[Airline Reservation Project Part 3]

Table of Contents

Statement of Independent Effort

I, [], hereby certify that is my original work completed without the assistance of anyone or any outside resources.

OR

I, [Imani Harrison], hereby certify that this is my original work completed with the assistance of [w3schools] listed in the reference. I used these resources in the following areas: [C++ Functions, C++ Operators, C++ Variables,C++ Break and Continue, C++ Conditions,C++ Do/While Loop, C++ Ouput,C++ User input, C++ If Else, C++ Strings]

Imani Harrison
[YOUR NAME]

Analysis of Specification

Process: Prompt user to input first and last name, then ask the user to input username and password, display error message if username and password is wrong, display ATM menu where it gives the user the option to Deposit, Withdraw, Balance Inquiry, Transfer Balance, and Log out next reading the money amounts for each transaction will only accept ints, meaning if you put a decimal in, it will read the first integer and then use the rest of the characters as input for whatever function is running. Display invalid message if not entered in whole dollars . Display error message if overdraft occurs, After a transaction is completed, the program will update the running balance and give the customer a detailed description of the transaction. End program

Input: username, password, first and last name, the letter D for Deposit, the letter W for Withdraw, the letter B for Balance Inquiry, the letter T for Transfer Balance, and the letter L for Logout.

Output: error message displayed if input a non-whole number for any of the options displayed in the ATM menu, another letter other than D,T,W,B,L, number of the transactions, error message displayed if overdraft occurs.

Pseudocode

Start → Include needed C++ Libraries → declare variables for first and last name, username and password, saving and checking, deposit, transaction, withdraw, balance inquiry, transfer and log out using integers, strings, and characters → allow user to login by putting their username and password in → Display Error Message if username and password are not correct → prompt user to choose from ATM menu

Select from Options (Deposit,Withdraw,Transfer,Balance Inquiry,Log out)

→ Perform Transactions based on menu choice →

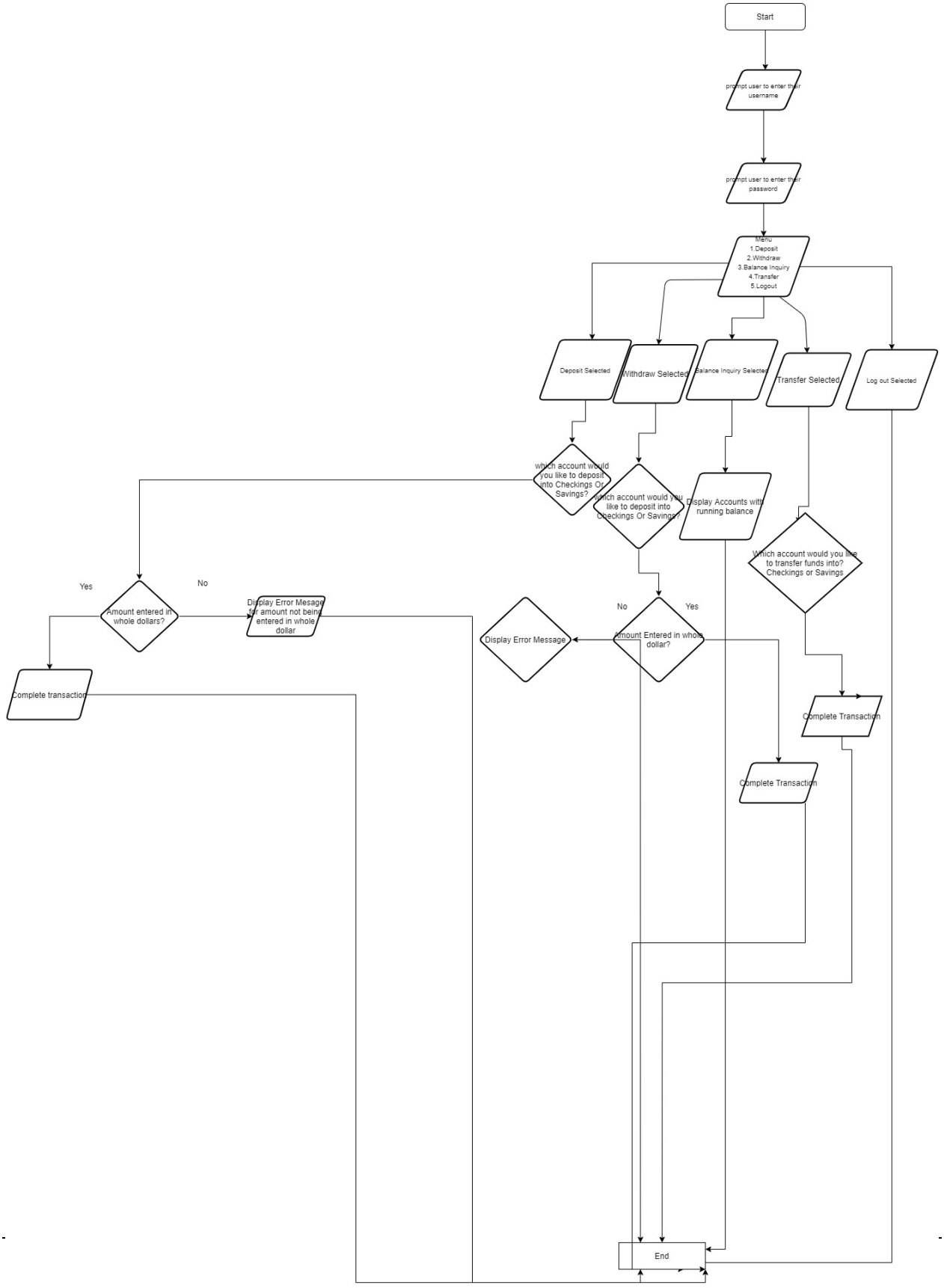
Based on menu choice display a different function to do different transactions →

Display error message if amount isn't entered In whole number

→ Declare loop and use if, else if, and else statements

→ Display number of Transactions → End Program

Flowchart


Start
prompt user to enter their username
prompt user to enter their password
Menu
1.Deposit
2.Withdraw
3.Balance Inquiry
4.Transfer
5.Logout
Deposit Selected
Withdraw Selected
Balance Inquiry Selected
Transfer Selected
Log out Selected
which account would you like to deposit into Checkings Or Savings?
which account would you like to deposit into Checkings Or Savings?
Display Accounts with running balance
Which account would you like to transfer funds into? Checkings or Savings
Yes
No
Amount entered in whole dollars?
Display Error Message for amount not being entered in whole dollar
Complete transaction
No
Yes
Display Error Message
Amount Entered in whole dollar?
Complete Transaction
Complete Transaction
End

Test Cases

1.
```
Please enter your password. red123

You have tried logging in 1 times.
You have 2 attempt(s) left.
Please try again.


Please enter your username. lwhite

Please enter your password. red678

You have tried logging in 2 times.
You have 1 attempt(s) left.
Please try again.


Please enter your username. lwhite

Please enter your password. red904

You have failed to log in too many times.
Terminating program.


...Program finished with exit code 1
Press ENTER to exit console.
```

2.

```
B
Balance Inquiry Selected...
|=======================================|
| Balance Inquiry for Robert Brown - rbrown
|=======================================|
| Balance Inquiry                       |
| Savings Account: $2500
| Checking Account: $35
|=======================================|


|=======================================|
|                ATM MENU               |
|=======================================|
| (D) Deposit                           |
| (W) Withdraw                          |
| (B) Balance Inquiry                   |
| (T) Transfer Balance                  |
| (L) Log Out                           |
|                                       |
| You have 2 transactions left.         |
|                                       |
|=======================================|
```

3.

```
verifying user...
Please enter your username. mblack

Please enter your password. green789

|=======================================
|                ATM MENU
|=======================================
| (D) Deposit
| (W) Withdraw
| (B) Balance Inquiry
| (T) Transfer Balance
| (L) Log Out
|
| You have 3 transactions left.
|
|=======================================

L
Log Out Selected...


...Program finished with exit code 0
Press ENTER to exit console.
```

4.

```
D
Deposit Selected...
What account would you like to deposit into? (Checking or Savings) Savings
How much would you like to deposit? $1.50
|=======================================|
|               ATM MENU                |
|=======================================|
| (D) Deposit                           |
| (W) Withdraw                          |
| (B) Balance Inquiry                   |
| (T) Transfer Balance                  |
| (L) Log Out                           |
|                                       |
| You have 2 transactions left.         |
|                                       |
|=======================================|

That is an invalid option. Please choice a valid choice.
```

5.

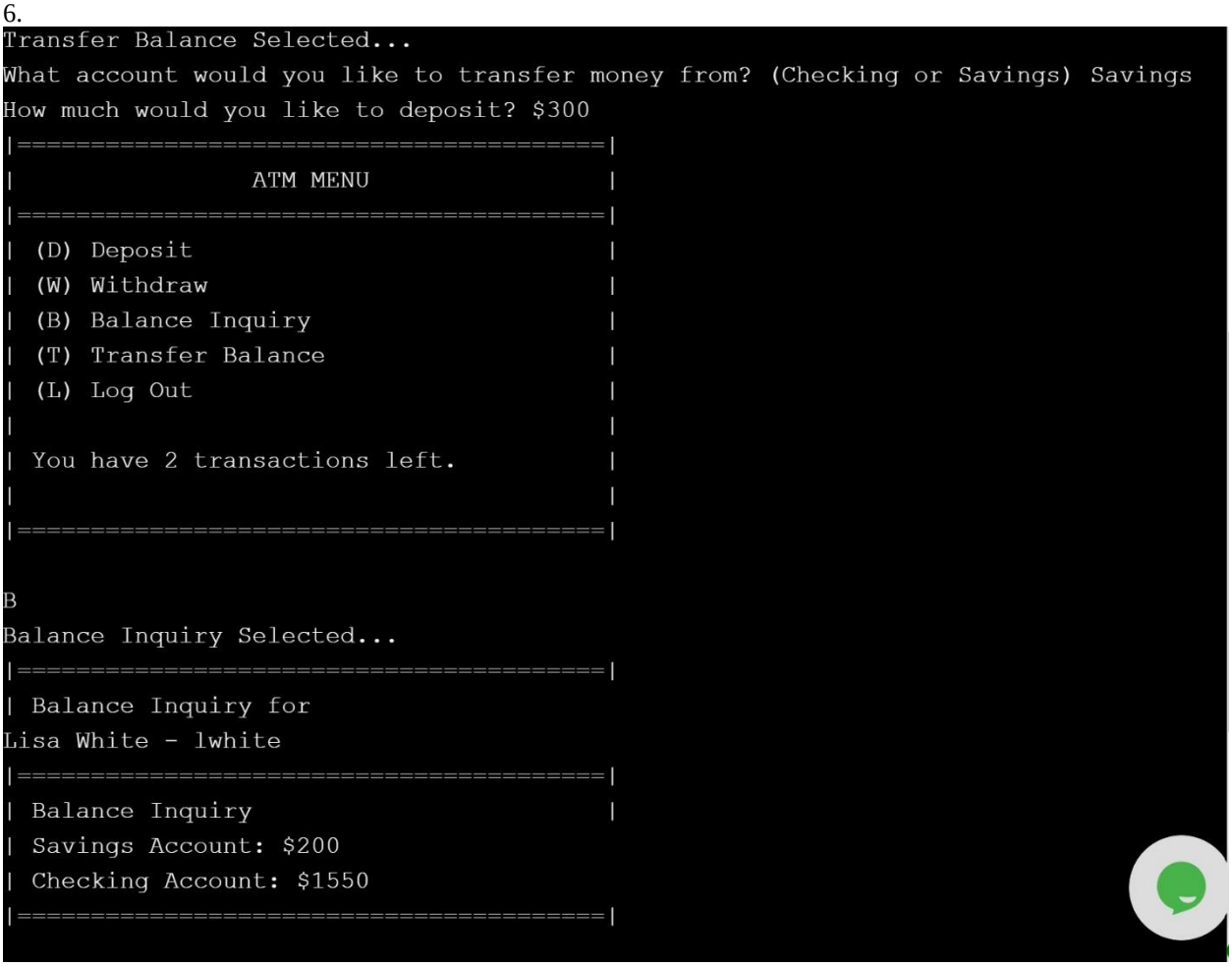
```
verifying user...
Please enter your username. lsmith

Please enter your password. yellow136

|=======================================|
|                ATM MENU               |
|=======================================|
| (D) Deposit                           |
| (W) Withdraw                          |
| (B) Balance Inquiry                   |
| (T) Transfer Balance                  |
| (L) Log Out                           |
|                                       |
| You have 3 transactions left.         |
|                                       |
|=======================================|

B
Balance Inquiry Selected...
|=======================================|
| Balance Inquiry for
Lee Ann Smith - lsmith
|=======================================|
| Balance Inquiry                       |
| Savings Account: $600
| Checking Account: $900
```

