

Assignment Prefix: Lab08

Due Date: Friday, March 10<sup>th</sup> @ 11:59pm

Points: 100

This is an individual assignment.

**Objective:** In this lab assignment you will be implementing the abstract binary tree, the linked binary tree.

Create a NetBeans project named Lab08 and ensure it is saved to a location like desktop or your flash drive.

In completing this assignment, you must include and use the interfaces and classes for the abstract tree, abstract binary tree and the linked binary tree ADTs presented in the textbook.

A large part of this exercise will be gathering together the various code fragments and placing them in the right classes.

Be sure to review code fragments 8.1 to 8.28 to make sure that you have include all of the necessary code. In some cases, the needed code may not appear in a formal code fragment but is implied in the text.

You will probably not get rid of all of the compiler errors until you include all of the necessary code fragments in their correct classes.

Implement the `LinkedBinaryTree` class described in code fragments 8.8 to 8.11 in the textbook. Also implement all classes that the `LinkedBinaryTree` class depends on.

Many of these classes will just need to be copied from previous assignments.

When you implement the `Tree` interface you may import the `java.util.Iterator` class.

When you implement the `AbstractBinaryTree` class you may import the `java.util.ArrayList` class and the `java.util.List` class.

Once you have successfully implement the `LinkedBinaryTree` class:

Create a client class that does the following:

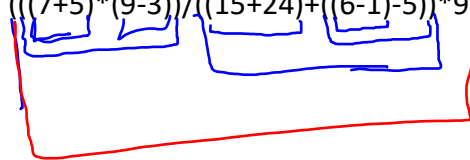
- Manually creates an instance of a binary expression tree that represents the following expression:
- $((7+5)*(9-3))/((15+24)+((6-1)-5))*9$
- Do this by using methods such as `addRoot`, `addLeft`, `addRight`, `set`, and `attach`.
- The element should be of type `String`

- Note that you are manually building this specific expression tree, the pseudo code might look something like:
  - o Create a LinkedBinaryTree
  - o Add "\*" as the root to the tree
  - o Add "/" as the left child to the tree
  - o Add "8" the right child to the tree
  - o Add a left child to the root's left child
  - o Add a right child to the root's left child
  - o etc.

You may want to consider if it would be easier to build the tree in a top down fashion or in a bottom up fashion.

Once you have created the expression tree have your client print out the following:

- The string that represents the expression, i.e.  $((7+5)*(9-3))/((15+24)+((6-1)-5))*9$
- The preorder traversal of the tree
- The inOrder traversal of the tree
- The postOrder traversal of the tree
- The breathFirst traversal of the tree
- The preOrderIndent traversal of the tree
- The parenthesized representation of the tree



#### Things to turn in:

- Open a Microsoft Word document
- Copy and Paste the source code of each of your classes
- Create a screen capture of your NetBeans IDE that includes the contents of the Output Windows from both the programs and paste it into your Word document below your source code.
  - o To create a screen capture of your NetBeans IDE
    - Select, left-click in the NetBeansIDE
    - Use Alt-PrintScreen to place an screen capture image on the clipboard
    - Use Ctrl-V to paste the contents of the clipboard into your Word document
- Next, zip the Project folder.
- Finally on blackboard, submit both your Word document and project zipped file using separate uploads but in one submission.