#include <iostream>

#include <cstdlib>

using namespace std;

class BinarySearchTree

{

//private:

public:

struct tree\_node

{

tree\_node\* left;

tree\_node\* right;

int data;

};

tree\_node\* root;

BinarySearchTree ()

{

root = NULL;

}

bool isEmpty() const { return root==NULL; }

void print\_inorder();

void inorder(tree\_node\*);

void insert(int);

int height (tree\_node\*);

};

// Smaller elements go left

// larger elements go right

void BinarySearchTree::insert(int d)

{

tree\_node\* t = new tree\_node;

tree\_node\* parent;

t->data = d;

t->left = NULL;

t->right = NULL;

parent = NULL;

// is this a new tree?

if(isEmpty()) root = t;

else

{

//Note: ALL insertions are as leaf nodes

tree\_node\* curr;

curr = root;

// Find the Node's parent

while(curr)

{

parent = curr;

if(t->data > curr->data) curr = curr->right;

else curr = curr->left;

}

if(t->data < parent->data)

parent->left = t;

else

parent->right = t;

}

}

void BinarySearchTree::print\_inorder()

{

inorder(root);

}

void BinarySearchTree::inorder(tree\_node\* p)

{

if(p != NULL)

{

if(p->left) inorder(p->left);

cout<<" "<<p->data<<" ";

if(p->right) inorder(p->right);

}

else return;

}

int BinarySearchTree::height(tree\_node\* count){

int l = 0;

int r = 0;

cout<<"test";

if(cout == NULL)

{

return 0;

}

{

l = height(count->left);

r = height(count->right);

if( l > r || l == r)

{

return (l + 1);

}

else

{

return (r + 1);

}

}

cout<<"left is "<<l<<endl<<"right is "<<r<<endl;

}

int main()

{

BinarySearchTree b;

int ch,tmp,tmp1;

int counter=1;

while(counter==1)

{

cout<<endl<<endl;

cout<<" Binary Search Tree Operations "<<endl;

cout<<" ----------------------------- "<<endl;

cout<<" 1. Insertion/Creation "<<endl;

cout<<" 2. In-Order Traversal "<<endl;

cout<<" 3. Exit "<<endl;

cout<<" Enter your choice : ";

cin>>ch;

switch(ch)

{

case 1 : cout<<" Enter Number to be inserted : ";

cin>>tmp;

b.insert(tmp);

break;

case 2 : cout<<endl;

cout<<" In-Order Traversal "<<endl;

cout<<" -------------------"<<endl;

b.height();

b.print\_inorder();

break;

case 3 :

system("pause");

counter=0;

break;

}

}

}