CS 3307A - Object-Oriented Design and Analysis

C++ Assignment: The Banking System”

Joel Kennedy 250 813 112 jkenn24@uwo.ca

October 2016

1) Deliverable 1 – Minimum Set of Requirements: [8 max]

**Customer:**

C1) Withdraw from an account

C2) Deposit to an account

C3) Check the balance of an account

C4) Transfer between account

C5) Print all account info

C6) Warning messages for insufficient funds and other input errors

C7) Warning message for chequing accounts to have a service fee if balance drops below $1000

**Manager:**

M1) Open an account

M2) Close an account with a zero balance

M3) Get info on a specific customer

M4) Get info on all customers

**Technician/ maintainer:**

T1) Check the trace state

T2) Toggle execution state off

T3) Toggle execution state on

2) Deliverable 2 – Enhancement Requirements: [4 max]

The enhanced requirements I have chosen to implement are:

E1) The ability for a customer to E-transfer to another customer

E2) Each of the customers, the manager, and the technician have an ID and a password which they set and can change at any time

E3) If a customer forgets their password, it can be retrieved by a manager at any time

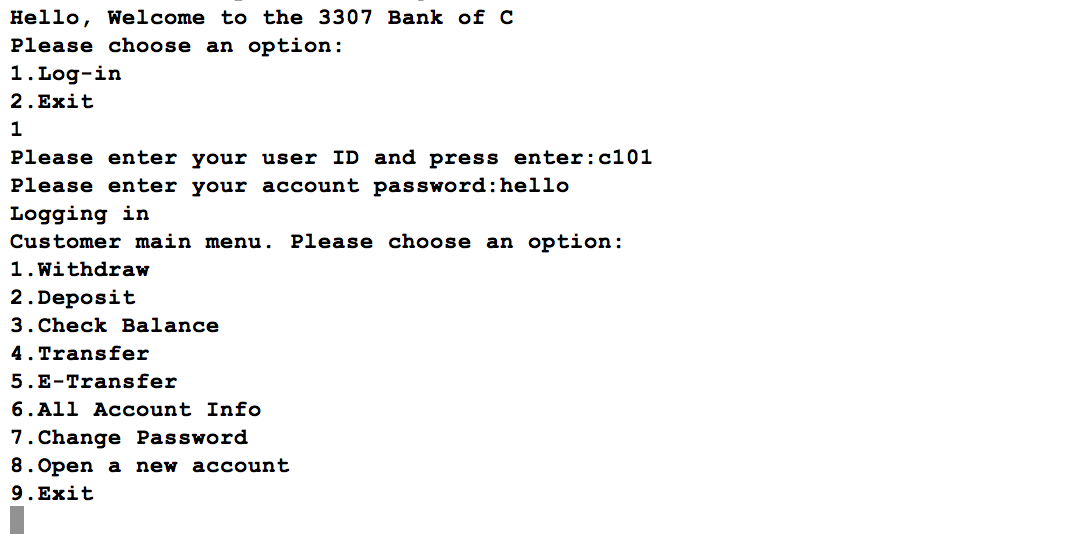
E4) When a customer has been given a “profile” by a manager, the customer can open up to 20 total accounts on their profile without the manager

4) Deliverable 4 -- Scenarios: [30 max – 6 marks per scenario fully functional]

