COMPUTER SCIENCES AND SOFTWARE ENGINEERING
AUBURN UNVERSITY

COMP 2710
Software Construction

Fall 2013

Lab 1
TigerBook Messaging System

Due: October 2, 2013

Points Possible: 100 (25 Design, 75 code)
Due:
Design Portion: **Wednesday, September 25th, 2013 by 11:55 pm** turned in via CANVAS SUBMISSION
Programming Portion: **Wednesday, October 2nd, 2013 by 11:55 pm** turned in via CANVAS SUBMISSION

Goals:
To develop some proficiency with basic C++ syntax and semantics.
To organize code into classes.
To gain some familiarity with formal testing.
To write a fun application!

Design Portion:
Create a text or PDF file named "<username>_1w.txt" (for example, mine would read "lim_1w.txt"). Please submit a text (.txt) file or PDF (.pdf), *do not submit MS Word Documents or other proprietary formats.*

Process (steps 1-3 due with initial turn-in, steps 4-5 due with final turn-in:
Each of the following should address the program specification below. Concern yourself more with thinking about the problem than worrying about the specifics of implementation at this stage. For the first turn-in, you will provide steps 1-3. For the final (electronic) turn-in, you will provide all the steps (including any revisions you may have made). Please note revisions in your final copy! You should provide:

1. **Analysis (10):** Write a few important **use cases**. Remember, these describe how a user interacts with the system (what they do, what the systems does in response, etc.). These need not be long, but they should have enough basic details such that someone unfamiliar with the system can have an understanding of what is happening. They should not include internal technical details that the user is not (and should not) be aware of. Make sure that any special rules/features you plan to add are clearly described in your Analysis section.

2. **Design (10)**: From your use cases, identify likely candidates for software objects, the processes related to the objects and the data stores. List the classes (or draw a Class Diagram, if you like) and include:

1) The name and purpose of the class
2) The member variables and the functions of the class
3) Any other classes that this class depends on or uses
4) Any relevant notes that don't fit into the previous categories

You must also describe the system using data flow diagrams.

3. **Testing (5)**: Develop lists of specific test cases:

1) For the system at large. In other words, describe inputs for "nominal" usage. You may need several scenarios. Also, suggest scenarios for abnormal usage and show what your program should do (for example, entering a negative number for a menu might ask the user to try again).
2) For each object. (Later, these tests can be automated easily using a simple driver function in the object).

Remember, test cases are specific, repeatable, and should have a clearly defined result.

4. Implement your solution in C++ (70).

5. **Test Results (5)**: ***After developing your solution***, actually try all of your test cases (both system and unit testing). Actually show the results of your testing (a copy and paste from your program output is fine – don't stress too much about formatting as long as the results are clear). You should have test results for every test case you described. If your system doesn't behave as intended, you should note this.

Program Portion:

One of the most commonly used application in the smart phones, laptops and desktops is the facebook social network for sharing information, messaging or texting application. In this Lab 1, you will analyze, design, implement and test a simple message sharing application, called TigerBook. It will have a simple text-based user interface that allows multiple users to share messages if they are in the same friend group. In this lab project all the users and the message buffer must be implemented in the same address space, i.e. the message buffer memory space can be accessible by all the users, although your program should allow only users in the same friend group to display the messages in a user home page.

What is TigerBook?

