Description: https://lh4.googleusercontent.com/bpfGlrXcwL2InVVTEmmoD_RjmplIRjQ_qTei0xro8dNAn1_LWfEzxzwbA6Ph-12qiEAOhlUhNU8lPPq-x6jsbRG0r59XzrcxgbNNAZbAQvOMofzDXY8UgF2dYyKOfh8XmWHIBgv7CYrN-Z6srCJdzCEa-Cl_sPh8A67eEpwJKJBpnFroi2ANV9WYy4jU1nQkymy8v_s

**“lab 12”**

**COURSE :**

**DATA STURCTURES**

**SUBMITTED TO :**

**SIR RSHAN**

**SUBMITTED BY :**

**Rabia Batool (2021-BSE-064)**

**SECTION :**

**B**

**Data Structures and Algorithms**

**Lab 12**

# Objective

This lab session is aimed at introducing students to the ‘Tree’ data structure.

**Task 1 :**

Give answers to the following.

1.

For the given binary tree, state the following.

a.

Number of leaf nodes:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_

b.

Number of descendants of node containing 11:\_\_\_

\_\_\_\_\_\_\_

c.

Depth of the tree:

\_\_\_\_\_\_\_\_\_\_

d.

Parent node of the node containi

ng 30: \_\_\_\_\_\_\_\_\_\_\_\_

e.

Type of Binary T

ree:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

f.

Level/Depth of node containing

10:

\_

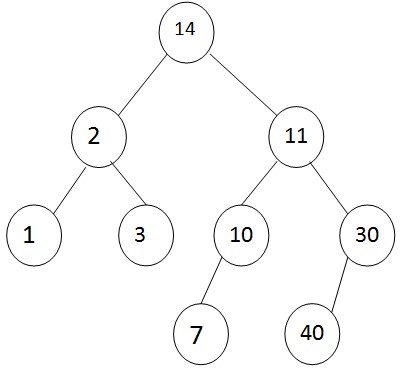
\_\_\_\_\_\_\_\_\_\_\_\_\_\_

g.

C

hildren of the root:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



|  |  |
| --- | --- |
| h. Name the ancestors of node containing 7:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  C:\Users\lenovo\AppData\Local\Temp\Rar$DIa9688.4282\WhatsApp Image 2023-12-30 at 8.14.53 PM.jpeg | |
| 2. Traverse the binary tree given above in pre, post and inorder.     1. Preorder Traversal: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_      1. Post Traversal: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_      1. In-order Traversal: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   C:\Users\lenovo\AppData\Local\Temp\Rar$DIa9688.34687\WhatsApp Image 2023-12-30 at 8.15.28 PM.jpeg | |
| 3. | Draw the expression tree of the given algebraic expression and traverse the tree in pre, post and inorder.    **(a+b\*c)+((d\*e+f)\*g)**  C:\Users\lenovo\AppData\Local\Temp\Rar$DIa9688.3471\WhatsApp Image 2023-12-30 at 8.15.47 PM.jpeg |

# Code Task

|  |
| --- |
| Complete the given class to implement a binary search tree.  struct Node  {  Node \*left, \*right; int data;  Node()  {  left=right=NULL;  }    };  class bst  {  Node \*root; public:  bst()  {  root=NULL;  }    bool isempty();  void insert(int item);// if already exist do not insert bool search(int item);  void Preorder(node \* ptr) void Postorder(node \* ptr) void Inorder(node \* ptr)      }; |

# Code :

// lab 11.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

struct Node

{

Node \*left, \*right;

int data;

Node()

{

left=right=NULL;

}

};

class bst

{

Node \*root;

public:

bst()

{

root=NULL;

}

bool isempty()

{if(root==NULL)

return true;

else

return false;}

void insert(int item)

{Node \*s1=root;

Node\*s2=0;

while (s1!=0)

{s2=s1;

if(item< s1->data)

s1=s1->left;

else if(item> s1->data)

s1=s1->right;

else

cout<<"valuue already exist\n";

}

Node\* temp=new Node;

temp->data=item;

temp->left=0;

temp->right=0;

if(s2==0)

{root=temp;

cout<<temp->data<<endl;}

else if(item<s2->data)

{s2->left=temp;

cout<<temp->data<<endl;}

else

{s2->right=temp;

cout<<temp->data<<endl;}

}

bool search(int item) {

Node\* ptr = root;

bool found = false;

while (ptr != NULL && !found) {

if (item < ptr->data) {

ptr = ptr->left;

} else if (item > ptr->data) {

ptr = ptr->right;

} else if (item == ptr->data) {

found = true;

cout << "Value found: " << ptr->data << endl;

} else {

cout << "Unexpected case!" << endl;

}

}

return found;

}

void Preorder(Node \* ptr)

{

if(ptr!=NULL)

{

cout << ptr->data<<endl;

Preorder(ptr->left);

Preorder(ptr->right);

}

}

void Postorder(Node \* ptr)

{

if(ptr!=NULL)

{

Postorder(ptr->left);

Postorder(ptr->right);

cout << ptr->data <<endl;

}

}

void Inorder(Node \* ptr)

{

if(ptr!=NULL)

{

Inorder(ptr->left);

cout << ptr->data<<endl;

Inorder(ptr->right);

}

}

Node\* get()

{Node \*a=root;

return a;}

};

int \_tmain(int argc, \_TCHAR\* argv[])

{bool a;

bst t;

cout<<"enpty tree"<<endl;

a=t.isempty();

cout<<a<<endl;

cout<<"nodes that are inserted:\n";

t.insert(16);

t.insert(15);

t.insert(19);

t.insert(9);

t.insert(66);

t.insert(8);

t.insert(12);

cout<<"if 1 node is found if zero node is not found"<<endl;

t.insert(61);

cout<<"node that is searched:\n";

cout<<t.search(45)<<endl;

Node\*b=t.get();

cout<<"preorder:\n";

t.Preorder(b);

cout<<"ineorder:\n";

t.Inorder(b);

cout<<"prostorder:\n";

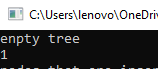
t.Postorder(b);

system("pause");

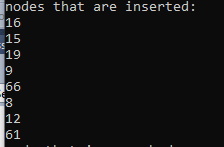
return 0;

}

# Empty tree:



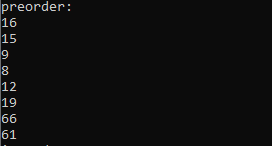
# Node to be inserted:



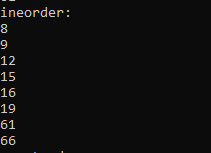
# Searched nodes:



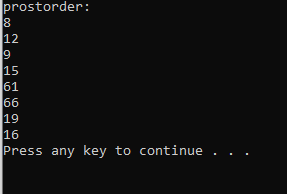
# Pre order:



# Inorder :



# Postorder:



# Output:

