Keith Nakamatsu
ICS 355
11/18/2017

Introduction

The Program created is 99% finished but can complete simple tasks that were asked. This report will be divided into several sections. Section 1 will contain what was completed and what was not. Section 2 will discuss how completed items were implemented and what I tried to do for the non completed items. Keep in mind that I did try to implement fully working systems but could not figure certain things out. Section 3 will contain the assumptions and how to run the program.

Section 1

This program is a simple command-line driven financial calculator that supports multiple currencies and can add, subtract values from a user's account. In addition to this, it prevents the user from entering invalid data, such as currencies it does not support. In addition to this, it supports multi-users including an admin account by default, and this account can not be deleted. The admin account can create and delete other accounts; default settings were set to 0 USD. In addition to this, each account is protected by a password that is created by the user on creation of the account and this password is hashed with MD5 and salt. Also, this program can transfer money from one account to another but they need to have the same currencies, I could not figure out how to do a transfer and conversion at the same time.

Section 2

This program used C++ programing language, the reason I choose this language is that I learned object-oriented programming with it. OOP allows the admin to create data classes and subclasses that share some or all of the main classes characteristics. Meaning it is very simple to add additional functionality if needed.

There are five classes that were created, Currency, Currency_Collection, Money, User, and User_Collection. The Currency class keeps track of the different currency conversions that are allowed. Along with the USD dollar equivalent. I decided to use a golden standard for my conversion table. Most of the code took place in the User_Collection class. There will be comments within the code explaining what each function does.

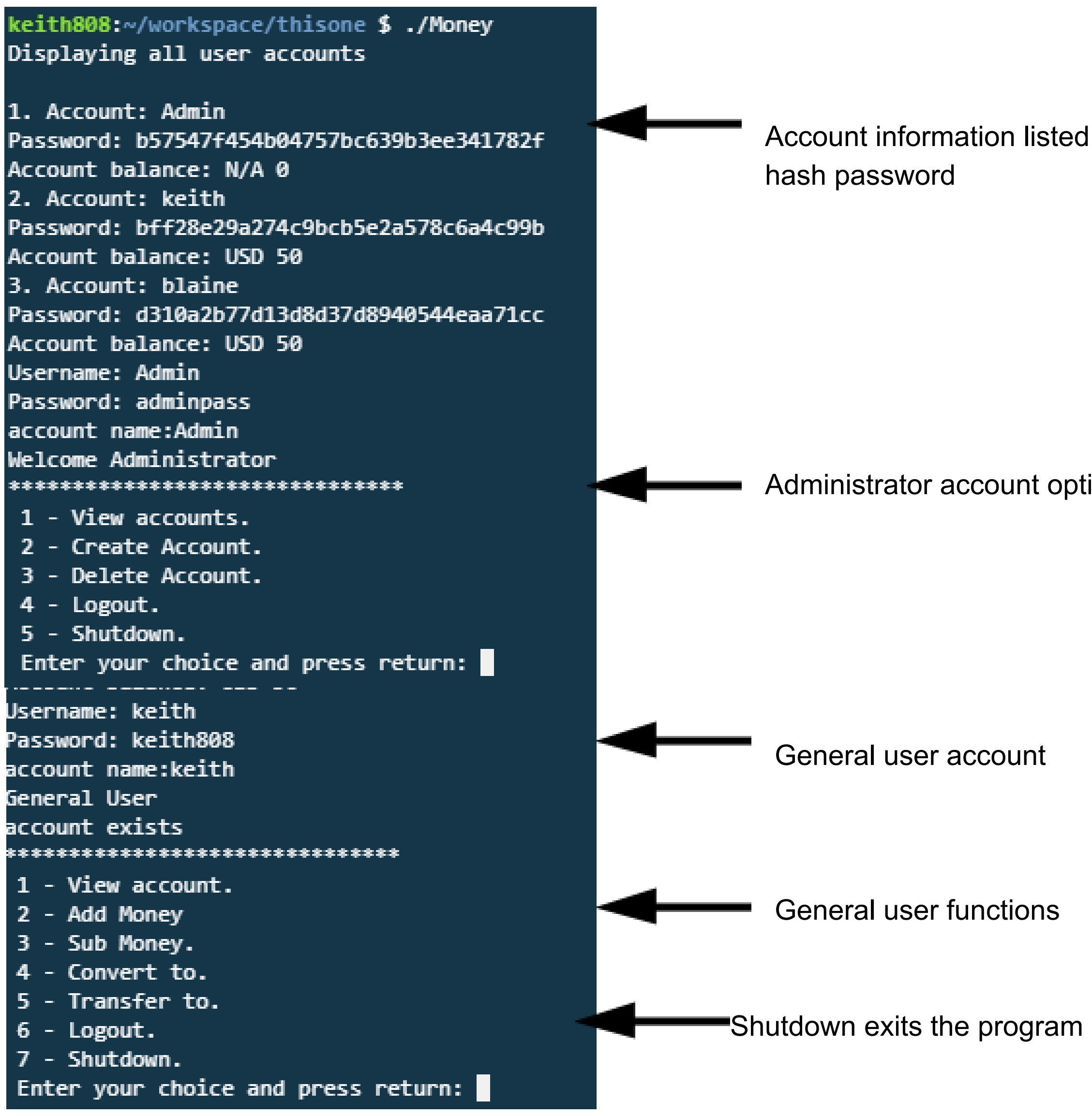
The only thing that does not work is the on the fly transfer and conversion combination. What I tried to do is convert the receiving account to the sending accounts currency transfer the money then convert the receiving account back to its original currency but could not get this implemented