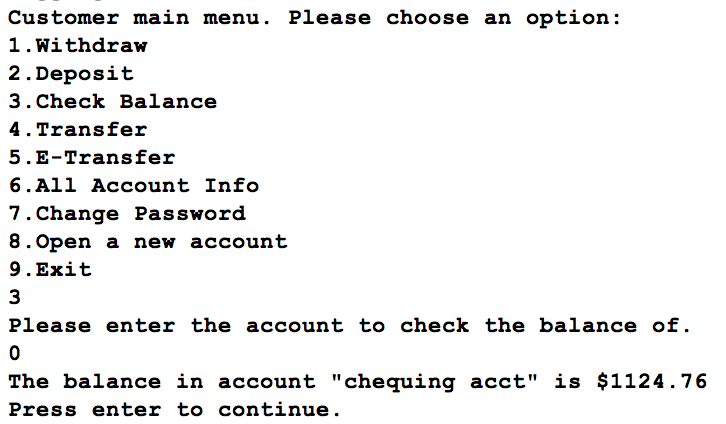
**Customer:**

Withdrawing funds

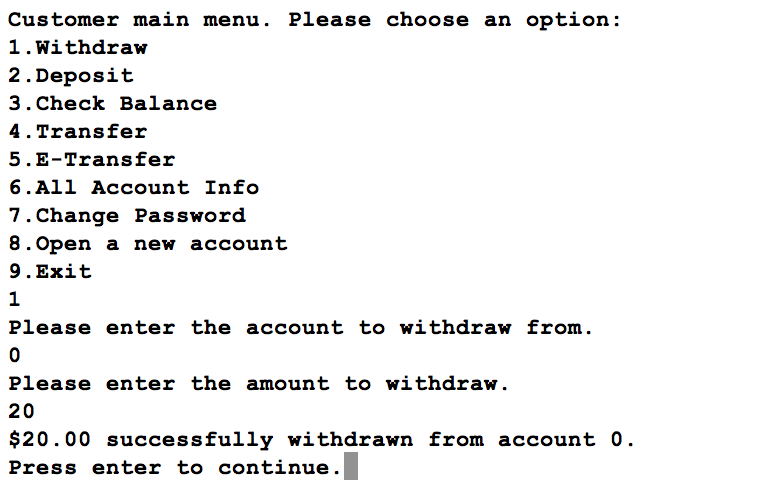
1. Customer logs in



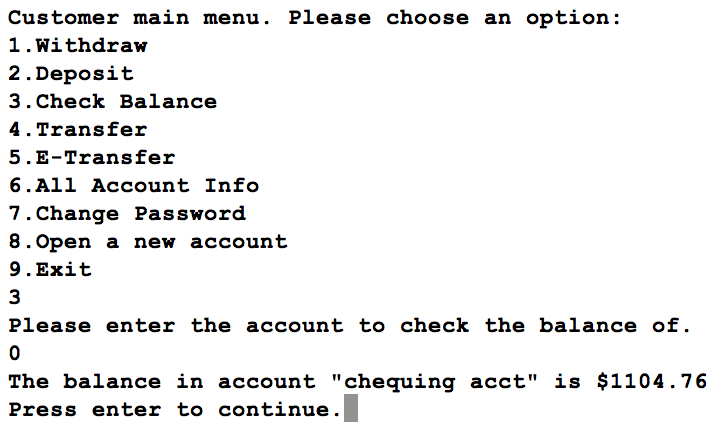
1. Checks balance of an account (account 0 in this example)



1. Chooses the deposit feature and deposits $20 into account 0
2. System withdraws the funds



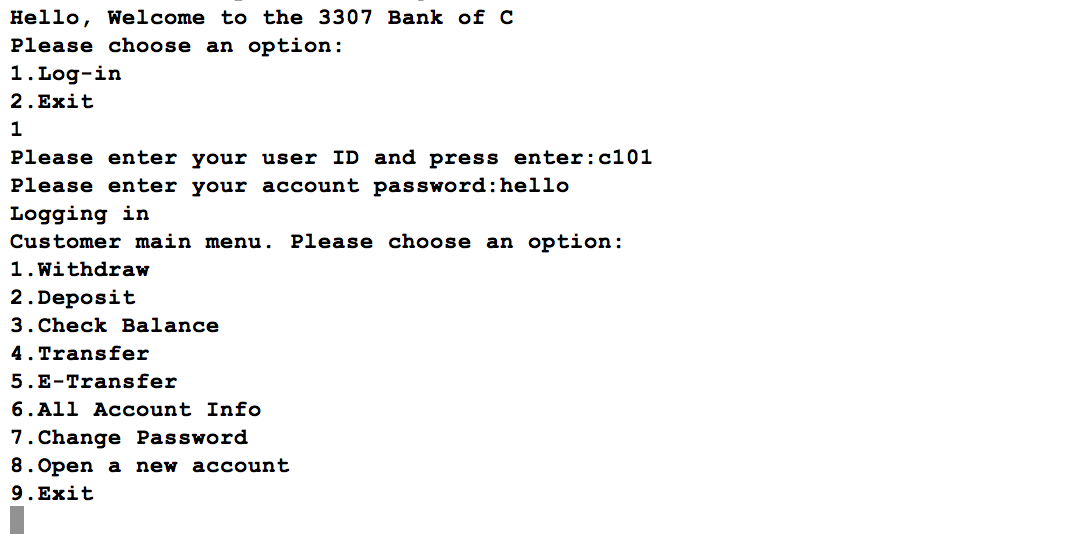
1. Checks the balance again to verify the deposit



