CS3307 – Object-Oriented Design & Analysis

"C++ Assignment: The Banking System"

Joshua Jackson, 2551, jjacks95@uwo.ca

October 2016

**Deliverable 1:**

Minimum set of requirements for:

**Customer** class defines the user role of customer or client. The minimum set of requirements for the Customer is to Deposit, Withdraw, and Transfer funds between their accounts. All requirements for customer are implemented and are demonstrable by the execution of the system. As well all users have a unique ID, categorizing their options when prompted, as to not let a Customer have Manager privileges.

Requirments:

1.1 Customer can Deposit into their Account
1.2 Customer can Withdraw from their Account
1.3 Customer can Transfer between their Accounts
1.4 Customer can get Balance of their Account

**Manager** class defines the user role of manager, or system admin. The minimum set of requirements for the manager is to Open/Create the User accounts. As well as, to display User Account Details. The Manager role is defined by the manager ID and Password, which allows the role of manager to create, close, view account details, and add complimentary account. All functions are implemented and demonstrable by the program.

Requirments:

1.5 Manager can Create Users, and their Accounts
1.6 Manager can Close Users Accounts
1.7 Manager can see User Details
1.8 Manager can add Complimentary Account

**Maintenance** class defines the user role of execution trace. Its only function is to turn on or off the execution trace. As well defined by a User ID and Password as to not allow other roles to turn the execution trace on or off.

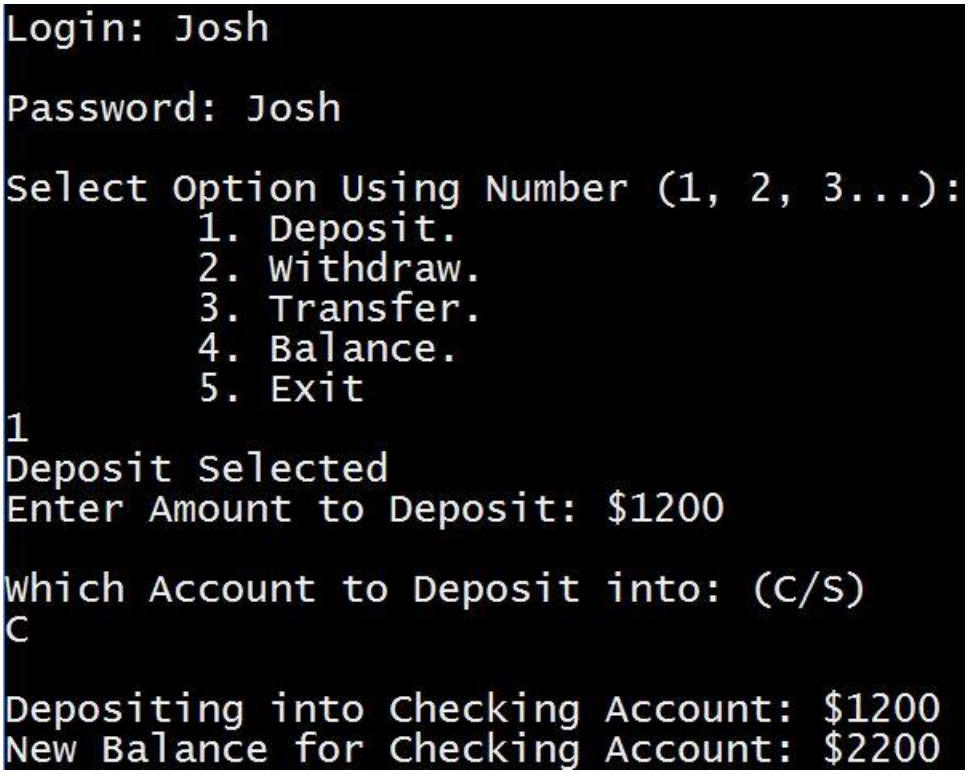
Requirments:

