**cs311 Yoshii HW4 Binary Search Tree (based on week 8)**

**===================================================**

**DUE: Week 10 Tuesday at the beginning of class**

**TOTAL: 33 pts Your score is:**

**Your name: Thomas Griffin**

**Date turned in: 11/3/15**

**\*Did you do the extra credit?[5pts] <answer here or we will not look for it>**

**---------------------------------------------------**

**Purpose: To learn the representation and implementation of a binary search tree.**

**----------------------------------------------------**

**Questions: [3pts] Your score:**

Q1. Depth first traversal is the same as **preorder** traversal.

Q1. When we add a new node to an existing binary search tree, it will always become a **leaf**.

Q2. When we delete a node with 2 children from a binary search tree, we replace the node with

**The right child.**

**PROGRAMMING: Binary Search Tree [30pts] Your score:**

**=============================================================**

**Your job is to complete my binstree.C according to my instructions**

**And test it with my hw4Client.C**

**Some of these files in /cs/cs311RY are in WORD (.doc) format**

**to show in blue exactly where changes need to be made.**

**Q) State of the program statement [2pts] <answer here>**

* **Does your program compile without errors?**

**Yes**

* **List any bugs you are aware of, or state “No bugs”:**

**No bugs**

**Submit these files:**

1. **this assignment sheet**
2. **my binstree.h unchanged**
3. **binstree.C updated**
4. **my Hw4Client.C unchanged**
5. **test results (one run) which is either a script or screen dump**

**showing exactly what was compiled and what was executed.**

**EXTRA credit [5pts] – highly recommended to do**

**Do this separately from HW4. After your HW4 is completed, make a copy of the files and rename them as binstreeEC.h, binstreeEC.cpp, hw4EC.cpp, and then add the functionality to compute and store the height of each node.**

**Ideally this should be done each time Insertvertex or Deletevertex is called – updating all nodes so that the tree can be balanced as soon as a bad balance factor is noticed.**

**However, to make this assignment easier, you should simply compute, store and display the height and the balance factor of all nodes when ShowPostOrder is called.**

**Must work perfectly to receive any points. No partial credit. So, if it does not work perfectly, do not submit it.**

* **Need to update the header file to allow storing of the height and balance factor in each node**
* **Need to update the implementation file to compute, store, and display the height of each node in ShowPostOrder.**
* **Update the hw4Client.C so that ShowPostOrder is called right after each ShowInOrder.**

**Submit these additional files:**

* **The header binstreeEC.h, implementation binstreeEC.cpp and client hw4EC.cpp.**
* **The test results TestEC**