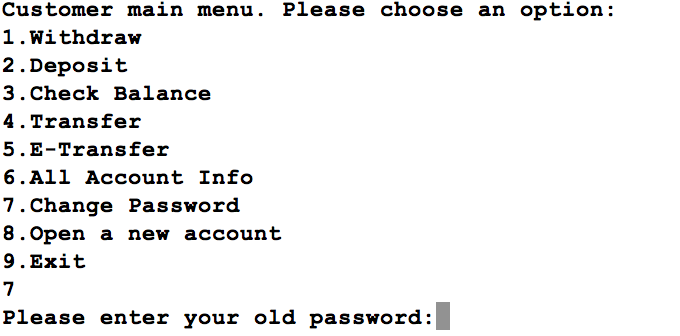
**Customer:**

Changing password

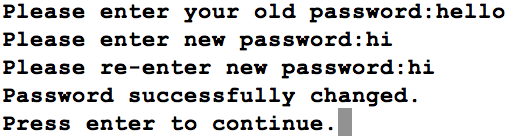
1. Customer logs in



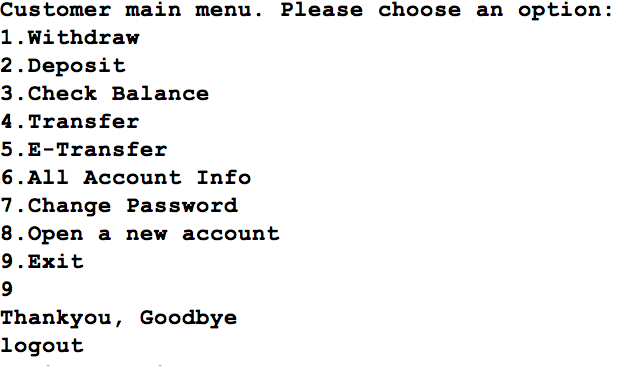
1. Chooses the change password option



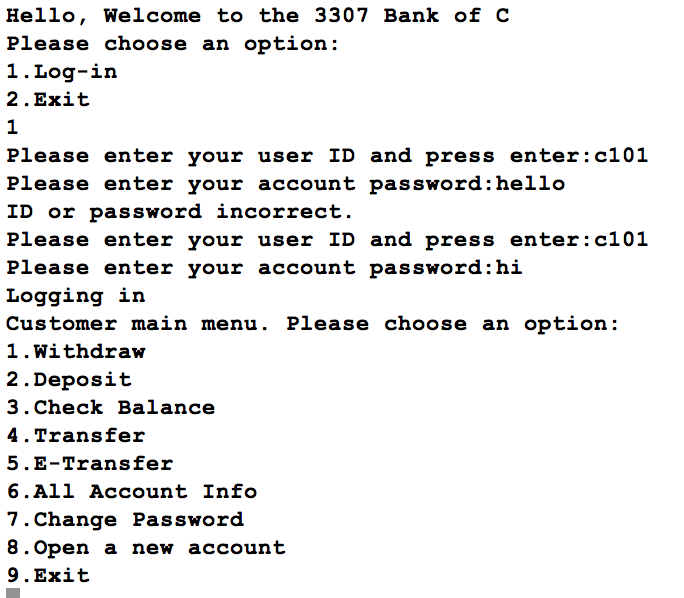
1. Enters their old password
2. Enters the new password twice
3. System changes password