User_Collection class now can create new accounts through the administrator account. There are several boolean statements that are used to check if the account is an administrator account or a general user, default settings are used for a newly created account. In addition to this, the amount of money that can be transferred between accounts has to be less than or equal to the amount of money in the account.

Keith Nakamatsu
ICS 355
11/18/2017

Section 3
To run the program, you need to have four files within the same directory, Money.cpp, con.txt, UserCollection.txt, md5.cpp, md5.h and the makefile. Just as a note there is a warning I receive every time. The warning message will be posted below.

```
Money.cpp:413:310: warning: non-static data member initializers only available with -std=c++11
or -std=gnu++11 [enabled by default]
   const char alphanum[100]
={'0','1','2','3','4','5','6','7','8','9','!','@','#','$','%','^','&','*','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O
','P','Q','R','S','T','U','V','W','X','Y','Z','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x
','y','z'};
Money.cpp:416:42: warning: non-static data member initializers only available with -std=c++11
or -std=gnu++11 [enabled by default]
   int stringLength = sizeof(alphanum) - 1;
                                                                      ^
```



```
keith808:~/workspace/thisone $ ./Money
Displaying all user accounts

1. Account: Admin
Password: b57547f454b04757bc639b3ee341782f
Account balance: N/A 0
2. Account: keith
Password: bff28e29a274c9bcb5e2a578c6a4c99b
Account balance: USD 50
3. Account: blaine
Password: d310a2b77d13d8d37d8940544eaa71cc
Account balance: USD 50
Username: Admin
Password: adminpass
account name:Admin
Welcome Administrator
*******************************
 1 - View accounts.
 2 - Create Account.
 3 - Delete Account.
 4 - Logout.
 5 - Shutdown.
 Enter your choice and press return:
```

```
Username: keith
Password: keith808
account name:keith
General User
account exists
*******************************
 1 - View account.
 2 - Add Money
 3 - Sub Money.
 4 - Convert to.
 5 - Transfer to.
 6 - Logout.
 7 - Shutdown.
 Enter your choice and press return:
```

Account information listed with salted hash password
Administrator account options
General user account
General user functions
Shutdown exits the program