6.


Transfer Balance Selected...
What account would you like to transfer money from? (Checking or Savings) Savings
How much would you like to deposit? $300
ATM MENU
(D) Deposit
(W) Withdraw
(B) Balance Inquiry
(T) Transfer Balance
(L) Log Out
You have 2 transactions left.
B
Balance Inquiry Selected...
Balance Inquiry for
Lisa White - lwhite
Balance Inquiry
Savings Account: $200
Checking Account: $1550

7.
```
verifying user...
Please enter your username. lsmith

Please enter your password. yellow136

|=======================================|
|               ATM MENU                |
|=======================================|
| (D) Deposit                           |
| (W) Withdraw                          |
| (B) Balance Inquiry                   |
| (T) Transfer Balance                  |
| (L) Log Out                           |
|                                       |
| You have 3 transactions left.         |
|                                       |
|=======================================|

W
Withdraw Selected...
What account would you like to withdraw from? (Checking or Savings) Savi
ngs
How much would you like to withdraw? $300000
A withdraw of that size would result in an overdraft. Please try again

What account would you like to withdraw from? (Checking or Savings) Savi
ngs
```

8.
```
File failed to be read. Terminating program...

...Program finished with exit code 0
Press ENTER to exit console.
```

(Did not work because the correct file “accounts.dat” was not inputted so It could not be read)

9.

```
Withdraw Selected...
What account would you like to withdraw from? (Checking or Savings) Che
king
How much would you like to withdraw? $190
|========================================|
|                ATM MENU                |
|========================================|
| (D) Deposit                            |
| (W) Withdraw                           |
| (B) Balance Inquiry                    |
| (T) Transfer Balance                   |
| (L) Log Out                            |
|                                        |
| You have 1 transactions left.          |
|                                        |
|========================================|

B
Balance Inquiry Selected...
|========================================|
| Balance Inquiry for
Mark Black - mblack
|========================================|
| Balance Inquiry                        |
| Savings Account: $750
| Checking Account: $10
|========================================|
```

10.

```
What is your first and last name?
Robert Brown
Please enter your username.
rbrown
Please enter your password.
blue123
Which one would you like to deposit money into today, Checking or Savings?
Please choose A for savings or B for checking.
B
You have $35 in your savings account.
How much would you like to deposit?
100.35
Entry Error. It has to be a whole dollar amount.You have $35 in your savings account
.
How much would you like to deposit?
100.35
Entry Error. It has to be a whole dollar amount.You have $35 in your savings account
.
How much would you like to deposit?
100.35
Entry Error. It has to be a whole dollar amount.You have $35 in your savings account
.
How much would you like to deposit?
135
Amount acceptedYou had this many transactions: 4
You deposited:$ 135 in your account.
```

