

Central Washington University
College of the Sciences
Department of Computer Science
CS-301 Data Structures Fall 2016

Lab Practice 08

In this lab, we will try to measure the problems with insertion in an array of implementations of lists. We will also improve our toy class `SinglyLinkedList`. We will also try to serialize an `ArrayList < E >` object.

Normally you, will find the source and data files in `/home/cs-301/Labs/Lab08`.

```
lb08.pdf
ArrayInsertion.java
SinglyLinkedList.java
Executive.java
executive.txt
Date.java
```

Some of programs above have been at some point in `WeeklyPrograms`.

1. To assess quantitatively how bad insertion can be on an array based implementation, time carefully the program `ArrayInsertion.java`. Time the insertion of 100,000 elements at index 0, and the appending (insertion at end) of similar amount of elements. Use the static method in *System* class

```
public static long nanoTime()
```

for timing. You should see a significant difference.

2. Create a program that does the same thing, but using `ArrayList()`. Insertion at index 0, becomes `add(0,e1)` and appending `add(e1)`. Time the operations as you did before. You should see similar differences.
3. We are going to get our `SinglyLinkedList<E>` class enhanced by implementing more methods of the `List<E>` interface

```
boolean add (int ix, E value)
void add (E value)
```

Implement them and test them with a client of your choice.

4. Create a client that creates an `ArrayList<Executive>` of `Executive`, using the data on the file `executive.txt`. Make sure that your programs reads the data correctly, by displaying some elements.

5. Now we will try to *serialize* the `ArrayList` into the file `executives.obj`. Recall that the I/O classes needed are in `java.io`. What modifications are needed in the classes that define the element? *Hint:* to serialize, use the layering

```
FileOutputStream f = new FileOutputStream(filename);
ObjectOutputStream fo = new ObjectOutputStream(f);
```

and then use the instance method

```
void writeObject( Object c)
```

6. Similarly, you should be able to *deserialize* the object from the file using the corresponding method in `ObjectInputStream`. Check the API for details.
7. Finally, repeat what we did in 1. (and 2.) for our class `SinglyLinkedList<E>`, using the newly implemented methods, (or `addToFront()`) Can you explain the results?