Lab Assignment #4

Due Date: **On or before Second Class of Week 10** **Marks/Weightage: 30/8%**

Purpose: The purpose of this Lab assignment is to:

- Practice the use of Collections and Multi-threading

References: Read the course's text "Java How to program, 11th edition Early Objects", **chapters 16 & 23** and the lecture notes/ppts. This material provides the necessary information that you need to complete the exercises.

Instructions: Be sure to read the following general instructions carefully:

This lab should be completed individually by all the students. You will have to demonstrate your solution in a scheduled lab session and submitting the assignment **through drop box link on e-Centennial**.

>> At the start, you must name your **Eclipse work space** according to the following rule:

FirstName_LastName_SectionNumber_COMP228_Labnumber

For Example: ***John_Smith_Sec006_COMP228_Lab04 (say if your section number is 006)***

>> And after that your **project name** should be as follows:

FirstName_LastName_SectionNumber _Labnumber

For Example: ***John_Smith_Sec006_Lab04***

>>Each exercise should be placed in a separate package named as firstname_last-name_*exercise1*, firstname_last-name_*exercise2* etc.

>> After you complete, exit eclipse and go to workspace folder, zip it up and you will get the following zip file.

FirstName_LastName_SectionNumber_COMP228_Labnumber.zip

Example: **John_Smith_Sec006_COMP228_Lab04.zip** *(if your section is 006..)*

>> Apply the naming conventions for variables, methods, classes, and packages:

- *variable names* start with a *lowercase* character for the first word and uppercase for every other word
- *classes* start with an *uppercase* character of every word
- **packages** use only *lowercase* characters
- *methods* start with a *lowercase* character for the first word and uppercase for every other word

Note: Late submissions are accepted until up to three days past due date with 25% deductions. After that no submission will be considered.

Exercise 1:

[5 marks]

Write a program that reads in a series of first names and eliminates the duplicates by storing them in a set. Allow the user search for first name.

Exercise 2:

[5 marks]

Write a program that determines and prints the number of duplicate words in a sentence. Treat uppercase and lowercase letters the same. Ignore punctuation.

Exercise 3:

[10 marks]

Write an app in java that sorts and displays the objects of Employee class based on increasing salary. Create an **Employee** class having **name** and **salary** as instance data members. Define getters and setters. Add one constructor which initializes the instance variables. Also add **toString()** method displaying these members.

Create another public class – **EmployeeTest,** having main method, in the same file. In this class, create a list of three employee objects, initialize them and display them in ascending order based on their salaries.

Exercise 4:

[10 marks]

Refer Example 03 of code examples for Chapter 23 – Multithreading posted on e-centennial. It has three classes defined – **SimpleArray**.java, **ArrayWriter**.java and **SharedArrayTest**.java
There are two threads created in this example. Every thread is writing **three integers values** in a **synchronized** way, in a shared array of size six.

You need to modify the above example as follows:

#1. You need to create three threads instead of two.
#2. You need to define array size of 9
#3. Every thread is writing three random integer values in the following manner:

- Thread 1 is writing any three values in the range of 0 to 10
- Thread 2 is writing any three values in the range of 11 to 20
- Thread 3 is writing any three values in the range of 21 to 30