CSCI 1101 – Winter 2013
Lab. No. 7

This lab is on ArrayLists.

Your task is write and compile the class file followed by the tester program. Include all the source codes for each program. Make sure that you test your program for a number of different input scenarios. Please submit on Moodle before 12 noon, March 9, 2013.

Revision of ArrayLists– Here's a simple test program that creates an ArrayList of String objects and manipulates them. Try it out. Make modifications to the method calls and understand their operations.
No submission required for this exercise.

Warning: Cutting and pasting the code may cause errors!

```
import java.util.ArrayList;
public class ArrayListTest {
  public static void main(String[] args) {
    ArrayList<String> list = new ArrayList<String>(); // Create a list

    // Add elements to the list
    list.add("America"); // Add it to the list
    System.out.println("(1) " + list);

    list.add(0, "Canada"); // Add it to the beginning of the list
    System.out.println("(2) " + list);

    list.add("Russia"); // Add it to the end of the list
    System.out.println("(3) " + list);

    list.add("France"); // Add it to the end of the list
    System.out.println("(4) " + list);

    list.add(2, "Germany"); // Add it to the list at index 2
    System.out.println("(5) " + list);

    list.add(5, "Norway"); // Add it to the list at index 5
    System.out.println("(6) " + list);

    // Remove elements from the list
    list.remove("Canada"); // Same as list.remove(0) in this case
    System.out.println("(7) " + list);

    list.remove(2); // Remove the element at index 2
    System.out.println("(8) " + list);

    list.remove(list.size() - 1); // Remove the last element
    System.out.println("(9) " + list);

    if (list.contains("France"))
      System.out.println("France found at index: " + list.indexOf("France"));

    System.out.println("What is the element at index 1 ? : " + list.get(1));
  }
}
```

Exercise 1: Write a program that reads words into two ArrayLists list1 and list2 and then creates a third ArrayList that contains words which are common to both list1 and list2. You may assume that the strings are entered as words separated by a space on a single line and the end is signaled by "-1" (String -1). You can use keyboard.next() to read each word.

A sample dialog is shown below:

```
Enter words on one line, end with -1
java c pascal ada java c++ -1
Enter words on one line, end with -1
c pascal java lisp lisp -1
[java, c, pascal, ada, java, c++]
[c, pascal, java, lisp, lisp]
Array List with common strings:
[c, pascal, java]
```

Exercise 2: Write a program that reads words into an ArrayList `list1` and creates another ArrayList `list2` that has the same words in `list1` but no duplicates.

A sample dialog is shown below:

```
Enter words on one line, end with -1
java c pascal ada java java ada c++ -1

Array List with no duplicates:
[java, c, pascal, ada, c++]
```

Exercise 3: Modify the above program so that it removes the duplicates in the original ArrayList.

Note that it can be tricky to scan an ArrayList and simultaneously remove items from it. This is due to the fact that when you remove or add an item, the list adjusts itself and hence the index used for scanning will not be pointing to the same location.

Exercise 4:

The objective of this exercise is to write a program that reads a number of Strings and sorts them by inserting each string into the appropriate place in an arraylist. For example, if the strings are:

Shai
Ralph
Hillary
Tom
Barbara
Fred

Then the arraylist should grow as follows:
[Empty]
[Shai]
[Ralph, Shai]
[Hillary, Ralph, Shai]

[Hillary, Ralph, Shai, Tom]
[Barbara, Hillary, Ralph, Shai, Tom]
[Barbara, Fred, Hillary, Ralph, Shai, Tom]

The algorithm to sort is simple. As you read each name (say name1), compare it with each name (say name2) stored in the arraylist starting from the index 0. As soon (name1.compareTo(name2) >0), that is the right index to put name1. Be sure that you do not cross the arraylist boundary.