CS163 Lab Session #7 – Binary Search Trees

Please complete this to become familiar with Binary Search Trees.
Submit code to the CS199 D2L dropbox. Limit the time invested to 1 hour and 50 minutes maximum.

Coding: *With this lab, we will be working with an existing class implementing a BST of integers. You have access to the .h files through D2L's online "locker" to see what functions are available. Your job will be to implement functions* ***recursively****.*

_____Step 1. Begin implementing the member functions, in a .cpp file and upload these to D2L's CS199 dropbox. Examine the .h file for the correct prototypes

1. Count the number of nodes in a BST with no children
2. Traverse the tree to determine the height
3. Make a complete copy of a BST, creating a new BST
4. Calculate the sum of all data in the BST, returning the result
5. Deallocate all nodes in a BST

_____Step 2. Compile and run each:

1. Modify main to call your functions (or double check main is correct for your implementation)
2. Download the .h and .o files from D2L's online "locker"
3. Compile: g++ *.cpp *.o
4. Run: ./a.out

_____Step 3. **Develop the test plan:** *For each member function that you plan to write, think about how to test it – what flow of control exists in the member function and how would you test out all conditions:*

Test Case(s)	Expected Result	Verified? (yes/no)
Count leafs	Output the number of leafs	Yes
Determine the height	Output the height of the tree	yes
Sum the data in the nodes	The sum of the nodes (verified by the provided #)	yes
Copy a tree	Nothing outputted	yes
Compare the copied tree to the original	Same output	yes
Remove the data from a tree	Output "Empty Tree!"	yes

Self-Assessment: *What could you do to improve for next time?*