Lab 7: Trees

SE2205: Data Structures and Algorithms using Java – Winter 2015
Discussion: March 19th   **Due**: Sunday March 29

Modified and reviewed by Alexandra L'Heureux (alheure2@uwo.ca) and Vasundhara Sharma
(vvasundh@uwo.ca) , for more information please send an email to Alex (TA, SE2205) or Vasu
(TA.SE2205)

A. Rationale and Background

In this lab we will be discussing the ADT Tree. In particular we will be taking a look at **Binary Search Trees.** You will be responsible for writing a small application which will handle the creation and the traversal of a tree. First, you will write a GUI-based application which will allow you to create a tree using integers. This application shall allow you to capture the input from the user to add to the tree. At each step, you shall print out the tree. Finally, you will add 3 other functions to your GUI ; one to print out the structure of the tree, one to print out the tree content in pre-order traversal and an other to print out the content in post-order.

B. Evaluation

You will get credit for your lab when you demonstrate it and get your TA's approval. Submitting your lab work online is required; failure to do so will result in a grade of 0%. Additionally, your TA will ask some questions about your coding during demonstration and you need to articulate your idea clearly. In place of questions, the TA may also ask you to modify your program in a small way - it is expected that you have sufficient understanding of the code to do so.

Your mark will be determined as follows:
Completion: Program Demonstration of Base Requirements (50%)
Understanding: Program Demonstration of Changes or Response to Questions (50%)

Please note that you may only demonstrate the code you have previously submitted to OWL.
Please ensure that the file naming conventions are being followed. You should be able to rename your classes once they are done by right clicking on them, selecting *refactor* and *rename*.

1. SUBMISSION INSTRUCTIONS

Number of files :   2
Files to be submitted :   All Classes required
Rename these files as :   your_uwo_user_name_lab07_Tree.java,
                                        your_uwo_user_name_lab07_Main.java

C. Lab Questions

1. Design your Tree Class

In the first part of the lab you shall design your tree ADT. You must keep in mind that each of the tree "nodes" keep track of what is on their left and what is on their right. You could make use of code you have previously written when you implemented your own linked list. You shall not use any of java's classes for this step.

a. Iterator

Your class should use iterators to allow for the different tree traversal and for the printing of the tree.

b. Functions

Your tree class shall have the following functionalities:
- Add an item
- Print the tree structure
- Print the tree content in post order
- Print the tree content in pre order

2. Create a GUI

Like in previous labs, you should build a simple GUI. The GUI shall have the following functionalities:

a. Enter a Number

This function shall allow the user to enter a number. Upon pressing enter within the field, the number shall get added to the tree. In the case that a tree does not exist, a new tree shall be created. Every time a new number is entered the tree structure shall be printed within the GUI Window. You shall erase the previous content before printing again.

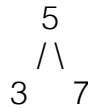For example within the windows the output shall look like this :
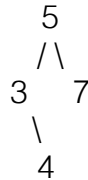
Enter 5:

5

Enter 3:

```
  5
 /
3
```

Enter 7:

```
        5
       / \
      3   7
```

Enter 4:

```
        5
       / \
      3   7
           \
            4
```

b. Print the Tree Structure

This functionality shall print the tree structure as it is at the current stage as shown in the previous step. HINT: In order to do this you must keep track of whether or not you are currently on left or right of a node, the slashes shall be printed accordingly. You shall erase the previous content before printing again.

c. Print in Post-Order

The content of the tree shall be printed on a single line, with each number separated by a comma according to Post-Order traversal of the tree. You shall erase the previous content before printing again.

d. Print in Pre-Order

The content of the tree shall be printed on a single line, with each number separated by a comma according to Pre-Order traversal of the tree. You shall erase the previous content before printing again.

d. GUI Structure

Your GUI shall look something like this:

Lab 07: SE2205b, Winter 2015
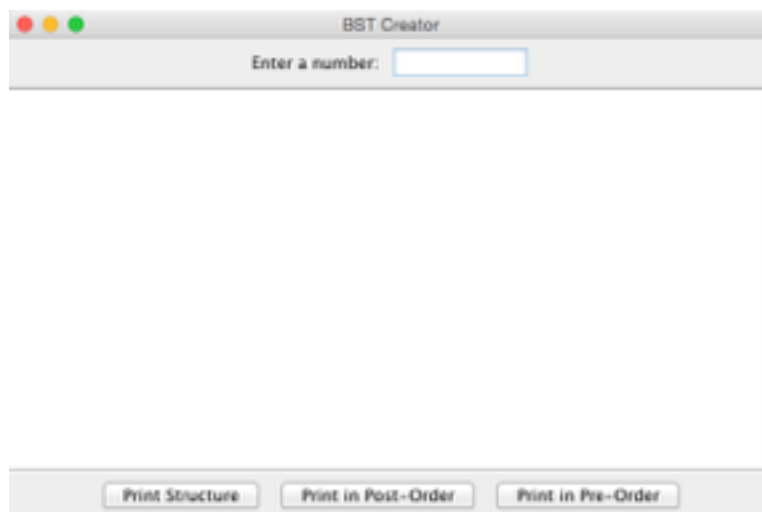
E. Learning Checkpoint

This lab should have introduced you to these concepts of Java:
1. Introduction to Tree ADT.
2. Design of an ADT.
3. Tree Traversals.

Lab 07: SE2205b, Winter 2015