1.9 Maintenance can turn On or Off the execution trace

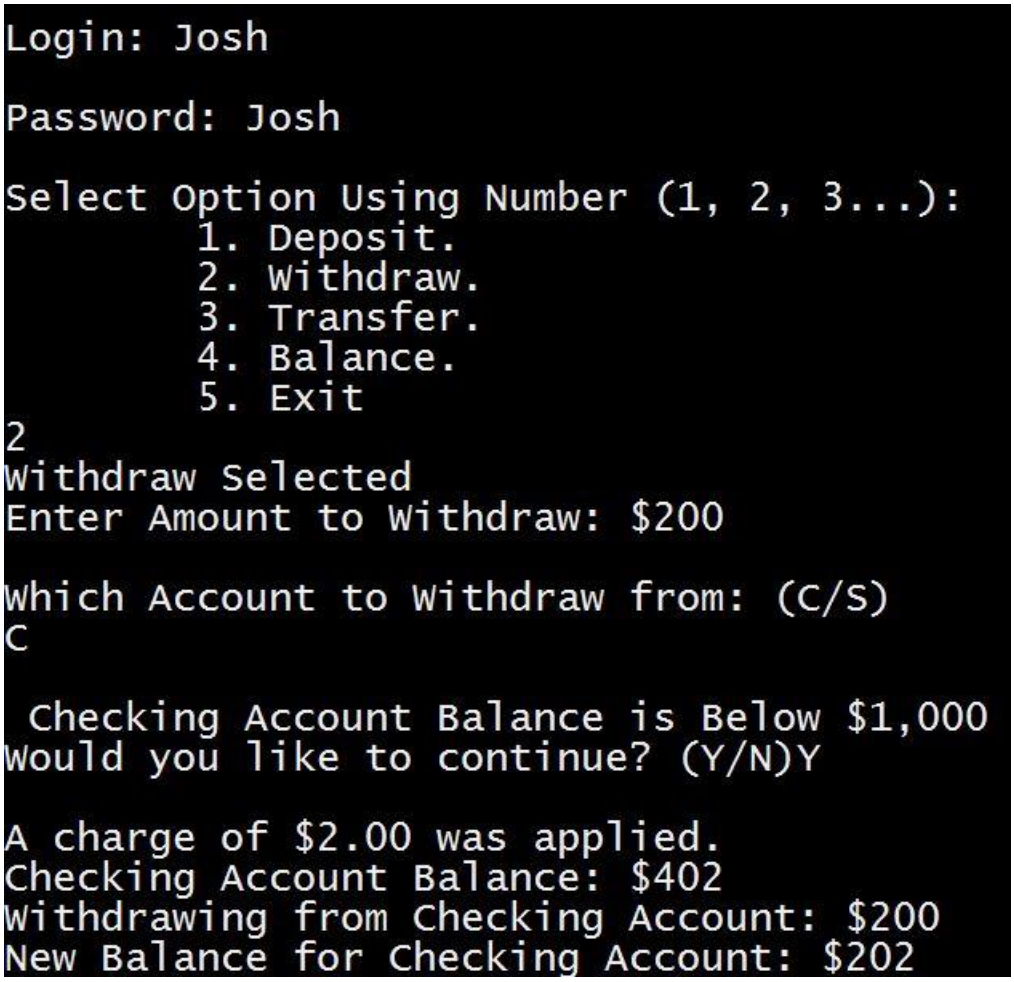
**Deliverable 4**

**Scenarios**

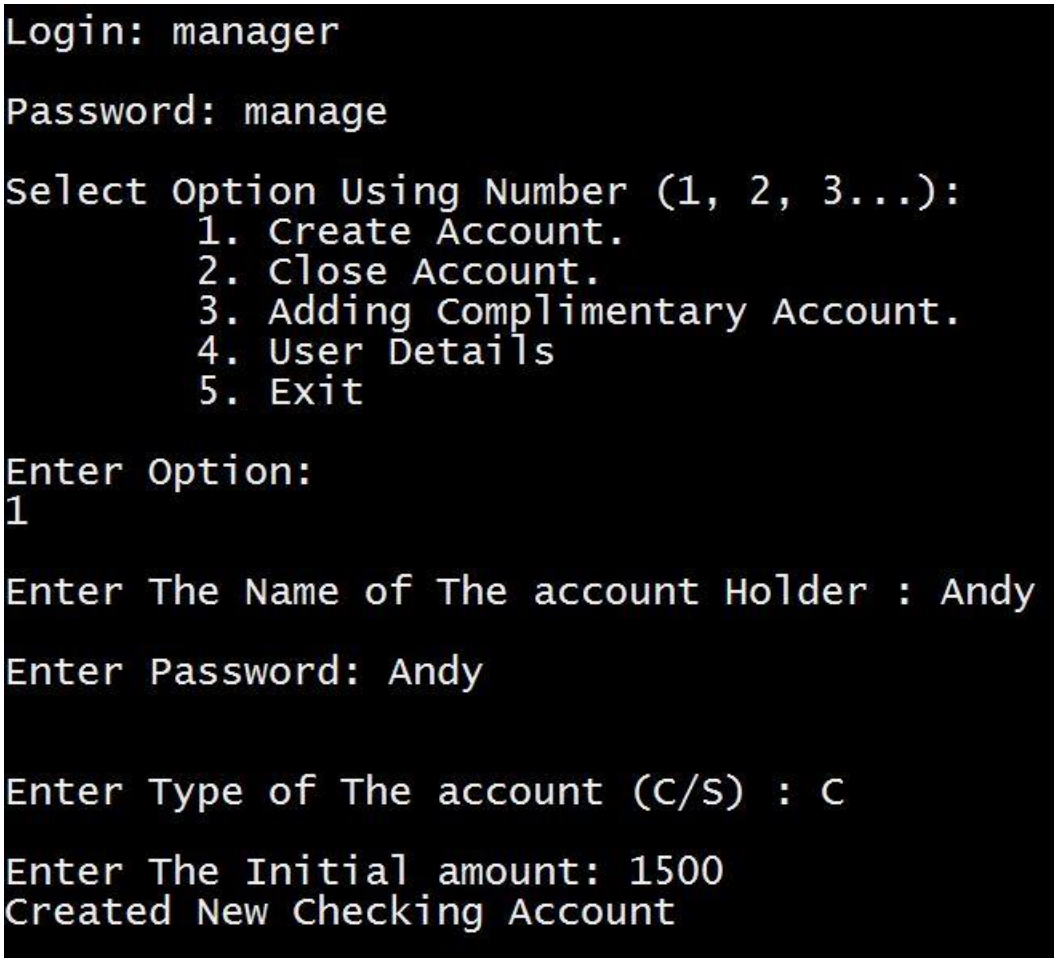
1. Depositing Funds
    - Client Logs into ATM
    - Selects Action of Depositing
    - Enters Amount to Deposit
    - Enters Account to Deposit either C or S
    - System Deposits Amount into Specified Account
    - System Displays Amount Deposited and New Account Balance



