not, a warning must be prompted. Next input the amount to transfer.

```
Please input the amount to transfer: 10000
```

Verify if it is a reasonable amount, and output the updated withdrawable amount. Otherwise warning must be given.

```
Transfer success! Your withdrawable amount has been updated to 90000.23
```

e) `Loan`

Input the amount to loan, verify if it is a reasonable amount (warning message should be given appropriately), and output the updated `withdrawable` and `loan_quota_left` entry.

```
Please input the amount to loan: 1000

Your withdrawable amount now is 91000.23

Your loan quota left is 19000
```

f) `Repayment`

Program will automatically calculate the total amount to repay to bank according to the customer' loan amount and bank's loan interest. Verify if he/she has enough money, and output the updated withdrawable amount. Otherwise warning should be given. Note that 89950.23 = 91000.23 - 1000*(1+0.05). After customer's repayment, bank receives the 1050, and bank's `borrowable_amount` is increased with that amount.

```
Cheers! Your loan amount now is 0. Your loan quota now is 20000.

Your withdrawable amount has been updated to 89950.23
```

g) `Change password`

```
Please input your original password:
```

Here you input your original password otherwise you cannot change it with a warning.

```
Please input your new password:

Please input your new password again:
```

Customer is required input two times with the same new password otherwise it will fail to change password with a warning.

```
Update password successfully.
```

After each operation, you can choose

```
1. Continue //Back to Operation menu
2. Quit     //Terminate the program
```

Bank manager account

Note: the account number 1000 is reserved for bank manager.

The bank manager can log into the system to

- create customer account;
- check `borrowable amount` of the bank;

- check the specified customer's information(name, withdrawable, enterprise etc, not included password) according to account number;
- reset password for customer

Bank information is stored into a file in the following format (bank.txt):

account	name	password	borrowable_amount	total_deposits	loan_interest_personal	loan_interest_enterprise
1000	manager	123456	1,000,0000	200000.46	5%	3%

account - Bank manager account

name - Bank manager name

password - Login password

borrowable_amount - A bank has a fixed amount money to lend

total_deposits - Total amount of deposits in the bank

loan_interest_personal - Interest rate personal account needs to pay

loan_interest_enterprise - Interest rate enterprise account needs to pay

Function list for the bank manager:

```
1. Create a new account
2. Check borrowable amount of bank
3. Check total_deposits of bank
4. Check customer's information
5. Reset password for customer
6. Quit
```

Scenario

1. Create a new account

```
Please input a customer information

(name   password    withdrawable    category    enterprise )
```

Input:

```
Alita   123456  100000.23   personal    y
```

Note, the first customer's account starts from 1001. The ID grows sequentially with new user acount created.

2. Check borrowable_amount of bank

Display the borrowable amount result.

```
The borrowable amount: 100000
```

3. Check total_deposits of bank

Display the total deposits of the bank.

```
The total deposits amount now is: 200000.46
```

4. Check customer's information

```
Please input the account number:  1001
```

account	name	withdrawable	category	loan_amount	loan_quota_left	net_value
1001	Alita	100000.23	personal	0	20000	100000.23

5. Reset password for customer

Manager can help reset password for costumer according to account number.

```
Please input the account number: 1001

Please input the new password: 123321

Please input the new password again: 123321

Cheers! Password updated !
```

Note, when the information of bank or customer is changed, you should update the data files.

Notice

1. Use classes and inheritance in C++.
2. Separate main function & class definition in different file.

Submission

compressed into one zip file includes:

1. code(.cpp files and .h files)
2. .exe and .txt file
3. Document (one page)
    - User manual
    - Job allocation
4. Each group only needs to submit **one copy**