1. Logs off



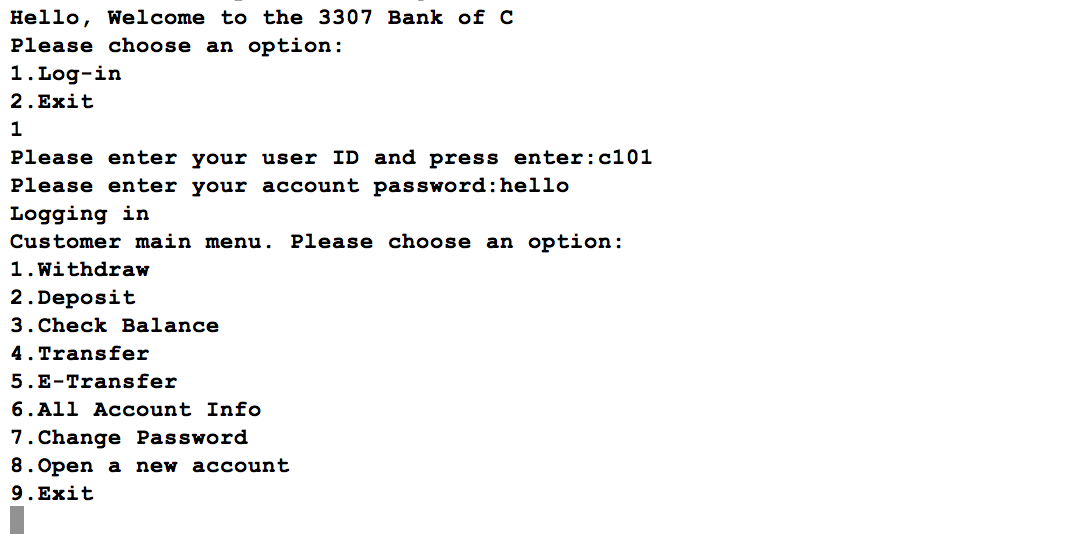
1. Logs back in to verify the password is different now



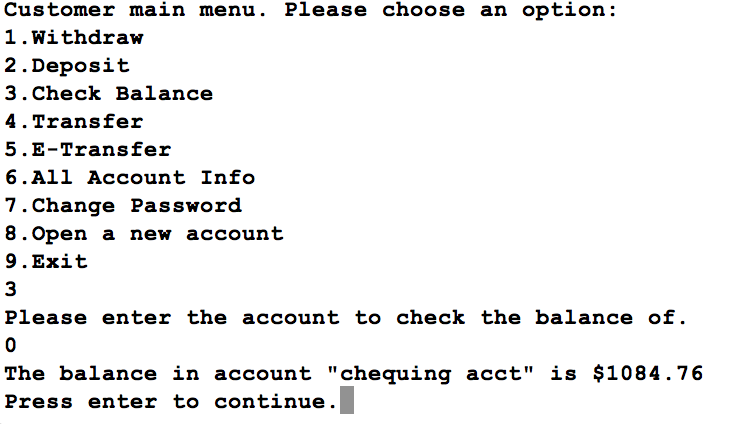
**Customer:**

Transferring Funds

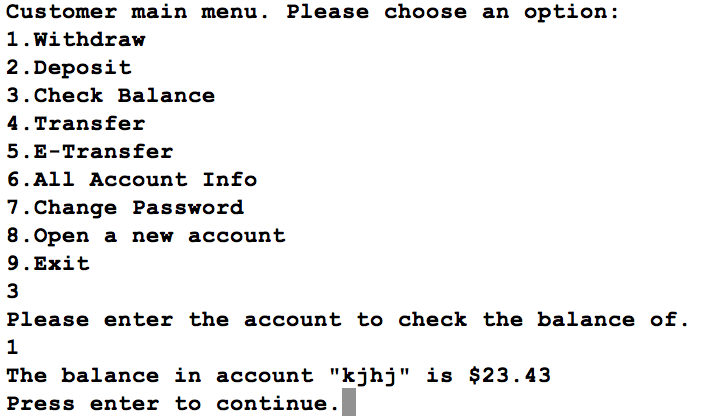
1. Customer logs in



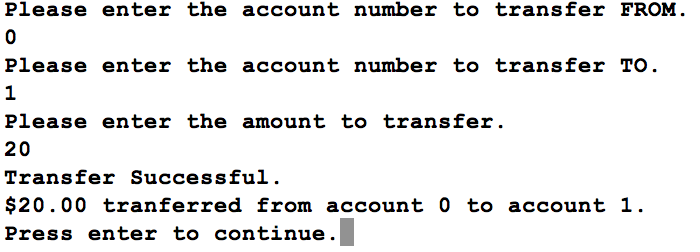
1. Checks balance of an account (account 0 in this example)



