Account

```
AccountChecking    derived
AccountSaving      derived
```

Accounts
Container for Account class

AccountSystem
Operations to perform on accounts

Account

Default constructor
Delete constructor
copy assignment operator
+= operator overload

Deposit
Withdraw

get/set ID
get/set name
get/set balance

ID (int*)
name (string)
balance (float)

Accounts

Default constructor
Delete constructor

Add account
findID
at(int i)

getSize

MAX_SIZE(const int)
accounts (Account*)
size (int)

AccountSystem

Add
Deposit
Withdraw
Transfer
PrintAllAccounts

accts (Accounts)

AccountChecking
derived from Account
transactionFee (float)
PrintInfo

AccountSaving
derived from Account
minBalance (float)
PrintInfo

Account

Deposit
- Take transaction as parameter and add to account balance

Withdraw
- Take transaction as parameter and subtract from account balance

+=

- Ask user for transfer amount
- If the transfer amount is greater than the balance, do not perform operation
- otherwise, withdraw money from source account and deposit into destination account
- return the destination account

PrintInfo
- prints info of account in formatted way

AccountChecking
- derived from Account

Deposit
- calls Account::Deposit using transaction fee

Withdraw
- calls Account::Withdraw using transaction fee

+=
- same as Account::+= only factoring in transaction fee

AccountSaving
- derived from Account

Deposit
- calls Account::Deposit only if minBalance is not passed

Withdraw
- calls Account::Withdraw only if minBalance is not passed

+=
- same as Account::+= if minBalance is not passed

Accounts
Add
- push_back the new account parameter in vector

findID
- Loop through all accounts to see if there is a matching id
- if so, return the account

at
- return the account at the index specified by the parameter

getSize
- return size of vector

loadAccounts
- read info from file and store corresponding checking and saving accounts to accounts vector

storeAccounts
- writes account vector information to file in format

AccountSystem
Add
- Prompt user for the account ID
- Check if ID already exists in accts and if so, do not make a new account
- Prompt user for information of account
- Create an account variable and store user information in it

- ● Add account to accts using add() method of Accounts
- ● Inform user account has been added

Deposit
- ● Check if accts size is > 0, if not end method
- ● prompt user for the account ID
- ● Loop through accts to find account
  - ○ if found, ask user for deposit amount
  - ○ Deposit money using at() method of Accounts and deposit() method of Account
  - ○ inform user of deposit
- ● If not found, inform user account has not been found

Withdraw
- ● Check if accts size is > 0, if not end method
- ● prompt user for the account ID
- ● Loop through accts to find account
  - ○ if found, ask user for deposit amount
  - ○ Withdraw money using at() method of Accounts and withdraw() method of Account
  - ○ inform user of withdrawal
- ● If not found, inform user account has not been found

Transfer
- ● Check if accts size is > 0, if not end method
- ● make two boolean variables to store matches of the two accounts and initialize as false
- ● Prompt user for the source account id and destination account id
- ● Loop through accts to find the accounts
  - ○ if an account is found, turn the corresponding boolean variable to true
- ● if both the accounts were found, transfer money between accounts using += operation of Account
- ● If either account wasn't found, inform user account has not been found and end method

PrintAllAccounts
- ● Check if accts size is > 0, if not end method
- ● Loop through accts and print each account information in nice formatted way

loadAccountSystem
- ● call accts loadAccounts()

storeAccountSystem
- ● call accts storeAccounts()