**CSCI1110**

**Assignment 7**

**Due: 12:00pm noon, Thursday, Nov. 23, 2017 (submit on Brightspace)**

**This assignment is worth 2.5% of your grade.**

The purpose of this assignment:

To use LinkedLists.

*Your assignment is on LinkedLists. You should use adapt the Node class and LinkedList class that we wrote in class (do not use Java's LinkedList class). You can adapt the Node class and/or add any necessary methods to the LinkedList class.*

For the assignment, you will be provided with sample inputs and corresponding outputs. All input is done via the console (unless specified otherwise) and all output is to be done to the console as well. Your code **must** compile. If it does not compile, you will receive a 0 on that portion of the assignment.

In this assignment, you will use linked lists to analyze the spread of the a strange virus that has been reported in the area. Suppose that your program reads a number of lines of text, each line containing a pair of names. The first name indicates the person who sneezes or coughs, and the second name indicates the person who is sneezed /coughed on. Assume that this strange virus is potent enough so that the second person catches the virus immediately. For example, the input to the program could be something like this:

Joe Bill

Vic Bill

Joe Vic

Bill Jane

Joe James

Vic Jake

James Rob

Jane Greg

end

(The string “end” signals the end of input).

Your program must read the lines of input and create a linked list. Each node in the list has three attributes – first person’s name, second person’s name and pointer to the next node.

**Part I:** In a Demo program, create methods that uses a the linked list,

* Method 1: Print out the names of all people responsible for spreading the virus.

In the above example, your program should print (print the names only once):

Joe Vic Bill James Jane

* Method 2. Print out the names of all people NOT responsible for spreading the cold virus, even if they have it themselves. In the above example, your program should print

Jake Rob Greg

* Method 3. Given a name s, print the names of all people directly infected by s.

For example, if the given name is Vic, your program should print (print the names only once):

Bill Jake

* Method 4: given a name s and a positive integer n, print out the names of all people who may have contracted the virus from s via a chain of n or fewer sneezes/coughs. For example, given s = “Joe” and n = 2, your program should print (print the names only once)

Bill Jane Vic Jake James Rob

**Bonus:** Implement Part I again using a Double LinkedList (adapted your current node and linked list class to include 'prev' links as well as the next links). *Note, you don't have to implement last method for the bonus*.

**Input**

Node class, LinkedList class and demo program that populates the linkedlist.

**Semantics**

Create a LinkedList with Nodes that have two names (the first is the spreader of the virus, the second is the receiver of the virus). Then using the methods in the demo class, implement the methods to find out more about the virus spread.

**Output**

The names of who spread the virus and the names of the receivers of the virus.

|  |  |
| --- | --- |
| Input | Sample Output (from Play Class) |
| Node class, Linkedlist class and pair of names | Output from the methods |

You will have three files for this program: Node.java, LinkedList.java, Demo.java

If you do the bonus – you will have three additional files: Node2.java, LinkedList2.java, Demo2.java

**What to Hand In**

* Your code must compile. If it does not compile, you will receive a 0 on that portion of the assignment.
* You **MUST** follow the class names declared in this assignment. If you fail to do so you will lose marks.
* Submit the source files for your program in a **.zip** file. The file should not contain other files inside. Submission is to be done via Brightspace.

**If your program does not compile, it is considered non-functional and of extremely poor quality, meaning you will receive 0 for the solution.**

**Marks**

|  |  |
| --- | --- |
|  | Marks |
| **Node Class**  Functionality  Quality of Solution | 2  2 |
| **LinkedList Class**  Functionality  Quality of Solution | 11  12 |
| **Demo**  Functionality  Quality of Solution | 8  10 |
| Code Clarity | 5 |
| **Total** | 50 |
| **Bonus +5 (New Total)** |  |