The concept of TigerBook is similar to facebook that is widely used for social networking and sharing of information among friends. The exception is that for the purpose of this class assignment, we will simplify it to deal with only text messages; so no images or videos will be posted. For this assignment's purposes, TigerBook is a program (or collection of programs) that does the following:

- There are many users that use TigerBook for posting their own messages and sharing them with their friends. Each user posts their message in their own **wall page**. In this assignment your implementation must handle **at least four users**. *The program must allow multiple lines of a message to be posted by each user.*

- Each user can also tweet a message which will be seen by all users. *The program must allow multiple lines of a message to be tweeted by each user.* Messages tweeted by any user will be stored in the same message buffer as other messages..
- Messages are shared among users in different **friend groups**, identified by a **friend list** associated with each user. The user must be allowed to enter any name in the friend list only if the name is a valid user. Friends are not mutual, e.g. `Joe` is a friend of `Jane`, but `Jane` is not a friend of `Joe`.
- Each user has a home page. The **home page** of a user displays all the messages of all their friends, all of their own messages and *all tweeted message from all users*, in reverse chronological order in which they are created, i.e. the most recent messages are displayed before the less recent messages.
- In the home or wall pages, the program will first display only the two most recent messages from each friend and himself, and prompt if the user wants to display more messages. If the response is "yes", then all messages from themselves and all their friends and will be displayed, otherwise, the program will stop the display.
- Since this assignments implements a simple interface, TigerBook will allow only one user to be logged in at a time; so it will allow for change of user. When a user is logged in, all operations will be for the current user.
- Each message must contain the name of the user and the text message.

Message format

For this assignment, the messages and user names must be stored in the following format, similar to the format used in Homework 1. There must be only *one* message buffer to store all messages from all the users, placed in a *reverse* chronological order from which they are created. Each message must contain the user's name and their message whereby the user name is enclosed by the strings `"{["` and `"]}"` followed by the user's message. Different lines of a message must be separated by a '`\n`' character. For examples, if `John` first entered the message `"Welcome to the group!"` and later, when `Jane` entered `"Glad I'm in the group!"` then the message buffer must contain `"{[Jane]}Glad I'm in the group!{[John]}Welcome to the group!"`. Each message must be of variable length. The purpose for this standard message format is that everyone's wall message page should end up being compatible. That is, every user should be able to work with every friend's wall messages.

Tweeted messages must also be stored in the same message buffer, except that the name of the users is appended with the string "::tweet". For example, if Bill tweets a message "`Greetings, everybody!"`, then when the message is stored in the message buffer, the content of the buffer will be as follows: `"{[Bill::tweet]}Greetings, everybody!{[Jane]}Glad I'm in the group!{[John]}Welcome to the group!"`.

Your program must meet this requirement for message format, otherwise significant points will be deducted.

The user interface

Write a menu-based and text-based user interface for interacting with the TigerBook system. The main menu has (at least) 8 options to choose from as shown below. *You can abbreviate the menu so that all the 8 options can be shown in only one or two lines.* The user interface will accept the option and all related inputs from the user, perform the operation and then repeatedly shows this menu of options again and prompts for the next option.

- **Create a new user:** When selected, the program will then prompt for the name of the user and insert the name in one of the user objects. When a user is created, that user becomes the *current user*. The program must prompt a welcome banner, e.g. saying "Welcome to TigerBook, Joe!", which should be centered on the screen and surrounded by a box.
- **Post a message**: When selected, the program will prompt for the message and post message on the current user's wall page. *Your program must allow multiple lines of messages. To end a message, the user will enter a new line with a string " ! ! "followed by the enter key. The final string " ! ! "must not be stored in the message buffer, neither should it be displayed in the wall or home pages.*
- **Tweet a message**: When selected, the program will prompt for the message and stores the message in the message buffer. As in posted messages, *your program must allow multiple lines of messages. To end a message, the user will enter a new line with a string " ! ! "followed by the enter key.*
- **Display wall page**: When selected, the program will first display a title indicating that it is displaying the current user's wall page, e.g. "Joe's Wall Page". It then displays the *two* latest messages in the current user's wall page or *the user's tweeted messages*, in reverse chronological order. Since the wall page contains only messages from the current user, *the wall page must NOT display the user names*; instead it only displays the user's messages in reverse chronological order. Messages from different posting must be separated by a blank line. After displaying the two latest messages, it will then prompt the user if they want more messages. If the response is "no", then it will stop displaying messages, but if the response is "yes", it will display all the remaining messages from that user. If there are two or fewer messages then the program must not prompt for more messages.
- **Display home page**: When selected, it first displays the current user home page title, e.g. "Joe's Home Page", and then it displays only the two latest messages either from the current user's wall page or the wall pages of all the *friends* of the current user (and friends only) or tweeted messages, whichever are the two latest messages. When displaying the message from each user, it must display the user name following by a ':' and a '\n' and then followed by the message. Messages from different users must be separate by a blank line. Tweeted messages must not display the string "::tweet", only the user name and the message in the same format as posted messages above. It will then prompt the user if they want more. If the response is "no", then it will stop displaying messages, but if the response is "yes", it will display all the remaining messages from that user and or the wall pages of all the friends of the current user. If there are two or fewer messages then the program must not prompt for more messages. In both the cases above, the

