ITCS 1212 – Fall 2015
Programming Assignment 2

Assigned: Thursday, November 5th, 2015
Due: Monday, November 23 by 11:55pm (submitted electronically via your moodle lab page)
NO LATE SUBMISSIONS WILL BE ACCEPTED

NOTE: Be sure to adhere to the University's Policy on Academic Integrity as discussed in class. Programming assignments are to be written individually and submitted programs must be the result of your own efforts. Any suspicion of cheating will be dealt with accordingly. The minimum punishment I will impose for Academic integrity violation will be a 0% on the programming assignment and a 2-letter grade reduction in the course. Please refer to the following site for additional details, as it is your responsibility to be aware:
http://www.legal.uncc.edu/policies/ps-105.html
Important: I reserve the right to revisit program grades after completion of semester and final exams. If there is a marked discrepancy in program grades and your answers to related questions on tests, this will be challenged and appropriate grade adjustments given. In addition, appropriate measures will be taken with respect to the University's policy on Academic Integrity where indicated as a result of the outcome of such a challenge.

Purpose: This assignment is designed to demonstrate your understanding of the creation, compilation, and program submission process we will be using this semester based on the contents of the lecture and textbook.

Assignment Description:

In this assignment you will create a program that allows a user to choose one of the following main menu items:

C) Create a bank account by supplying a user id and password.
L) Login using their id and password.
Q) Quit the program.

If User enters an option other than (uppercase or lowercase) C, L, or Q, the program does not do anything and shows a message wrong option and displays the menu again.

When user enters option **C**, the program asks the user to enter a user ID and a password and successfully returns back to the main menu. The login and password should be stored in a file.

If the user chooses option **L**, the program will display the login prompt and then, it will ask for the password. At this stage, the entered login and password are compared to the stored login/password pair and proper message will show up if the

match was not found which takes the user back to the main menu.
Now, if login was successful, the following banking menu will be display to let the user choose one of the following tasks:

W) Withdraw money.
D) Deposit money.
B) Request balance.
Q) Quit the program.

(again, remember that if the user enters an option other than (uppercase or lowercase) **W, D, B, Q,** the program should show a message and display the menu again.)

The initial balance for the user account should be $0.00
If the user chooses option **W**, the program should ask the user to enter amount user wishes to withdraw.
Validation: In the case of withdraw, if the amount is more than balance the user should be notified and no withdraw will occur.

If the user chooses option **D**, the program should ask the user how much amount the user wishes to deposit and add it to initial balance.

If the user chooses option B, the program should display the balance amount in the user account.

Sample Output:

```
Hi! Welcome to Future Computer Programmer ATM Machine! Please select
an option from the menu below:
l -> Login
c -> Create New Account
q -> Quit

Enter your choice: L

Please enter your user id: 12
Please enter your password 2345

No match was found! Login Failed !

Please select an option from the menu below:
l -> Login
c -> Create New Account
q -> Quit

Enter your choice: c
```

```
Please enter your user name: 12
Please enter your password: 2345

Thank You! Your account has been created!

Please select an option from the menu below:
l -> Login
c -> Create New Account
q -> Quit

Enter your choice: l

Please enter your user id: 12
Please enter your password: 2345

Access Granted!

Please select an option from the menu below

d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
q -> QuitEnter your choice: d

Enter amount of deposit: $20
$20 was deposited.

Please select an option from the menu below
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
q -> Quit

Enter your choice: R
Your balance is $20.

Please select an option from the menu below
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
q -> Quit

Enter your choice: W
Enter amount of withdrawal: $25
Sorry withdrawal amount exceeds the balance. Can’t withdraw !
```

```
Please select an option from the menu below

d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
q -> Quit

Enter your choice: W

Enter amount of withdrawal: $2.5

Please select an option from the menu below
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
q -> Quit

Enter your choice: r
Your balance is $17.5.

Please select an option from the menu below
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
q -> QuitEnter your choice: Q

Thanks for stopping by!
```

Additional Instructions: Please make sure your code has following functions:
1. mainMenu(): Function to display the main menu for login, create account, and quit.
2. createAccount(): Function that accepts user ID and password to create the account.
3. login(): Function to match user ID and password to indicate if login is successful or not.
4. bankingMenu(): Function to display the menu choices for Deposit, Withdrawal, Balance and Quit.
5. deposit (): Function to accept the amount user wishes to deposit in his/her account.
6. withdraw(): Function to accept the amount user wishes to withdraw from the account.
7. displayBalance(): Function to display the balance amount in the user account.
You can use additional functions (optional) for input validation and withdrawal amount validation. Your program should make use appropriate switch and if-else statements where necessary.

- You will need a file to hold the login/password and another file to store the transaction information similar to the information shown on the screen.
- Make sure you work out an algorithm and submit it as well either in the form of pseudo-code or flowchart before you start implementing the code. You can include your pseudo code within the block comment/header documentation part of the code. A pseudo code/algorithm should consist of clear and concise steps that mirror your implementation logic.

A grading rubric has been provided to you as a reference. Few important things to remember:

1. Start working early. Do not wait till the due date to start working on your assignment.
2. Always store a backup copy of your code online.
3. Make sure you submit a .cpp file. A .cbp file will result in grade – 0.
4. Ask for timely help. You have various resources available (Teaching Faculty, TA's, CCI Tutors), make sure you ask for help if you are struggling with something.
5. Submit your own work!

Grading Rubric:

Program compiles Pseudo code/Algorithm (Clear step-by-step instructions)	20 Points 10 Points
Documentation Header documentation (Author, Date, Description) Internal documentation (process doc, function doc)	5 Points 10 Points
Correct Input gathering	10 Points
Correct calculations / functions Appropriate use of functions and function calls	20 Points 5 Points
Correct use of switch and if-else statements	5 Points
Well-formatted correct output (screen and file)	15 Points

Total: 100 Points

How to submit your assignment:

1. Rename your file with a meaningful name. Do not keep it main.cpp
2. Upload your file on your lab moodle page before the due date. There will be a submission link on your moodle page.
3. Make sure you upload the correct filename.cpp file. Don not upload the project file (.cbj) as this will result in no grade for your assignment.