CST-201

24 November 2018

Green Group

Jackie Adair

Zachary Chambers

Dylan Dorn

// 6.14. 2. Write functions

// a.to count the number of nodes in a binary tree

// b.to count the number of leaves

// c.to count the number of right children

// d.to find the height of the tree

// e.to delete all leaves from a binary tree

6.14.2a – NodeCountPrivate(node\* ptr) is called by NodeCount() and utilizes recursion to increment the count integer everytime there is a node called. The right and left calls are added and the value is returned.

6.14.2b – LeafCountPrivate(node\* ptr) is called by LeafCount() and utilizes recursion to check each node for children (left and right). If none are found, it returns 1. Left and right are again added.

6.14.2c – RightChildCountPrivate(node\* ptr) is called by RightChildCount() and again, utilizes recurision. This time, while both left and right are traversed, only right nodes are tallied.

6.14.2d – HeightOfTreePrivate(node\* ptr) is called by HeightOfTree() and uses the helper function DeepestLevelPrivate(node\* ptr) which recursively checks each node for the bottom of a branch and compares each branch to one another to find the deepest. The deepest is then returned to HeightOfTreePrivate(node\* ptr) and returned.

6.14.2e – DeleteAllLeavesPrivate(node\* ptr) is called by DeleteAllLeaves() and utilizes a conditional to determine if a node is a leave. If so, it deletes it and returns. Recursion is used to traverse down the left and right sides deleting leafs.