messages are all displayed in reverse chronological order, i.e. the most recently posted messages will be displayed before the earlier messages.

- **Add a friend**: When selected, the program will then prompt for the name of the friend. It then checks if the friend's name is a currently a valid user. If so, it adds the friend in the current user friend list. If not, it will display an error message. Friends are not mutual, i.e. User X is a friend of User Y but Y may not be a friend of X.
- **Switch to a different user**: When selected, the program will prompt for the user name, e.g. Jane, and then checks if the name is a valid user, i.e. has been created. If so, it switches current user to Jane. If it is not a valid user, it displays the error message and repeatedly asks for the user name. (Your program must not exit or terminate when this naming error occur.) When switched to a different user, the program must prompt a welcome back banner, e.g. saying "Welcome back to TigerBook, Jane", which should be centered on the screen and surrounded by a box.
- **Quit the TigerBook application:** When selected, the program will exit gracefully.

The user interface must check for correct input value from the users. If there is any error, e.g. invalid user name, then the program must display the error and continue to prompt for the correct input. You program must not exit or terminate when there is an incorrect input value.

The name of your program must be called <username>_1.cpp (for example, mine would read "lim_1.cpp"

Use comments to provide a heading at the top of your code containing your name, Auburn Userid, and filename. You will lose points if you do not use the specific program file name, or do not have a comment block on **EVERY** program you hand in.

Your program's output need not exactly match the style of the sample output (see the end of this file for one example of sample output).

Important Notes:
You must use an object-oriented programming strategy in order to design and implement this TigerBook application (in other words, you will need write class definitions and use those classes, you can't just throw everything in main()). A well-done implementation will produce a number of robust classes, many of which may be useful for future programs in this course and beyond. Remember good design practices discussed in class:

a) A class should do one thing, and do it well
b) Classes should NOT be highly coupled
c) Classes should be highly cohesive

Some potential classes:

1. A ***Menu*** class which handles basic user screw-ups (choosing an option out of bounds).
2. A ***System*** class which instantiates the other objects that must be initialized.
3. A ***User*** class which maintains information on the user name, friend list and wall page.
4. A ***FriendList*** class that maintains the list of friends of a user.
5. A ***WallPage*** class that allows messages to be posted (stored) on the wall page of a user.
6. A ***HomePage*** class that allows messages from the user and their friends to be displayed in reverse chronological order.

You should at a very minimum have classes to handle menus, users and wall pages.
Use your judgement to define additional classes or not use some of the classes above.

- You should follow standard commenting guidelines. Follow the C++ programming style guideline that was handed out.

- You DO NOT need any graphical user interface for this simple, text-based application. If you want to implement a visualization of some sort, then that is extra credit.

Error-Checking:

You should provide enough error-checking that a moderately informed user will not crash your program. This should be discovered through your unit-testing. Your prompts should still inform the user of what is expected of them, even if you have error-checking in place.

Please submit your program through the Canvas system online. If for some disastrous reason Canvas goes down, instead e-mail your submission to TA – Seungbae Lee – at szl0039@tigermail.auburn.edu
Canvas going down is not an excuse for turning in your work late.