CODE

```
//Programmer:Imani Harrison

//Project name: Airline Reservation Part Three

//Date written: 11/8/2020

//Program description: A C++ program that allows the user to complete one of the 5 functions of the ATM: Deposit, Withdrawal, check current balance, transfer and log out.

#include <iostream>
#include <fstream>
#include <cmath>
#include <math.h>
#include <string>
#include <limits>
#include <ios>
#include <iomanip>
using namespace std;

// Variables - These are made global because we use them all over the program
// User Info
string fullName;
string username, password;

// Bank Accounts
/* These variables are used for the two bank account types.
    Since the ATM can only withdraw and deposit full dollars, all
    banking accounts will be using ints instead of floats.
    */
int savingsAcct = 0, checkingAcct = 0;

/* Function Prototypes -
  Declared for reference in main */
void atmMenuDisplay();
void atmMenuSelect(char menuChoice);
void depositCash();
void withdrawCash();
void balanceInquiry();
```

```
void transferMoney();
void getUserInfo(string& username, string& password);
void userVerify(ifstream& inF);


int main()
{
   ifstream inF;
   // ofstream outF;

   inF.open("accounts.dat");

   if (inF.fail())
   {
      cout << "File failed to be read. Terminating program...";
      return 0;
   }

   userVerify(inF);
   return 0;
}

// Basically the Login Screen
void getUserInfo(string& username, string& password)
{
   // Read User Info
   cout << "Please enter your username. ";
   cin >> username;
   cout << endl;
   cout << "Please enter your password. ";
   cin >> password;
   cout << endl;
}

// Checks to see if user exists
void userVerify(ifstream& inF)
{
   // Variables -
   // User Info

   cout << "verifying user..." << endl;

   // Verified?
```

```
bool isBankMember = false;
int loginAttempts = 0;

// Vars needed to verify
string bankName;
string bankUsername;
string bankPassword;
int bankSavings, bankCheckings;
char nihil; // reads end of line character

do
{
   getUserInfo(username, password);

   while (!inF.eof())
   {
      // Read User Info
      getline(inF, bankName, ',');
      getline(inF, bankUsername, ',');
      getline(inF, bankPassword, ',');
      inF >> bankSavings;
      inF.get(nihil);
      inF >> bankCheckings;
      inF.get(nihil);

      if ((username == bankUsername) && (password == bankPassword))
      {
         isBankMember = true;

         fullName = bankName;
         savingsAcct = bankSavings;
         checkingAcct = bankCheckings;
      }
   }

   // Return to beginning of the file
   inF.clear();
   inF.seekg(0);


   if (isBankMember == true)
   {
      // Display Menu and Start the Program
```

```
        atmMenuDisplay();
      }

      if (isBankMember == false)
      {
        loginAttempts++;
        if (loginAttempts < 3)
        {
          cout << "You have tried logging in " << loginAttempts << " times." << endl;
          cout << "You have " << (3 - loginAttempts) << " attempt(s) left." << endl;
          cout << "Please try again." << endl << endl << endl;
        }

        if (loginAttempts == 3)
        {
          loginAttempts++;
          cout << "You have failed to log in too many times." << endl;
          cout << "Terminating program." << endl;
          exit(1);
        }
      }
   } while (isBankMember == false && loginAttempts < 3);
}

// Display the ATM Menu
void atmMenuDisplay()
{
   int transactionCount = 0;
   char menuChoice;

   do {
      cout << "|=======================================|" << endl;
      cout << "|              ATM MENU                 |" << endl;
      cout << "|=======================================|" << endl;
      cout << "| (D) Deposit                           |" << endl;
      cout << "| (W) Withdraw                          |" << endl;
      cout << "| (B) Balance Inquiry                   |" << endl;
      cout << "| (T) Transfer Balance                  |" << endl;
      cout << "| (L) Log Out                           |" << endl;
      cout << "|                                       |" << endl;
      cout << "| You have " << (3 - transactionCount) << " transactions left.          |" << endl;
      cout << "|                                       |" << endl;
      cout << "|=======================================|" << endl << endl;
```

```
        cin >> menuChoice;
        // Perform Transactions based on menuChoice
        atmMenuSelect(menuChoice);
        transactionCount++;
    } while (menuChoice != 'L' && transactionCount < 3);  // do this until L is selected
}

// Select from Options and Perform the Transaction
void atmMenuSelect(char menuChoice)
{
    // Based on menuChoice, we go to a different function to do different transactions
    switch (menuChoice) {
        // Deposit Case
    case 'D': cout << "Deposit Selected..." << endl;
        depositCash();
        break;

        // Withdraw Case
    case 'W': cout << "Withdraw Selected..." << endl;
        withdrawCash();
        break;

        // Balance Inquiry Case
    case 'B': cout << "Balance Inquiry Selected..." << endl;
        balanceInquiry();
        break;

        // Transfer Case
    case 'T': cout << "Transfer Balance Selected..." << endl;
        transferMoney();
        break;

        // Log Out Case
    case 'L': cout << "Log Out Selected..." << endl;
        return;
        break;

