Assignment 3 – Due Friday Sept. 21

This exercise is designed to introduce you to sorting and linked lists.

Assignment:

1) Build a program using the Person/Employee/Student hierarchy that you implemented for assignment 2. Your program should have the following functionality:
   - Create an array of Person references, containing at least five employees and five students.
   - Print all persons, ordered by last name.
   - Print all employees, ordered by salary. (You’ll need to first copy references to all employees into a new array before sorting.)
   - Print all students, ordered by major. (You’ll need to first copy references to all students into a new array before sorting.)

You can sort using either an insertion sort or a selection sort algorithm. It is easier to define three separate sorting methods (rather than one method that can sort by three different criteria).

2) Take the singly linked list class from the September 12 lecture notes (the link is called “Basic Linked List Operations”). Add the ability to move through the array and to insert an element after any node in the list.

The best way to do this is to use a current_node reference, that operates in a manner similar to the one given for doubly linked lists in the September 14 lecture.

To do this, you will need to:
   - Add a getNext() method to the SLinkedList class.
   - Add a current_node field to the TestSLinkedList class and initialize it in the main() method.
   - Expand the functionality of TestSLinkedList.getUserOption() and TestSLinkedList.doSomething().

Details:

Submit your solutions via Blackboard.

All programs must compile and run using Java 1.6. No credit will be given for programs that do not compile.

All source code files must include your name and the assignment number (3) in a Javadoc comment at the top of the file.