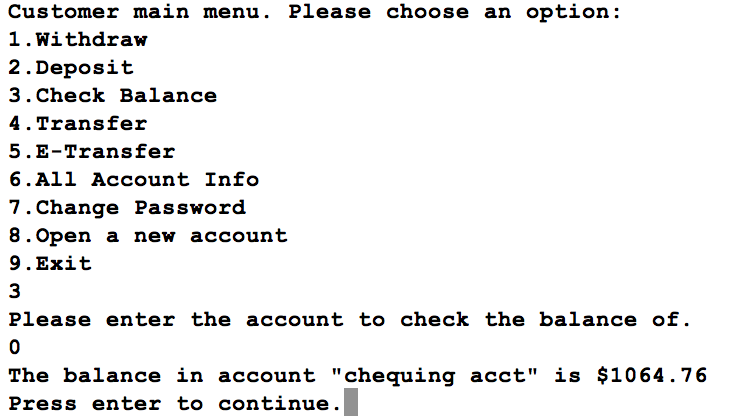
1. Checks the balance of a second account (account 1 in this example)

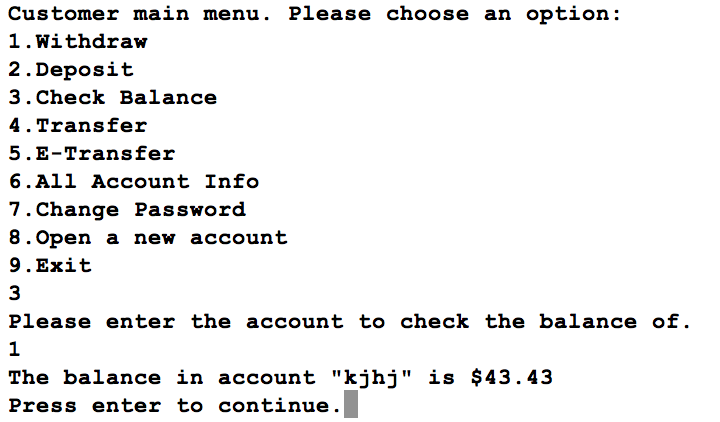


1. Chooses the transfer feature from account 0 to 1 in the amount of $20
2. System transfers the funds



1. Checks the balance again of account 0 to verify the deduction

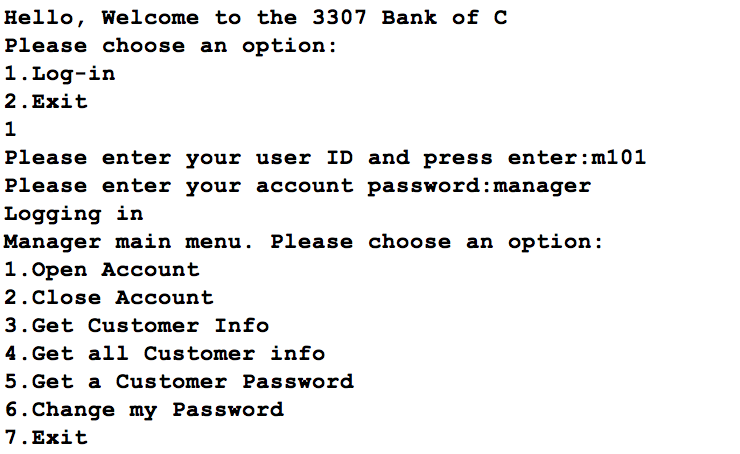


1. Checks the balance of account account 1 to verify the credit  
   