        // Default
    default: cout << "That is an invalid option. Please choice a valid choice." << endl;
        break;
    }
}
```

```
// This function deposits cash into the selected account.
void depositCash()
{
   int depositAmount;
   string accountType;

   // Prompt for accountType and depositAmount
   cout << "What account would you like to deposit into? (Checking or Savings) ";
   cin >> accountType;

   // Check Account Tyoe
   if (accountType != "Checking" && accountType != "Savings")
   {
      cout << "This is an invalid account type. Please try again." << '\n' << '\n';
      depositCash();
   }
   else
   {
      cout << "How much would you like to deposit? $";
      cin >> depositAmount;

      // Check if depositAmount is valid
      if (cin.fail())
      {
         // Print error message then recurse into this function
         cout << "This is an invalid amount. Returning to main menu." << '\n' << '\n';
         return;
         // depositCash();
      }

      if (depositAmount < 0)
      {
         cout << "Please use positive numbers" << endl;
         depositCash();
      }

      if (accountType == "Checking")
      {
         checkingAcct += depositAmount;
      }

      if (accountType == "Savings")
```

```
        {
            savingsAcct += depositAmount;
        }
    }
}

void withdrawCash()
{
    int withdrawAmount;
    string accountType;

    cout << "What account would you like to withdraw from? (Checking or Savings) ";
    cin >> accountType;

    // Check Account Type
    if (accountType != "Checking" && accountType != "Savings")
    {
        cout << "This is an invalid account type. Please try again." << '\n' << '\n';
        withdrawCash();
    }

    else
    {
        cout << "How much would you like to withdraw? $";
        cin >> withdrawAmount;

        if (withdrawAmount < 0)
        {
            cout << "Please use positive numbers" << endl;
            withdrawCash();
        }

        // Check if withdrawAmount is valid
        if (cin.fail())
        {
            // Print error message then recurse into this function
            cout << "This is an invalid amount. Returning to main menu." << '\n' << '\n';
            return;

            // withdrawCash();
        }

        // If Checking
```

```
      if (accountType == "Checking")
      {
         if (withdrawAmount > checkingAcct)
         {
            cout << "A withdraw of that size would result in an overdraft. Please try again" << '\n'
<< '\n';
            withdrawCash();
         }
         else
            checkingAcct -= withdrawAmount;
      }

      // If Savings
      if (accountType == "Savings")
      {
         if (withdrawAmount > savingsAcct)
         {
            cout << "A withdraw of that size would result in an overdraft. Please try again" << '\n'
<< '\n';
            withdrawCash();
         }
         else
            savingsAcct -= withdrawAmount;
      }
   }
}

// This function displays both c=account balances.
void balanceInquiry()
{
   cout << "|=======================================|" << '\n';
   cout << "| Balance Inquiry for " << fullName << " - " << username << '\n';
   cout << "|=======================================|" << '\n';
   cout << "| Balance Inquiry                       |" << '\n';
   cout << "| Savings Account: $" << setw(6) << left << setprecision(2) << showpoint <<
savingsAcct << '\n';
   cout << "| Checking Account: $" << setw(6) << left << setprecision(2) << showpoint <<
checkingAcct << '\n';
   cout << "|=======================================|" << '\n' << '\n';

}

// This function transfers money from one account to the other.
```

```
void transferMoney()
{
   int transferAmount;
   string accountSource;     // This account is the source of the transfer.

   cout << "What account would you like to transfer money from? (Checking or Savings) ";
   cin >> accountSource;

   // Check Account Type
   if (accountSource != "Checking" && accountSource != "Savings")
   {
      cout << "Please input an account type (Checking or Savings)" << '\n';
      transferMoney();
   }

   cout << "How much would you like to deposit? $";
   cin >> transferAmount;

   if (transferAmount < 0)
   {
      cout << "Please use positive numbers" << endl;
      transferMoney();
   }

   if (cin.fail())
   {
      // Print error message then recurse into this function
      cout << "This is an invalid amount. Returning to main menu." << '\n' << '\n';
      return;

      // transferMoney();
   }
   else
   {
      if (accountSource == "Checking")
      {
         // Becareful of overdrafts...
         if (transferAmount > checkingAcct) {
            cout << "A transfer of that size would result in an overdraft. Please try again" << '\n';
            transferMoney();
         }
         else
         {
```

```
            checkingAcct -= transferAmount;
            savingsAcct += transferAmount;
         }
      }
      else if (accountSource == "Savings")
      {
         // Becareful of overdrafts...
         if (transferAmount > savingsAcct)
         {
            cout << "A transfer of that size would result in an overdraft. Please try again" << '\n';
            withdrawCash();
         }
         else
         {
            savingsAcct -= transferAmount;
            checkingAcct += transferAmount;
         }
      }
   }
}
```

