Name ____________________ CS 102 - Quiz 3

This is a timed quiz. There are 8 questions and you have 40 minutes to complete the quiz. Submit your answers on blackboard. (35 points)

1) (2 points) Which of the following is NOT a property of a complete binary tree of height h?
   a) all nodes at level h – 2 and above have two children each
   b) when a node at level h – 1 has children, all nodes to its left at the same level have two children each
   c) when a node at level h – 1 has one child, it is a left child
   d) all leaves are at level h

2) (2 points) A node on the path from the root to node n is a (n) ______ of node n.
   a) ancestor
   b) descendant
   c) subtree
   d) leaf

3) (2 points) In ______, the left and right subtrees of any node have heights that differ by at most 1.
   a) all trees
   b) all binary tress
   c) n-ary trees
   d) balanced binary trees

4) (2 points) A data element within a record is called a ______.
   a) field
   b) tree
   c) collection
   d) key

5) (1 point) All binary search trees are balance.
   a) True
   b) False

6) (1 point) Inorder traversal visits a node before it traverses either of its subtrees.
   a) True
   b) False

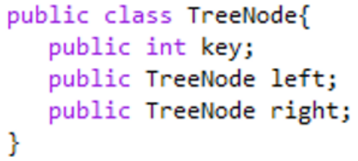
7) (15 points) Consider a binary search tree of integers, orders traditionally (smaller values to the left, larger values to the right)
   a) Draw the final binary search tree that results from adding the following items to an empty tree, in the order shown below:

      1, 7, 2, 5, 8, 3, 6

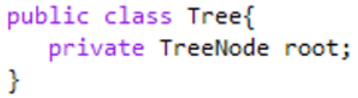
   b) Consider the binary search tree above, list the tree nodes as visited in the: preorder traversal and postorder traversal.

c) Draw the binary search tree that results from removing the value ‘2’ from the tree you drew above.

8) (10 points) Consider the flowing definitions for a binary tee:

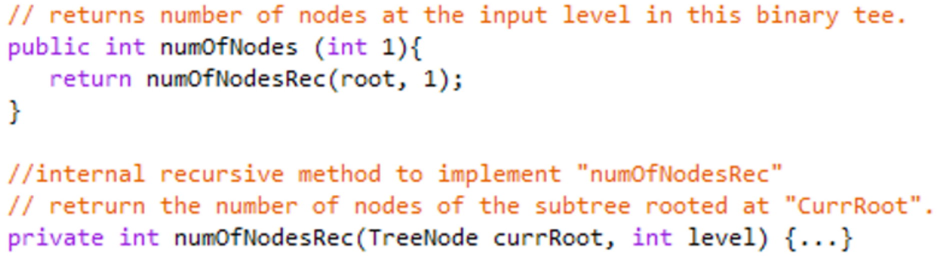


```
public class TreeNode{
   public int key;
   public TreeNode left;
   public TreeNode right;
}
```



```
public class Tree{
   private TreeNode root;
}
```

Implement (recursively) the method **numOfNodesRec** for the class **Tree** to compute the number of nodes at a specific level (e.g. the root node is at level 1, its children at level 2etc.) in a *binary tree.*



```
// returns number of nodes at the input level in this binary tee.
public int numOfNodes (int 1){
   return numOfNodesRec(root, 1);
}

//internal recursive method to implement "numOfNodesRec"
// retrurn the number of nodes of the subtree rooted at "CurrRoot".
private int numOfNodesRec(TreeNode currRoot, int level) {...}
```