2. Withdrawing Funds
    - Client Logs into ATM
    - Selects Action of Withdrawing
    - Enters Amount to Withdraw
    - Enters Account to Withdraw from either C or S
    - Checking Account is going below $1000 System displays warning
    - User continues with Transaction
    - System Displays $2.00 charge
    - System Withdraws Amount from Specified Account
    - System Displays Amount Withdrawn and New Account Balance



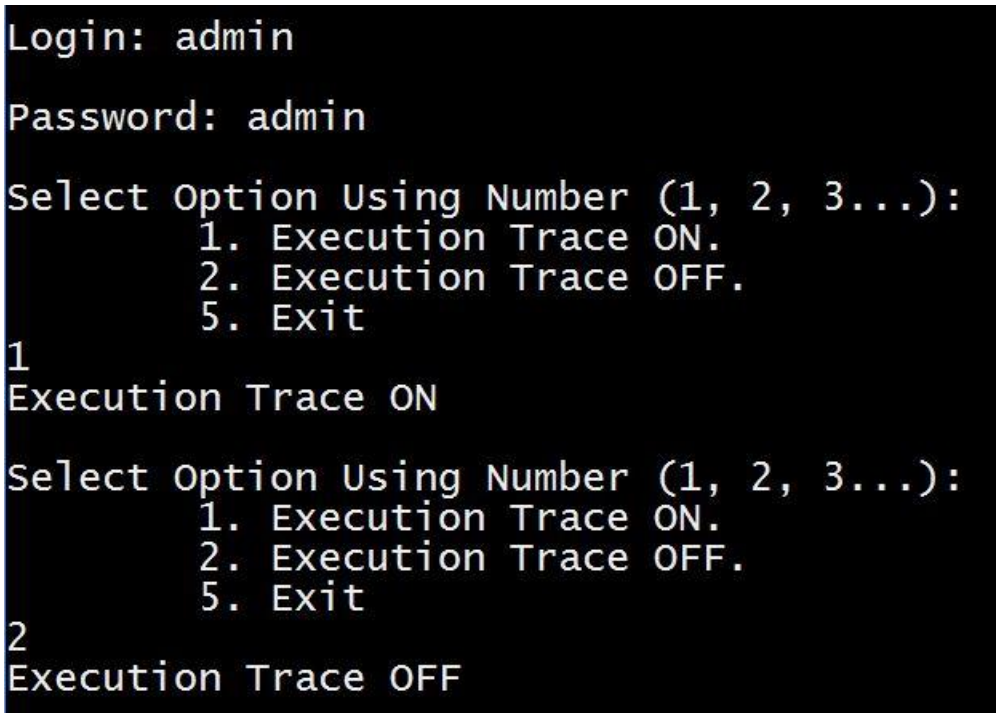
3. Creating Account
    - Manager logs into ATM
    - Manager Selects Create Account
    - System prompts manager for Username, Password, Account Type and Starting Balance
    - System Displays Message of Account Creation



4. User Details
    - Manager logs into ATM
    - Manager Selects User Details
    - System prompts, check if Manager wants to view all or one user account
    - Manager chooses all or individual users to see account details
    - Manager either enters S for individual or anything else for all
    - System Displays account Details

```
Login: manager

Password: manager

Select Option Using Number (1, 2, 3...):
        1. Create Account.
        2. Close Account.
        3. Adding Complimentary Account.
        4. User Details
        5. Exit

Enter Option:
4
For Sepcific User Enter S for All Enter Any Character
S
Enter Username: Josh

Account Details
User: Josh
Account Type: Savings
Account Balance: $1000

Account Type: Checking
Account Balance: $1000
```

5. Execution Trace
    - Maintenance logs into ATM
    - System Prompts Maintenance User to either Turn ON or OFF the execution trace
    - Maintenance Selects ON or OFF
    - System Displays Message of Outcome



**Deliverable 6**

Things I have learned about developing programs in C++:

- While developing the program I learned about C++ inheritance, it was a very useful tool that allowed for easy object oriented programming
- It is clear C++ is very scalable, much like Java it is easy to build implementation not specific to program requirements
- As well there is a lot of code, and a lot of need to redefine variables due to scope, this would cause maintainability nightmares
- As well, since I have never programmed in C++ deconstructors were introduced, and are a useful tool for deleting objects