Work Cited

1. Data, Refsnes. *C++ Tutorial*, 2020, www.w3schools.com/cpp/default.asp.

Grade Sheet

Fundamentals of Programming

Ms. Vanessa Coote

Before submitting the project package, the student should review each of the elements listed below and put a checkmark only in those checkboxes where the designated elements has been reviewed and meets specifications. After completing your document package, number your pages and write the designated page numbers onto the spaces provided on the grading sheet.

_____ Professionalism (10 points)

- ☐ Following directions
- ☐ Neatly assembled 8 ½ by 11
- ☐ Cover page
- ☐ Page numbers
- ☐ Documentation

_____ Source Program Listing and Proper Execution of Program (30 points)

It is expected that each student's program will run correctly

- ☐ Program source code listing matches code on submission and/or backups
- ☐ Inclusion of comment lines in source code
  - ~ Comments at the beginning of the program including programmer, project name and number, date written, and brief program description.
  - ~ Comments at key locations throughout the code
- ☐ Descriptive variable names (that follow naming convention)
- ☐ Logic is correct
- ☐ Logic is clear and easy to follow
- ☐ Proper formatting of statements
- ☐ Alignment, proper indentation, etc
- ☐ Proper use of data types and data conversions

_____ Test Data (5 points)

- ☐ Each test case properly calculated by hand and documented
- ☐ Suitable choice of you own test data case

_____ Input Window (10 points)

- ☐ Correct data type for each input section
- ☐ Analysis of data type (e.g. int, float, double etc.)
- ☐ Appropriate restrictions for each input section
- ☐ Data input value shown matches specified test data
- ☐ Appropriate display for each input section

_____ Output (15 points)

- ☐ Suitable layout of output (including required fields, easy to read layout, etc.)
- ☐ All data cases displayed
- ☐ Correct value displayed for each case
- ☐ Correct format of fields (e.g. use of integers and not float as appropriate, dollars and cents, display of $, etc)
- ☐ Required output format
- ☐ Aesthetics (User-friendliness, easy to understand output, alignments, etc)

_____ Documentation (40 points)

- ☐ Analysis of specifications
- ☐ Pseudocode
- ☐ Flowchart
- ☐ Hard copy of program

_____ Fully Functioning Program (30 points)

_____

Possible points = 140

Points Earned =