Programming Assignment #7

Binary Search Tree implementation

CS 3358.751, Summer II 2013
Instructor: Jill Seaman

Due: Wednesday, 8/7/2013 (upload electronic copy by 11:30am)

Problem:

You will implement the BST_3358 interface, which is a templated binary search tree.

bst_3358.h the interface and incomplete implementation

bst_test.cpp a test file

bst_out.txt the expected output for the test file

In the file bst_3358.h, you will find 10 functions with **//implement me** in their function definition bodies. These are the ones you need to implement.

Note that there are 15 public functions. There are 12 private functions. Many of the public functions call a private function to do their work. The private functions take pointers to TreeNodes as parameters so that they can be called recursively (the public functions cannot take TreeNodes as arguments, because TreeNodes are private). The public and private functions are overloaded (have the same names, different param lists). Feel free to change the names of the PRIVATE functions if you prefer.

You will implement 10 of the private functions:

- copyTree
- findItem
- findMin
- findMax
- insertItem
- countNodes
- makeEmpty
- inOrderTraversal
- preOrderTraversal
- postOrderTraversal

All of the other functions (including delete) have been implemented.

NOTES:

- bst_3358.h is set up so it will already compile with the bst_test.cpp file. Implement one function at a time, compile, test, and fix.

- I recommend implementing insertItem, and then findItem and inOrderTraversal, so that you can test your insert (comment out the other tests from bst_test.cpp). Everything depends on getting insert implemented correctly.

- Save makeEmpty and copyTree for last. They are a little more challenging. For copyTree, make sure your copy has the same structure as the original (same root node, same children, etc. -- do not produce a tree containing the same nodes with a different structure).

- Most of the definitions will be short, one or two recursive calls with a base case or two. Don't try to make it more complicated than it needs to be. Don't say "it can't be that simple". It could be.

Logistics:

Please submit the following file:

`bst_3358_xxxxxx.h`

The xxxxx is your TX State NetID (your txstate.edu email id).

Submit: an electronic copy only, using the Assignments tool on the TRACS website for this class.