11. Eleventh Assignment – Two Options

You have two options for the last homework assignment -- you can either do Problems 25.1 and 25.3 on page 960 of your textbook or you can do the "Graphical Display" assignment described below. Each assignment is worth ten points, and you must choose one or the other. In particular, you cannot turn in both assignments and take the higher of the two grades, and you can't earn extra credit by doing both of them.

The problem from the book is strongly related to the data structures and algorithms that we have focused on in this course (trees in particular). The graphical display assignment is a capstone of the Facebook application we have been building throughout the course. It involves modifying existing code that uses an external Java library to meet some new requirements, which is the type of task you could expect to be given as a newly hired software developer. The book problem involves writing more lines of code, but the graphical display involves gaining an understanding of more new concepts.

<u>Option 1: Problems 25.1 and 25.3</u>

For this option, you can download BST.jar from the course website as a starting point. Then implement the breadthFirstTraversal, height, and non-recursive inorder traversal methods described in Problems 25.1 and 25.3 (you need to do both). Write a driver program to test the operation of your methods for trees containing two different data types. Be sure to test both normal and boundary cases for each method.

If you choose this option, you will be graded according to the following rubric (note that not all items are worth the same number of points).

- The breadthFirstTraversal method is correctly implemented (2 points)
- The height method is correctly implemented (2 points)
- The inorder traversal method has been re-implemented such that it doesn't use recursion (2 points)
- You have written a driver program that tests these new methods on two trees containing different types of data (1 point)
- Your test plan is logical and considers both normal and boundary test cases (1 point)
- The program compiles and runs (1 point)
- The program is clearly written and uses standard coding conventions (1 point)
- ***Note:*** *If your program does not compile, you will receive a score of 0 on the entire assignment*
- ***Note:*** *If you program compiles but does not run, you will receive a score of 0 on the entire assignment*
- ***Note:*** *If your Eclipse project is not exported and uploaded to the eLearn drop box correctly, you will receive a score of 0 on the entire assignment*

- ***Note:*** *If you do not submit code that solves the problem for this particular assignment, you will not receive any points for the program's compiling, the program's running, or following standard coding conventions.*

Option 2: Graphical Display

For this option, you can download GraphicalFacebook.jar from the course website as a starting point. This code uses a popular external (i.e. not built-in to Java) library called the Java Universal Network Graph (JUNG). This program's driver contains a new menu option (13. Display graph) which calls a new display method in the Facebook class. The display method prompts for a username and password and pops up a window showing that user and all of the people she is friends with. The graphical representation is created by the new GraphViewer class.

You will need to make the following changes to this code:

- Change the graph so that the edges are directed (i.e. instead of plain lines, the edges are arrows that point from a person to their friends)
- Change the color of the circles -- keep the circle representing the original user red but make all the others blue
- Change the Facebook class's display method to prompt for the number of levels to display. This number must be non-negative. You can also place some reasonable upper bound on it, if it makes sense to do so.
- Modify the GraphViewer class so that instead of showing just the user and her friends, it shows the user, her friends, her friends' friends, her friends' friends' friends, and so on, up to the number of levels specified by the user.

If you choose this option, you will be graded according to the following rubric (note that not all items are worth the same number of points).

- The edges in the graph have arrows showing the direction of the friendship (2 points)
- The color of the original user is red and the color of all other nodes is blue (2 points)
- The display menu option prompts for the number of levels to display (1 point)
- The correct users and links between them are shown (3 points)
- The program compiles and runs (1 point)
- The program is clearly written and uses standard coding conventions (1 point)
- ***Note:*** *If your program does not compile, you will receive a score of 0 on the entire assignment*
- ***Note:*** *If you program compiles but does not run, you will receive a score of 0 on the entire assignment*
- ***Note:*** *If your Eclipse project is not exported and uploaded to the eLearn drop box correctly, you will receive a score of 0 on the entire assignment*

- ***Note:*** *If you do not submit code that solves the problem for this particular assignment, you will not receive any points for the program's compiling, the program's running, or following standard coding conventions.*