ENSF 519 – Principles of Software Development
Fall 2018

Lab Assignment #9: Data Structures and Algorithms

Due Dates	
Lab	Submit electronically on D2L **before 11:59 PM on Monday November 25**

This is an individual assignment.

The objectives of this lab are:

1. Data structures and algorithms
2. LinkedList
3. Algorithm complexity


The following rules apply to this lab and all other lab assignments in future:

1. Before submitting your lab reports, take a moment to make sure that you are handing in all the material that is required. If you forget to hand something in, that is your fault; you can't use `I forgot' as an excuse to hand in parts of the assignment late.
2. **20% marks** will be deducted from the assignments handed in up to **24 hours** after each due date. It means if your mark is X out of Y, you will only gain 0.8 times X. There will be no credit for assignments turned in later than 24 hours after the due dates; they will be returned unmarked.


Lab (25 marks)

Exercise 1: Developing a Simple LinkedList (20 Marks)

What to do: Download files Student.java and StudentLinkedList.java from D2L. Implement the following methods:

- **`void InsertInOrder (Student p)`**
  - `Inserts a student object in order of ascending student ids. If the list already contains a student that matches the id of p, p should NOT be added to the list. Note: to use this method, the list must be already sorted.`
- **`Student search (id)`**
  - `Returns s Student object if the id matches a student in the list. Returns null otherwise`
- **`Student removeEndElement ()`**
  - `Removes the element at the end of the list`
- **`Student removeFirstElement ()`**
  - `Removes the element at the beginning of the list`
- **`Student removeElement (int id)`**
  - `Removes an element with the matching ID`
- **`void sort()`**
  - **`sorts the list by implementing bubble sort, or insertion sort (your choice)`**

Also implement the following:

1. **Helper methods:** Implement any helper methods (i.e. private methods) that ensure your code doesn't violate any of the SOLID principles; particularly the single responsibility principle.

2. **In your main:** You must implement necessary code that will test each of the above methods in your main method.

What to Submit: Please submit all the java files including the files you have created/modified in a zip folder. You need to provide the Javadoc comments in your code, but you don't need to submit the HTML files.


Exercise 2: Complexity (5 marks)

Determine the complexity of the binary search algorithm:

```
int binarySearch(int[] arr, int key) {
        int lo = 0, mid, hi = arr.length-1;
        while (lo <= hi) {
             mid = (lo + hi)/2;
             if (key < arr[mid])
                 hi = mid - 1;
             else if (arr[mid] < key)
                 lo = mid + 1;
             else return mid;                    // success: return the index of
        }                                  // the cell occupied by key;
       return -1;                          // failure: key is not in the array;
 }
```

What to Submit: Submit a pdf file containing your answer to exercise 2.

How to submit: Include all your files for the lab section in one folder, zip your folder and upload it in D2L before the deadline.