LAB ASSIGNMENT #5 – Java Collections Framework and Multithreading

Due Date: Wednesday, March 26, 2014.

Purpose: The purpose of this Lab assignment is to:

- Practice the use of JCF in Java applications
- Develop multithreaded applications

References: Read the course's text "Introduction To Java Programming, 9th edition", http://www.cs.armstrong.edu/liang/intro9e/, chapter 21,22,23,32 and the ppt slides. This material provides the necessary information that you need to complete the exercises.

This lab must be completed individually by all the students. You will have to demonstrate your solution in a scheduled lab session when submitting the assignment. The assignments/projects should be submitted **through the assignment link on Blackboard**.

The Eclipse project for this assignment should be named as: *FullName_CXEC320_Assignment4.* Each exercise should be included in a separate package. For example, first exercise in a package named *exercise1*, etc.

The entire project directory should be zipped in a file named as *FullName_ CXEC320_Assignment4.*

1. Read and work through Chapters 20, 21, 22, 23, 32 in textbook.

2. **Exercise 22.21** (Use Comparator) Write the following generic method using bubble sort and a comparator.
   public static <E> void bubbleSort(E[] list,
   Comparator<? super E> comparator)

   Write a test program that creates an array of 10 GeometricObjects and invokes this method using the GeometricObjectComparator introduced in Listing 22.4 to sort the elements. Display the sorted elements. Use the following statement to create the array.

   GeometricObject[] list = {new Circle(5), new Rectangle(4, 5),
   new Circle(5.5), new Rectangle(2.4, 5), new Circle(0.5),
   new Rectangle(4, 65), new Circle(4.5), new Rectangle(4.4, 1),

new Circle(6.5), new Rectangle(4, 5)};

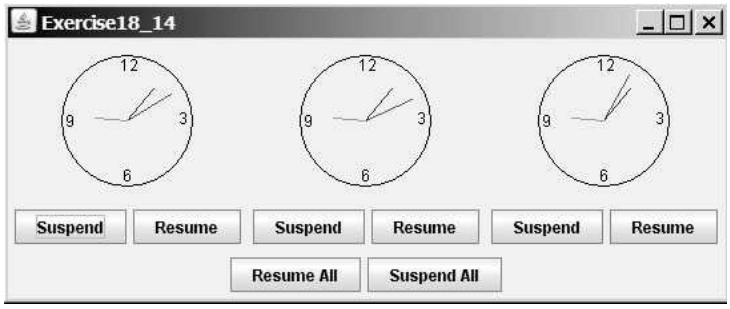
(3 marks)

3. **Exercise 25.13** (Execution time for sorting) Write a program that obtains the execution time of bubble sort, merge sort, quick sort, and heap sort for input size 50,000, 100,000, 150,000, 200,000, 250,000, and 300,000. Your program should create data randomly and print a table like this:

```
Array size   |Bubble Sort      Merge Sort      Quick Sort      Heap Sort       Radix Sort
-------------------------------------------------------------------------------------------------------
50,000       |
100,000      |
150,000      |
200,000      |
250,000      |
300,000      |
```

(2 marks)

4. **Exercise 32.7** (Control a group of clocks) Write a Java applet that displays three clocks in a group, with control buttons to start and stop all of them, as shown in Figure below. Use LISTING 13.10 StillClock.java to implement the clock. Use a thread to control the clock animation.



(5 marks)