You should submit the two files in digital format. No hardcopy is required for the final submission:

<username>_1.cpp
<username>_1p.txt (script of sample execution, especially the results of testing)

Sample Usage:

Bold letters indicate user's input.

```
     ===========================================================
     |             The TigerBook Messaging System!             |
     ===========================================================

1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

     Please choose an option: 1

     Please enter user name: Paul

     ===========================================================
     |                Welcome to TigerBook, Paul!              |
     ===========================================================

1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 2

         Enter message: Hello there, everyone!
         Welcome to our new chat group.
         Feel free to post messages any time.
         !!

1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 2

           Enter message: Our Bahamas cruise was super!
         Great food and fun in the gulf.
         !!


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 1

         Please enter user name: Mary

     ===========================================================
     |                Welcome to TigerBook, Mary!              |
     ===========================================================

1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 2

         Enter message: Welcome to my new msg page!
         I'm new to this, so be gentle, no flame please.
         !!


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 6

         Enter friend's name: Paul


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
```

```
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 3

          Enter tweet message: We did the Pinhoti trail last year.
          That was exhausting!
          !!


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 2

         Enter message: Hiking the Rockies in 5 days!
         Can't wait to head for the mountains.
         !!


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 4

     ============================================
     |            Mary's Wall Page              |
    ============================================

     Hiking the Rockies in 5 days!
     Can't wait to head for the mountains.

     We did the Pinhoti trail last year.
     That was exhausting!

     Welcome to my new msg page!
      I'm new to this, so be gentle, no flame please.

1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
```

```
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: **1**

         Please enter user name: **Peter**

    ===========================================================
    |                Welcome to TigerBook, Peter!             |
    ===========================================================

1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: **2**

         Enter message: **Summer was cool.
         Still trying to adjust to the semester drill.
         !!**


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: **6**

         Enter friend’s name: **Paul**


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: **2**

         Enter message: **We had a great time at the beach!
         Not the mention scuba diving and sailing.
         !!**
```

```
1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 5

    ===========================================
    |            Peter’s Home Page            |
    ===========================================

   Peter:
   We had a great time at the beach!
   Not the mention scuba diving and sailing.

   Peter:
   Summer was cool.
   Still trying to adjust to the semester drill.

                       More message? (yes/no): no
   Mary:
   We did the Pinhoti trail last year.
   That was exhausting!

   Paul:
   Our Bahamas cruise was super!
   Great food and fun in the gulf.

   Paul:
   Hello there, everyone!
   Welcome to our new chat group.
   Feel free to post messages any time.



1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 7

         Enter user’s name: Paul

    ===========================================================
    |               Welcome back to TigerBook, Paul!          |
    ===========================================================

1) Create a new user
```

```
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 6

         Enter friend's name: Peter



1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 2

         Enter message: Best cruise ever!
         Hit some rough sea though.
         !!

1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 5


     ===========================================
     |            Paul's Home Page             |
    ============================================

   Paul:
   Best cruise ever!
   Hit some rough sea though.

   Peter:
   We had a great time at the beach!
   Not the mention scuba diving and sailing.

                         More message? (yes/no): yes

   Peter:
   Summer was cool.
   Still trying to adjust to the semester drill.

   Mary:
   We did the Pinhoti trail last year.
   That was exhausting!

   Paul:
   Our Bahamas cruise was super!
   Great food and fun in the gulf.

   Paul:
   Hello there, everyone!
   Welcome to our new chat group.
   Feel free to post messages any time.


1) Create a new user
2) Post a message
3) Tweet a message
4) Display Wall page
5) Display Home page
6) Add a friend
7) Switch to a different user
8) Quit TigerBook

         Please choose an option: 8

    ===========================================================
    |               Thank you for using TigerBook             |
    ===========================================================
```