## CS 2401 Assignment #4

**Due Date:** Friday, October 06, 11:59PM
(See the syllabus for late policy)

**Objective:** The goal of this assignment is to practice linked list of objects.

**Background:** El Paso Packaging and Supply Co. comes back to you. They need your help to convert the array-based packaging to a linked-list based packaging. Additionally, the client would like to change the name of the class `Package` to `Box`.

From Lab Assignment 3, you know that a sample file may look like the following one.

```
20 10 8
4.5 8.45 12.2
8.0 2.5 4.0
1.0 15.0 18.0
3.5 3.5 3.5
6.0 5.0 10.0
```

Each line contains the width, height and length of a box. The dimensions are separated by spaces. In Lab Assignment 3, you could read the input file twice. For Lab assignment 4, you can read the input file exactly once. Yes, sometimes clients can be very difficult. ☹ The client is NOT allowing the use of any array to store `Box` objects, or java.util lists for this revised software.

**Assignment:** You need to write a program using object oriented programming idea for boxes. That is, each box has to be considered an object. To achieve this, you must write a class named `Box`. As you already know, this client likes to keep each class in a separate Java file. Therefore, make sure to create a Java file for the `Box` class. Some other required properties of the `Box` class are as follows.

1. You are allowed to keep only one of the status variables `public`. The rest of the status variables of the `Box` class must be `private`.
2. Write no more than two constructors.
3. The `Box` class must have a public method named `getVolume()` that will return the volume of the box.
4. The `Box` class must have a public method named `isCube()` that will return true if the box is cubic, false otherwise.
5. The `Box` class must NOT contain any main method.

Feel free to write any additional method in the `Box` class, as you see fit.

The program file (the Java file that contains the main method) must be written in `Runner.java`. Runner must have the following functionalities. Each functionality must be implemented in a separate method in `Runner`.

1. Read the input text file provided by the client and create a linked list of `Box` objects. The sequence of the lines should be used in the sequence of objects in the linked list.
2. Find the smallest box in the linked list. Report the position, dimensions, and volume of the smallest object. Position of an object in a linked list is equivalent to the index of an array.
3. Find the largest box in the linked list. Report the position, dimensions, and volume of the largest object. You already know what a position in a linked list means.
4. How many cubic boxes are there in the linked list?
5. What are the position, dimensions, and volume of the smallest cubic box in the linked list? If multiple boxes have the same smallest cubic size, report only the one that appeared the earliest in the input file.
6. What are the position, dimensions, and volume of the largest cubic box in the linked list? If multiple boxes have the same largest cubic size, report only the one that appeared the earliest in the input file.
7. Report average volume of all boxes.
8. Report average volume of cubic boxes.

***Note:*** *It is ok to use a one-dimensional array of length three to keep width, length, and height of a box for string-splitting purpose. You cannot use any array of size larger than three for this assignment. If you use the split method then this array of size three is inevitable. If you want, you can easily avoid the* `split` *method and this array of size three by using* `java.util.StringTokenizer`*. However, there is no extra credit for the use of* `StringTokenizer`*.* ☺

**Deliverables:** You are expected to submit two Java files (`Box.java` and `Runner.java`) via Blackboard. You have to demo your programs within one week after the due date. Your demo will be based on your last submission in the Blackboard. Your TA will instruct you with further details.