Programming Assignment #1

The "Linked List" Telephone List

Overview

You are to write a program that performs the functions of a simple Telephone List. This program will allow you to add and manage entries in your list. The program will have a simple user interface that will allow the manipulation of the list, and allow you to add and delete records.

Details

This program will require an implementation of a Linked list **class.** The class should have a collection of members that allow access to the data in the nodes, in addition to the functions to support the linked list operations. You need to be able to create a list of name records. You must support the following commands: Add, Delete, Search, Email, and Print.

You should create a **Node** Class that contains 3 Strings, one for name, one for email, and the other for the phone number, in addition to the reference to the next entry.

You need to write a small user interface (e.g. menu) to show that you can perform the operations. You should demonstrate the following functionality in your program:

- 'a' : Add a phone number to the list.
- 'p' : Print the entire List
- 's' : Search for Name.
- 'e' " Search for email address
- 'd' : Delete an Entry.

The commands can take arguments on the same line as the command, or they can prompt for input. **Add** should add the entry sorted alphabetically by the last word in the name field. **Print** should print each entry with an index in the list associated with it. **Search** should locate all the entries that contain the entered search string, and display them along with the index of the entry. Email functions as search but on email addresses. **Delete** should delete an entry by its index in the printed, or searched list.

All data must be private.

Suggestions (Use at your discretion)

Mapping all input to lower or upper case will allow a smoother interface.
If looking at the extra credit, investigate **Serializable**.

Grading

To receive full credit for this program the following must be submitted:

1)	Pseudo Code or Object Oriented design for the algorithms utilized.	(25%)
2)	Well commented Syntactically Correct Java source code.	(60%)
3)	3 Runs of the program showing all info listed above.	(15%)
4)	10 Points extra credit if you implement a fully functional save and restore to file	

Late programs will not be accepted, without prior approval. Waiting until the last minute (to start this) is probably not a good idea. The program is due at the start of class on October 16th.