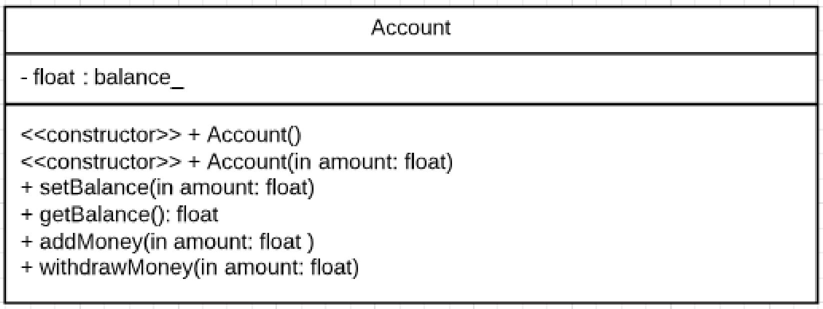
Computing 2 - Labs

Lab 2: Account Class

Define and implement an `Account` class that a bank might use to represent customers' bank accounts according to the following UML diagram.



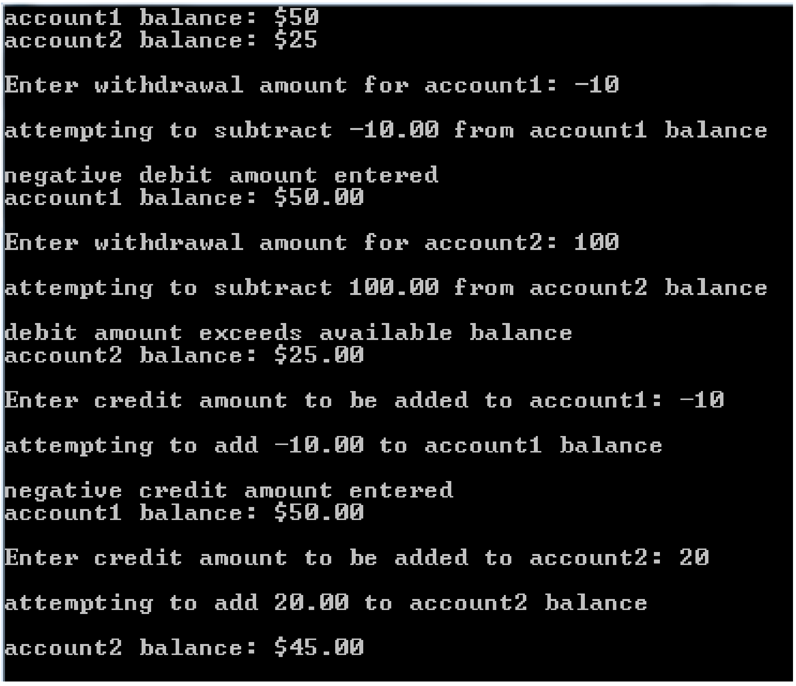
Account

- float : balance_

<<constructor>> + Account()
<<constructor>> + Account(in amount: float)
+ setBalance(in amount: float)
+ getBalance(): float
+ addMoney(in amount: float)
+ withdrawMoney(in amount: float)

Tasks:

1. Define the class in the header file Account.h.
2. In Account.cpp, implement the default constructor `Account( )` that initializes the balance to 0. In addition, program another constructor `Account( float amount ).`It receives an initial balance and uses it to initialize the data member `balance_`. The constructor should validate the initial balance to ensure that it's greater than or equal to 0. If not, set the balance to 0 and display an error message indicating that the initial balance was invalid.
3. Also, implement the three member functions in Account.cpp.
   a. Member function `void setBalance(float amount)` to set the current balance.
   b. Member function `float getBalance()const` to return the current balance.
   c. Member function `void addMoney( float amount )` to add an amount to the current balance. Ensure that the amount added is positive. If a negative amount has been entered display an error message and return.
   d. Member function `void withdrawMoney( float amount )` to withdraw money from the account. Ensure that the debit amount is positive and that it does not exceed the account's balance. If a negative amount has been entered, display an error message and return. If the credit amount exceeds balance, leave balance unchanged and print a message "Amount to be withdrawn exceeded account balance."
4. A `main` function that creates two `Account` objects and tests the member functions of class `Account` has been provided (see TestAccountClass.cpp).
5. Test drive the class. Display all floating point numbers with two decimals as shown below.



```
account1 balance: $50
account2 balance: $25

Enter withdrawal amount for account1: -10

attempting to subtract -10.00 from account1 balance

negative debit amount entered
account1 balance: $50.00

Enter withdrawal amount for account2: 100

attempting to subtract 100.00 from account2 balance

debit amount exceeds available balance
account2 balance: $25.00

Enter credit amount to be added to account1: -10

attempting to add -10.00 to account1 balance

negative credit amount entered
account1 balance: $50.00

Enter credit amount to be added to account2: 20

attempting to add 20.00 to account2 balance

account2 balance: $45.00
```