

## LAB ASSIGNMENT #2 – Using Graphics API, Exception Handling and Text I/O

**Due Date:** Wednesday, February 5, 2014.

**Purpose:** The purpose of this Lab assignment is to:

- Practice the use of Graphics capabilities in Java Applications
- Practice Exception Handling and Text I/O

**References:** Read the course's text "Introduction To Java Programming, 9<sup>th</sup> edition", <http://www.cs.armstrong.edu/liang/intro9e/>, chapter 13, 14 and the ppt slides. This material provides the necessary information that you need to complete the exercises.

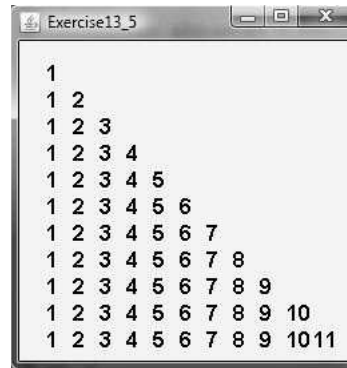
This lab must be completed individually by all the students. You will have to demonstrate your solution in a scheduled lab session when submitting the assignment. The assignments/projects should be submitted **through the assignment link on Blackboard**.

The Eclipse project for this assignment should be named as:

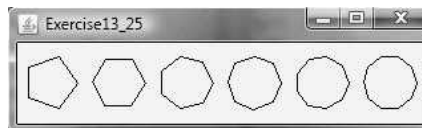
*FullName\_CXC320\_Assignment2*. Each exercise should be included in a separate package. For example, first exercise in a package named *exercise1*, etc.

The entire project directory should be zipped in a file named as *FullName\_CXC320\_Assignment2*.

1. Read and work through Chapters 13,14: "GUI Basics" in textbook (p. 480 - 554)
2. Implement the following exercises:  
Exercise 13.5: (Display numbers in a triangular pattern) Write a program that displays numbers in a triangular pattern, as shown in Figure 13.27b. The number of lines in the display changes to fit the window as the window resizes.



Exercise 13.25: (Geometry: display an n-sided regular polygon) Define a subclass of JPanel, named RegularPolygonPanel, to paint an n-sided regular polygon. The class contains a property named numberOfSides, which specifies the number of sides in the polygon. The polygon is centered in the panel. The size of the polygon is proportional to the size of the panel. Create a pentagon, hexagon, heptagon, octagon, nonagon, and decagon from RegularPolygonPanel and display them in a frame, as shown in Figure 13.33a.



(5 marks)

### 3. Implement the following exercises:

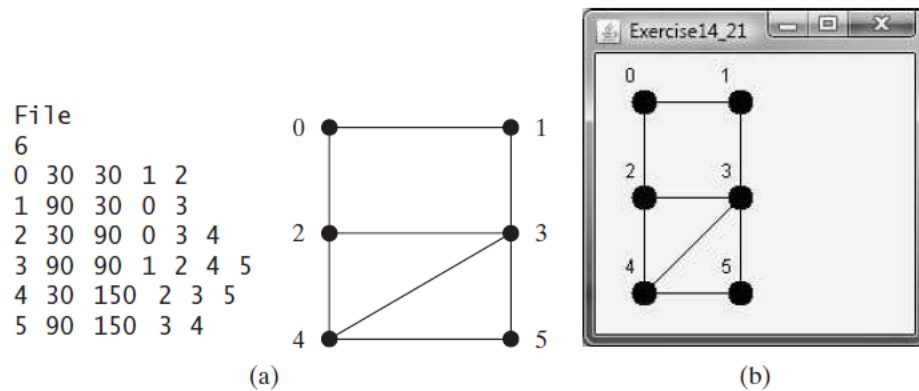
Exercise 14.11: (Remove text) Write a program that removes all the occurrences of a specified string from a text file. For example, invoking  
`java Exercise14_11 John filename`  
 removes the string John from the specified file. Your program should get the arguments from the command line.

Exercise 14.21 (Display a graph) A graph consists of vertices and edges that connect vertices.

Write a program that reads a graph from a file and displays it on a panel. The first line in the file contains a number that indicates the number of vertices (n). The vertices are labeled as 0, 1, . . . , n-1. Each subsequent line, with the format `u x y v1 v2 ...`, describes that the vertex u is located at position (x, y) with edges (u, v1), (u, v2), and so on. Figure 14.14a gives an example of the file for a graph. Your program prompts the user to enter the name of the file, reads data

from the file, and displays the graph on a panel, as shown in Figure 14.14b.

Write another program that reads data from a Web URL such as <http://cs.armstrong.edu/liang/data/graph.txt>. This program should prompt the user to enter the URL for the file.



**FIGURE 14.14** Exercise 14.21 reads the information about the graph and displays it visually.

(5 marks)