BinaryTrees1

0.1.0

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

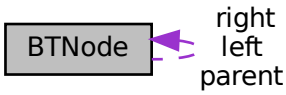
Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 BTNode Class Reference

Collaboration diagram for BTNode:



Public Member Functions

- BTNode ()
- int nodeNum ()

Public Attributes

- BTNode * left
- BTNode * right
- BTNode * parent
- int num

Static Public Attributes

- static int count = 0

3.1.1 Detailed Description

Binary Tree Node

This is from Open Data Structures in C++ by Pat Morin

Definition at line 18 of file main.cpp.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 BTNode()

```
BTNode::BTNode ( )  [inline]
```

BTNode constructor

Definition at line 29 of file main.cpp.

```
29            {
30       left = NULL;
31       right = NULL;
32       parent = NULL;
33       num = count++;
34   }
```

3.1.3 Member Function Documentation

3.1.3.1 nodeNum()

```
int BTNode::nodeNum ( )  [inline]
```

This reports the node's number

Definition at line 39 of file main.cpp.

```
39                {
40       return(num);
41   }
```

3.1.4 Member Data Documentation

3.1.4.1 count

```
int BTNode::count = 0  [static]
```

Definition at line 24 of file main.cpp.

3.1.4.2 left

```
BTNode* BTNode::left
```

Definition at line 20 of file main.cpp.

3.1.4.3 num

```
int BTNode::num
```

Definition at line 23 of file main.cpp.

3.1.4.4 parent

```
BTNode* BTNode::parent
```

Definition at line 22 of file main.cpp.

3.1.4.5 right

```
BTNode* BTNode::right
```

Definition at line 21 of file main.cpp.

The documentation for this class was generated from the following file:

- /home/drseth/CPTR227/20210224BinaryTreeStart/src/main.cpp

Chapter 4

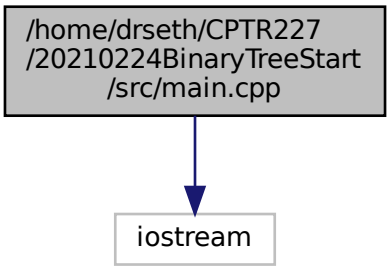
File Documentation

4.1 /home/drseth/CPTR227/20210224BinaryTreeStart/src/main.cpp File Reference

This is a demonstration of simple binary trees.

```
#include <iostream>
```

Include dependency graph for main.cpp:



Classes

- class BTNode

Functions

- int depth (BTNode *u)
- int height (BTNode *u)
- BTNode * genExampleTree (BTNode *root)
- int main (int, char **)

4.1.1 Detailed Description

This is a demonstration of simple binary trees.

This is a demo from CPTR 227 class

Author

Seth McNeill

Date

2021 February 24

4.1.2 Function Documentation

4.1.2.1 depth()

```
int depth (
            BTNode * u )
```

Calculates the depth (number of steps between node and root) of a node

Parameters

pointer	to BTNode to measure the depth of

Returns

integer count of depth

Definition at line 54 of file main.cpp.

```
54                          {
55     int d = 0; // depth counter
56     while(u != NULL) {
57         u = u->parent;
58         d++;
59     }
60     return(--d);
61 }
```

4.1.2.2 genExampleTree()

```
BTNode* genExampleTree (
            BTNode * root )
```

This generates a simple tree to play with

It is a bit of a hack.

Definition at line 84 of file main.cpp.

```
84                                          {
85     BTNode* one = new BTNode();
86     BTNode* two = new BTNode();
87     BTNode* three = new BTNode();
88     BTNode* four = new BTNode();
89     BTNode* five = new BTNode();
90     BTNode* six = new BTNode();
91     cout « "Created the nodes" « endl;
92     root->left = one;
93     cout « "Added root->left" « endl;
94     one->parent = root;
95     root->right = two;
96     two->parent = root;
97     two->left = three;
98     three->parent = two;
99     two->right = four;
100     four->parent = two;
101     one->left = five;
102     five->parent = one;
103     five->left = six;
104     six->parent = five;
105     cout « "root's number: " « root->nodeNum() « endl;
106     cout « "one's number: " « one->nodeNum() « endl;
107     cout « "two's number: " « two->nodeNum() « endl;
108     cout « "three's number: " « three->nodeNum() « endl;
109     cout « "four's number: " « four->nodeNum() « endl;
110     cout « "five's number: " « five->nodeNum() « endl;
111     cout « "six's number: " « six->nodeNum() « endl;
112     cout « "six's depth is " « depth(six) « endl;
113     cout « "root's height is " « height(root) « endl;
114     return root;
115 }
```

4.1.2.3 height()

```
int height (
              BTNode * u )
```

This calculates the height (max number of steps until leaf node)

Parameters

pointer	to a BTNode

Returns

integer count of height

Definition at line 70 of file main.cpp.

```
70                                  {
71     if (u == NULL) {
72         cout « "Reached NULL end of branch" « endl;
73         return(-1);
74     }
75     cout « "Calculating the height of node " « u->nodeNum() « endl;
76     return(1 + max(height(u->left), height(u->right)));
77 }
```

4.1.2.4 main()

```
int main (
              int ,
              char **  )
```

Definition at line 118 of file main.cpp.

```
118                       {
119     BTNode* rootNode = new BTNode(); // pointer to the root node
120     genExampleTree(rootNode);
121     cout « "Hello, world! Binary Trees\n";
122 }
```

Index