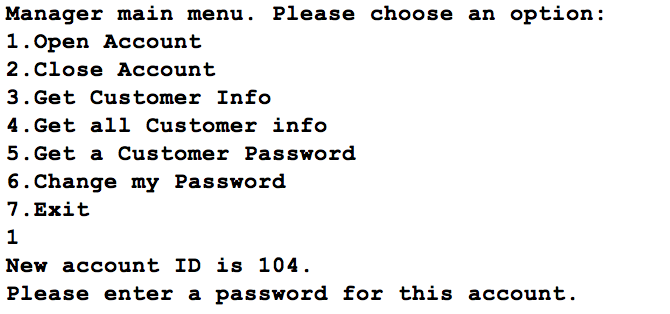
**Manager:**

Opening an account and retrieving the password

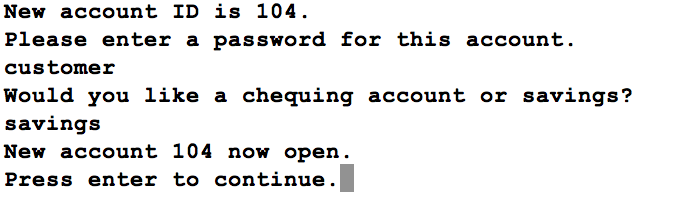
1. Manager logs in



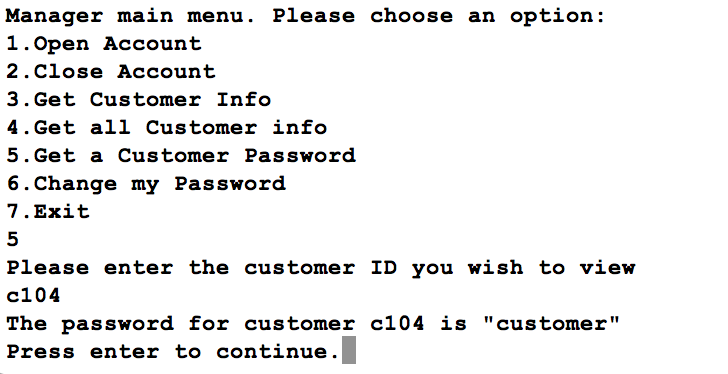
1. Opens an account



1. Sets the password of the new account and chooses account type
2. System sets up the account



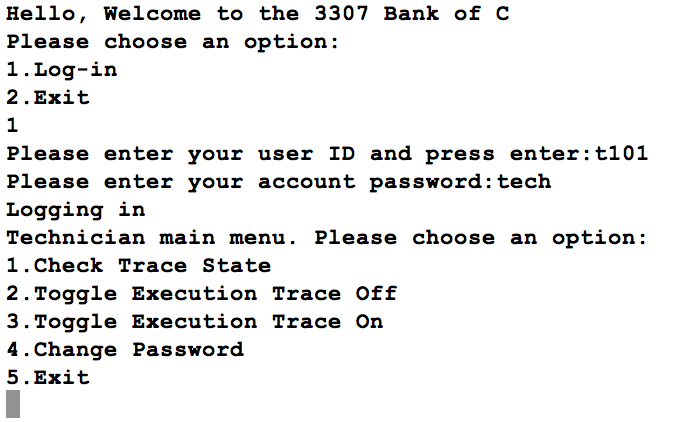
1. Manager checks the password for the account just opened



