Computer Science II: Data Structures: Programming Assignment 3

Alexander Starr – 00567613

July 29, 2013

Problem Description

The problem we are solving for this assignment is the construction of a binary search tree. Once the binary search tree is constructed, we should display it using a tree representation discussed in class, and print the inorder traversal of the tree.
The program should be able to handle both integers and strings.

Source Code and Test Results

Please see `main.cpp` for the source code of my solution.
The following are examples of input and output of the program:

```
Please choose (s)tring or (i)nteger: i
Enter the root node: 10
Enter the new node: 5
Add another? [y/n] y
Enter the new node: 15
Add another? [y/n] y
Enter the new node: 7
Add another? [y/n] y
Enter the new node: 4
Add another? [y/n] n
Your tree is represented below:
10
    5
        4
        7
    15

The inorder traversal of your tree yields:
4 5 7 10 15


Please choose (s)tring or (i)nteger: s
Enter the root node: how
Enter the new node: much
Add another? [y/n] y
Enter the new node: wood
Add another? [y/n] y
Enter the new node: could
Add another? [y/n] y
Enter the new node: a
Add another? [y/n] y
Enter the new node: woodchuck
Add another? [y/n] y
Enter the new node: chuck
Add another? [y/n] n
Your tree is represented below:
how
    could
        a
            NULL
            chuck
        NULL
    much
        NULL
        wood
```

```
            NULL
            woodchuck

The inorder traversal of your tree yields:
a chuck could how much wood woodchuck


Please choose (s)tring or (i)nteger: i
Enter the root node: 1
Enter the new node: 2
Add another? [y/n] y
Enter the new node: 3
Add another? [y/n] y
Enter the new node: 4
Add another? [y/n] y
Enter the new node: 5
Add another? [y/n] n
Your tree is represented below:
1
    NULL
    2
        NULL
        3
            NULL
            4
                NULL
                5

The inorder traversal of your tree yields:
1 2 3 4 5


Please choose (s)tring or (i)nteger: s
Enter the root node: hi
Enter the new node: hello
Add another? [y/n] y
Enter the new node: hey
Add another? [y/n] y
Enter the new node: hi
Add another? [y/n] n
Your tree is represented below:
hi
    hello
        NULL
        hey
    NULL

The inorder traversal of your tree yields:
hello hey hi
```

Comments

The program represents trees using indentation to represent the depth of the tree. Child nodes are indicated by indenting once in from the parent node. The ordering of children matters...the first child is the left child, and the second is the right child.
Duplicate nodes are not added to the search tree.