ATM

LAB # 6

By

Corey Henry, Samuel Howle and Michael Bishop

***“On my honor, as a Mississippi State University student, I have neither***

***given nor received unauthorized assistance on this academic work.”***

Signatures:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CSE-1284-06-201430 Intro to Computer Programming

Class Section # 6

Josh Crowson

10/28/2014

**Analysis and Conclusions**

We ended up doing the bonus part of the code but we started we just starting with a one-person pin code. We first started off creating functions for everything we thought we were going to need before we made our main function. We toyed with the idea of global versus local variables for a while because we were having problems getting the information in the functions if the variable wasn’t global.

The main function was very easy to create we just had to be able to call all the functions and connect them. We created an extra function just for the options that you had to make so we would be able to link all of the functions inside other functions. One of the largest problems we ran into was that the amounts would increase and decrease for one time, but we couldn’t get the amounts to stay current and they would return to the starting amounts each time we ended that round. After much debugging and analysis we discovered that we were resetting the values and not calling the options in the correct way. After we changed the options to be called correctly it fixed this problem. Doing the bonus was very easy, instead of setting only one amount and calling the user, we assigned two users, with different pins, bank account information and names. Then all we had to do was call the user depending on the pin number you entered and the rest of the program was already built, adding this user in only took 5 minutes to complete once we had the program worked out. While working with a partner was helpful in terms of having someone to bounce ideas off of, having 3 made us continually lose our train of thought and mess with our groove because we kept switching between people. It was a good exercise because it helped us get very familiar with functions and work with different styles of coding because each of us had a different approach to the same goal.

**Source Code:**

Corey Henry & Samuel Howle & Michael Bishop #Date Assigned: 015Oct2014

# #

#Course CSE 1284 Sec 02 #Date Due: 29Oct2014

#File name: lab6.py

#

#Program description- Create a Atm prompt user screen

#global variable for checking and savings ammounts

# inquiry function

def inquiry(checking,savings):

print('Checking account balance: ', checking)

print('Savings account balance: ', savings)

stop(checking,savings)

#deposit to checking function

def deposit\_check(checking,savings):

cash = float(input('Ammount to be deposited: '))

checking += cash

print('Transaction Complete')

inquiry(checking,savings)

return (checking)

stop(checking,savings)

#deposit to savings function

def deposit\_save(checking,savings):

cash = float(input('Ammount to be deposited: '))

savings += cash

print('Transaction Complete')

inquiry(checking, savings)

stop(checking,savings)

return savings

#Withdrawal from checking

def withdrawal\_c(checking,savings):

cash = float(input('Amount to be withdrawn: '))

while cash > checking:

print('Your account has insufficient funds.')

print('Transaction cannot be completed.')

cash = float(input('Amount to be withdrawn.'))

print('Transaction is complete')

checking -= cash

inquiry(checking,savings)

stop(checking,savings)

return checking

#Withdrawal from savings

def withdrawal\_s(checking,savings):

cash = float(input('Amount to be withdrawn: '))

while cash > savings:

print('Your account has insufficient funds.')

print('Transaction cannot be completed.')

cash = float(input('Amount to be withdrawn.'))

print('Transaction is complete')

savings -= cash

inquiry(checking,savings)

stop(checking,savings)

return savings

#quit function

def stop(checking,savings):

cont = input('Would you like to make another transaction? (Y/N): ')

if cont == ('n'):

print('\n')

end\_prog(checking,savings)

elif cont == ('y'):

options(checking,savings)

return

#create end fuction

def end\_prog(checking,savings):

print('Checking account balance: ', checking)

print('Savings account balance: ', savings)

print('Thank you, Come again.')

#options function

def options(checking,savings):

print('1. Account Inquiry')

print('2. Deposit to Checking')

print('3. Deposit to Savings')

print('4. Withdraw from Checking')

print('5. Withdraw from Savings')

print('6. Quit')

print('\n')

option = int(input('Please select an option: '))

#create validation loop

while option > 6 or option < 1:

print('Error: please select a number 1 through 6')

option = int(input('Please select an option: '))

if option == 1:

inquiry(checking,savings)

elif option == 2:

checking = deposit\_check(checking,savings)

elif option == 3:

savings = deposit\_save(checking,savings)

elif option == 4:

checking =withdrawal\_c(checking,savings)

elif option == 5:

savings = withdrawal\_s(checking,savings)

elif option == 6:

end\_prog(checking,savings)

# create main function

def main():

#user2 = ('Diddy Kong')

checking = 1500

savings = 37856

cont = ('y')

user1 = ('Lion King')

while cont == ('y'):

pin\_number = int(input('Welcome! Please enter your 4 digit PIN: '))

#if pin\_number == 1234:

#user = user2

#checking = 1000

#savings = 23000

#if pin\_number == 4321:

#checking = 1500

#savings = 37856

#validation loop

while pin\_number != 4321:

print('Sorry! Please try again.')

pin\_number = int(input('Welcome! Please enter your 4 digit PIN: '))

print('Welcome back, ', user1)

options(checking,savings)

#call main function

main()