Aim-coversion in bst

#include <iostream>

using namespace std;

struct TreeNode {

int key;

TreeNode\* left;

TreeNode\* right;

TreeNode(int k) : key(k), left(nullptr), right(nullptr) {}

};

class BST {

public:

TreeNode\* insert(TreeNode\* root, int key) {

if (!root)

return new TreeNode(key);

if (key < root->key)

root->left = insert(root->left, key);

else if (key > root->key)

root->right = insert(root->right, key);

return root;

}

void inOrder(TreeNode\* root) {

if (root) {

inOrder(root->left);

cout << root->key << " ";

inOrder(root->right);

}

}

void preOrder(TreeNode\* root) {

if (root) {

cout << root->key << " ";

preOrder(root->left);

preOrder(root->right);

}

}

void postOrder(TreeNode\* root) {

if (root) {

postOrder(root->left);

postOrder(root->right);

cout << root->key << " ";

}

}

};

int main() {

BST tree;

TreeNode\* root = nullptr;

root = tree.insert(root, 50);

root = tree.insert(root, 30);

root = tree.insert(root, 20);

root = tree.insert(root, 40);

root = tree.insert(root, 70);

root = tree.insert(root, 60);

root = tree.insert(root, 80);

cout << "In-order traversal: ";

tree.inOrder(root);

cout << endl;

cout << "Pre-order traversal: ";

tree.preOrder(root);

cout << endl;

cout << "Post-order traversal: ";

tree.postOrder(root);

cout << endl;

return 0;

}