

CSCI 230, Summer 2015  
HW 5 Binary Tree traversals

**Collaboration policy:** the assignment is to be completed individually. You should attempt to solve the problem. If necessary, you may use online resources to help you, but you should not search for solutions to this specific problem. For example, it is OK to find information on trees and traversals and queues, but you should not simply find copy and use it verbatim. **Be sure to site all resources** and be very specific about how you used what you found. You cannot be too honest in explaining this. You may discuss the assignment with the instructor or with the TA. You may post general questions to piazza, and specific questions to the instructors privately on piazza. You may not post code publically.

**Documentation:** Every file should begin with a header comment that describes the purpose of the file and includes your name.

Each method must be preceded by something like:

```
/**
 * methodName and what it accomplishes
 * @param variableName and what it is for- include for each parameter
 * @return required for value returning methods, what is being
 * returned?
 */
```

If there are preconditions required or exceptions thrown, those must also be described.

----

Problem description:

Modify the author's BinarySearchTree class (see General Resources on piazza).

1. Add 3 printTree methods (discussion in section 4.6 in text)
  - a. printTreePre displays tree contents in pre-order.
  - b. printTreePost displays tree contents in post-order.
  - c. printTreeLevel displays the tree contents in level-order. (Details of this one below)

**printTreeLevel algorithm:** Use a queue (a LinkedQueue from HW 4 will work fine) of Node type,.

```
If the tree is not empty
    Enqueue the tree's root node
    While the queue is not empty,
        Dequeue a node;
        Output the node's data value
        If the node's left child is not empty, enqueue the node that is the left child
        If the node's right child is not empty, enqueue the node that is the right child
```

2. Write a main method that instantiates multiple binary search trees and displays the results of all printTree algorithms on each tree. Your output should be clearly labeled. Include an empty tree in your testing.

**What to submit:** Submit BinarySearchTree.java to OAKS **no later than Friday, July 31, 10:00pm.**