**20CS2028 Data Structures**

**Programming assignment #3**

**Assigned on Friday June 6, 2015**

**Due on Tuesday, June 17 2015 on Blackboard at 11:59PM**

**Total number of points: 30**

The purpose of this program is to design and implement an Inbox class for an e-mail system. The Inbox for this assignment will be a very simplified version of what a real Inbox would be i.e., our Inbox will be for Goofy mail or gmail for short.

Like the real gmail, it will consist of a set of communications, where each communication consists of the set of e-mails involved in that communication, i.e., having the same subject.

A new e-mail will be added as follows:

1. If the subject is different from any communication already in the Inbox, then
   1. A new communication will be created consisting of the new e-mail.
   2. This new communication will then be placed at the beginning of the Inbox.
2. If the subject already occurs in a communication, then
   1. The communication will be added to the beginning of the communication
   2. The communication will be moved to the beginning of the Inbox.

Inbox will be implemented using a **doubly-linked list** of *communication nodes* representing the communications (i.e., e-mails having the same subject). The emails within a communication will also be implemented with a **doubly linked list**.

**Structure of the Communication node: The Communication node will have the following fields**

**Subject:** subject of e-mails associated with communication

**Number of e-mails**: keep track of the number of e-mails in communication

**Email:** A header for doubly linked list of emails associated with the communication

**Structure of the Email nodes**

**To:** string with name (e-mail address) of person who received e-mail

**From:** string with name (e-mail address) of person who sent e-mail

**Message:** Assume that message fits on one line

Some main functions associated with the class Inbox are:

**InsertEmail()** Inserts a new e-mail as described above.

**DeleteCommunication()** Deletes a communication having a given subject.

**SearchCommunicaton()** Searches the Inbox for a communication using subject as the key for the search. It returns a pointer to the communication node that matches the subject, or the null pointer if no such node is found. This function should be private, and will be needed for both insertions and deletion.

**DisplayInbox()** This will display Inbox by listing the communication in the order they appear in the Inbox from the most recent to the oldest, i.e., from beginning of doubly-linked list of communications, displaying the subject and the number of e-mails in the communication.

You will also need to include a **destructor** and a **copy constructor**..

Your main program will input a set of emails, where each mail consists of simply the subject, and displays the inbox.

**Example:**

Input e-mails (subjects) and terminate by entering “done”

*Sample input:*

Weather  
Weather  
Weather  
Picnic  
Picnic  
Birthday Party  
Picnic  
Weather  
Greetings  
Birthday Party

*Sample Run:*

Inbox: total number of emails is 10.

Birthday Party – 2  
Greetings - 1  
Weather - 4  
Picnic - 3