(BASE CLASS account)

**Class Account:**

* **Private:** Store a double balance, string array for transaction history, integer increment index to help store logs of transactions in correct spots on transaction array
* **Public:** Constructor that initializes balance to 0 and sets first log in transaction array to “Balance is 0”
  + **Virtual void getBalance():** Cout balance
  + **Account& operator+(double num)** will take in a double amount and add it to balance, add one to increment index, will use increment index to log balance change in transaction array, return itself incase more operators follow
  + **Virtual Account& operator-(double num)** will take in a double amount and subtract it from balance, add one to increment index, will use increment index to log balance change in transaction array, return itself incase more operators follow
  + **Void getTransactionHistory()** Display transaction history with a for loop printing each element in array

(Derived Account classes)

**Class Savings: public Account:**

* **Public:** 
  + **Void getBalance() override** Display Balance with Savings label in front
  + **Account& operator-(double num) override** Stops balance from going below 0, will take in a double amount and subtract it from balance, add one to increment index, will use increment index to log balance change in transaction array, return itself incase more operators follow

**Class Checkings: public Account**

**- Public:**

-**Void getBalance() override** display balance with a label for Checkings

(Base class)

**Class User**

* **Private:** stores string name, string id, int password
* **Public:** Constructor that initializes name to “Generic Name”, string id to “001”, int to “1234”.
  + **Checkings account instance called userCheckings**
  + **Savings account instance called userSavings**
  + **Void deposit()**
    - **While loop**
      * user presses 1: add to checkings balance using addition operator overloader function, exit loop
      * user presses 2: add to savings balance using addition operator overloader function, exit loop
      * User presses 3: cancel, exit loop
      * Else: run the loop again
  + **Void Withdraw()**
    - **While loop**
      * user presses 1: subtract from checkings balance using subtraction operator overloader function, exit loop
      * user presses 2: subtract from savings balance using subtraction operator overloader function, exit loop
      * User presses 3: cancel, exit loop
      * Else: run the loop again

- **String getName()** return name

- **String getId()** return id

- **int Getpassword()** return password

-**Void view balance(string name, string id)** will take in users name and id. Will display users name. Will display users Id with a C appended on end for Checkings. Using userCheckings .getBalance to show users checkings balance. Will display users Id with a S appended on end for Savings. Using userSavings .getBalance to show users savings balance.

-**void checkPassword()** will make user enter password before continuing on

-**void changepassword(int newPassword)** will take in new password and replace old

-**void changeName(string name)** will take in new name and replace old one

**-Main Function**

**Instance Of User called U**

Display all user information

**While loop**

Give user options :

0 end program.

1 display balance.

2 display transaction history

3. Deposit

4.Withdraw

5. Change name

6. Change password

**Take in users choice**

**Switch statement (choice)**

**1: Checkpassword** Make user enter password

-if right

-**U.viewBalance(U.getName(),U.getId())** plugs in user name and id into view balance for the display

**-**if wrong

**-**cout wrong password

Break to loop

**2: Checkpassword** Make user enter password

-if right

-if user selects one for savings

-U.usersavings.getTransactionHistory()

-if user selects two for checkings

-U.userchecking.getTransactionHistory()

-if user enters else

-cout invalid response

**-**if wrong

**-**cout wrong password

Break to loop

**3: Deposit** Make user enter password

-if right

-**U.deposit();** will run deposit function that leads to the addition operator overloaders

**-U.viewBalance(U.getname(),U.getId())** plugs in user name and id into view balance for the display

**-**if wrong

**-**cout wrong password

Break to loop

**3: Withdraw** Make user enter password

-if right

-**U.withdraw();** will run withdraw function that leads to the subtraction operator overloaders

**-U.viewBalance(U.getname(),U.getId())** plugs in user name and id into view balance for the display

**-**if wrong

**-**cout wrong password

Break to loop

**5: Change Name** Make user enter password

-if right

-**U.changeName(newName)**

Take in a name and plug it into the change name function

**-**if wrong

**-**cout wrong password

Break to loop

**6: Change Password** Make user enter password

-if right

-**U.changePassword(newPassword)**

Take in a password and plug it into the change password function

**-**if wrong

**-**cout wrong password

Break to loop