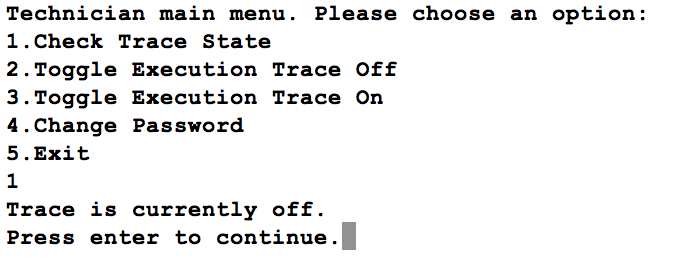
**Technician**

Checking/setting trace state

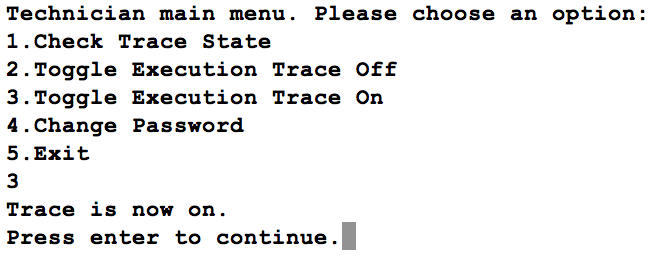
1. Technician logs in



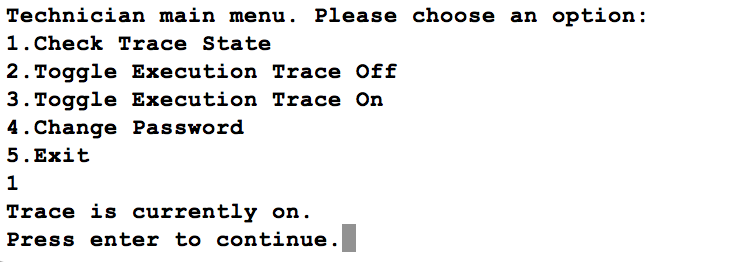
1. Checks the trace state



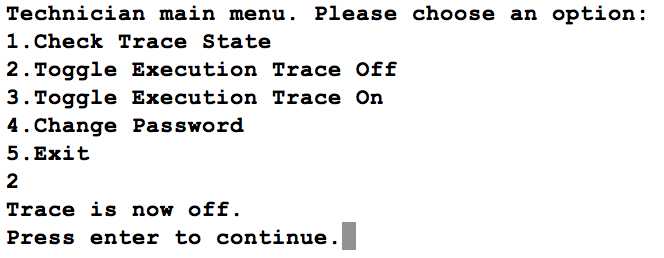
1. Turns the trace on



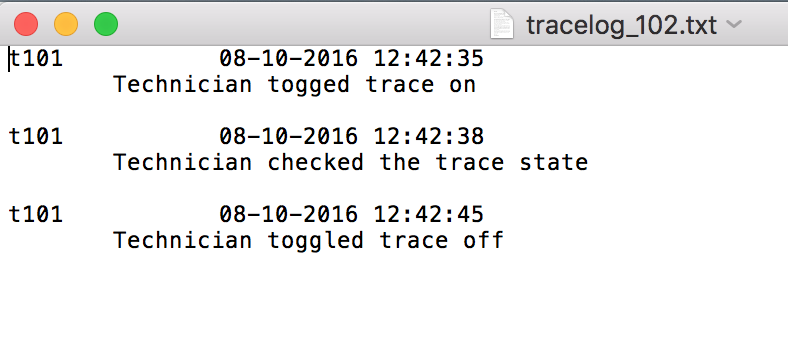
1. Checks the trace state again



1. Toggles trace state off



1. Checks the trace file to see if the trace occurred



5) Deliverable 5 – Implementation of the enhanced requirements: [5 max]

E1) The ability for a customer to E-transfer to another customer.

This is implemented mostly through the interface.cpp file. The user is prompted for all necessary information such as the customer to transfer to, the account to transfer from, and the amount to transfer. When all information is obtained, the system performs a few checks to make sure that both customers exist and that there are sufficient funds to perform the transfer. If all tests pass, the transfer is performed using a withdraw() and deposit() call.

E2) Each of the customers, the manager, and the technician have an ID and a password which they set and can change at any time

This is implemented mostly by each of the Customer, Manager and Technician objects and the private password variable that they each use. To change the passwords, each class has a changePassword() function which asks for the old and new passwords from the user.

E3) If a customer forgets their password, it can be retrieved by a manager

This is implemented by a function in the Manager class called retrievePass() which takes as an input, a customer ID and if the customer is a valid customer in memory, the password will be printed to the screen

E4) When a customer has been given a “profile” by a manager, the customer can open up to 20 total accounts on their profile without the manager

This is accomplished using a function in the Customer class called openNewAccount() which prompts the user for a name to the account. The account number is generated based on a private variable in the customer object that is incremented each time a new account is open. The accounts of a customer are held in a private array.

6) Deliverable 6: [3 max] – What I Have Learned

From this project, I have learned many useful things about C++, namely:

* Proper syntax
* File I/O
* Forming classes
* Linking files

In addition to learning C++, I have also learned about using good commenting and general coding style. As well as good time management skills for a significantly large individual coding project.