Lab_Practice_week5

Name: Heng Sovannrach

Exercise1 :

```cpp
#include <iostream>
using namespace std;

class node {
    public:
    int data;
    node *left;
    node *right;
    node (int newData){
        data = newData;
        left = nullptr;
        right = nullptr;
    }
};
class Tree{
    public:
    node *root;
    Tree(){
        root = nullptr;
    }
};
int main() {}
```

Exercise2 :

```cpp
#include <iostream>
using namespace std;

class node {
    public:
    int data;
    node *left;
    node *right;
    node (int newData){
        data = newData;
        left = nullptr;
        right = nullptr;
    }
};
class Tree{
    public:
    node *root;
    Tree(){
        root = nullptr;
    }
    node *insert(node *root, int data ){
        if(root == nullptr){
            return new node(data);
        } else if (data < root->data){
            root->left = insert(root->left , data);
        } else if (data > root->data){
            root->right = insert(root->right ,data);
        }
        return root;
    }
    void inOrder(node *root){
        if(root!=nullptr){
            inOrder(root->left);
            cout<<root->data<< " ";
            inOrder(root->right);
        }

    }
```

```
39      void preOrder(node *root){
40          if(root !=nullptr){
41              cout<<root->data<< " ";
42              preOrder(root->left);
43              preOrder(root->right);
44          }
45      }
46      void PostOrder(node *root){
47          if(root !=nullptr){
48              PostOrder(root->left);
49              PostOrder(root->right);
50              cout<<root->data<< " ";
51          }
52      }
53      void add(int val){
54          root = insert(root , val);
55      }
56      void displayInOrder(){
57          cout<<"InOrder:";
58          inOrder(root);
59          cout<<endl;
60      }
61      void displayPreOrder(){
62          cout<<"PreOrder :";
63          preOrder(root);
64          cout<<endl;
65      }
66      void displayPostOrder(){
67          cout<<"Post-Order:";
68          PostOrder(root);
69          cout<<endl;
70      }
71  };
```

```
72  int main () {
73      Tree tree ;
74      tree.add(50);
75      tree.add(17);
76      tree.add(72);
77      tree.add(12);
78      tree.add(23);
79      tree.add(54);
80      tree.add(76);
81      tree.add(9);
82      tree.add(14);
83      tree.add(19);
84      tree.add(67);
85      tree.displayInOrder();
86      tree.displayPreOrder();
87      tree.displayPostOrder();
88  }
```

Output :

```
OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   PROBLEMS

PS C:\Users\USER\Desktop\c++>  & 'c:\Users\USER\.vscode\extensions\ms-vsc
MIEngine-Out-iqhrddl1.wnh' '--stderr=Microsoft-MIEngine-Error-zbthemt0.rsi
InOrder:9 12 14 17 19 23 50 54 67 72 76
PreOrder :50 17 12 9 14 23 19 72 54 67 76
Post-Order:9 14 12 19 23 17 67 54 76 72 50
PS C:\Users\USER\Desktop\c++>
```

Exercise 3 :

Data-Structure-and-Algorithms > week6 > lab-practice > Exercise-3.cpp > main()

```cpp
#include <iostream>
#include <string.h>
using namespace std;
class node {
    public:
    string name;
    node *left;
    node *right;
    node (string s){
        name = s;
        left = nullptr;
        right = nullptr;
    }
};
class Tree {
    public:
    node *root;
    Tree () {
        root = nullptr;
    }
    node *insert(node *root , string name){
        if(root==nullptr){
            return new node(name);
        } else if(name < root->name){
            root->left = insert(root->left, name);
        } else if (name > root->name) {
            root->right = insert(root->right, name);
        }
        return root;
    }
    }
    void showStudents() {
        cout << "Students In A-Z order :";
        inorder(root);
    }
};
int main() {
    Tree student;

    // Testing BST with student names
    student.add("Reach");
    student.add("Chesda");
    student.add("Vichka");
    student.add("Aob");

    student.showStudents();

    return 0;
}
```

Output :

```
OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  PROBLEMS

PS C:\Users\USER\Desktop\c++>  & 'c:\Users\USER\.vscode\extensio
ft-MIEngine-In-abl50v1a.w5e' '--stdout=Microsoft-MIEngine-Out-k2
.gft' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Students In A-Z order :Aob  Chesda  Reach  Vichka
PS C:\Users\USER\Desktop\c++>
```

Exercise 4 :

Data-Structure-and-Algorithms > week6 > lab-practice > Exercise-4.cpp > main()

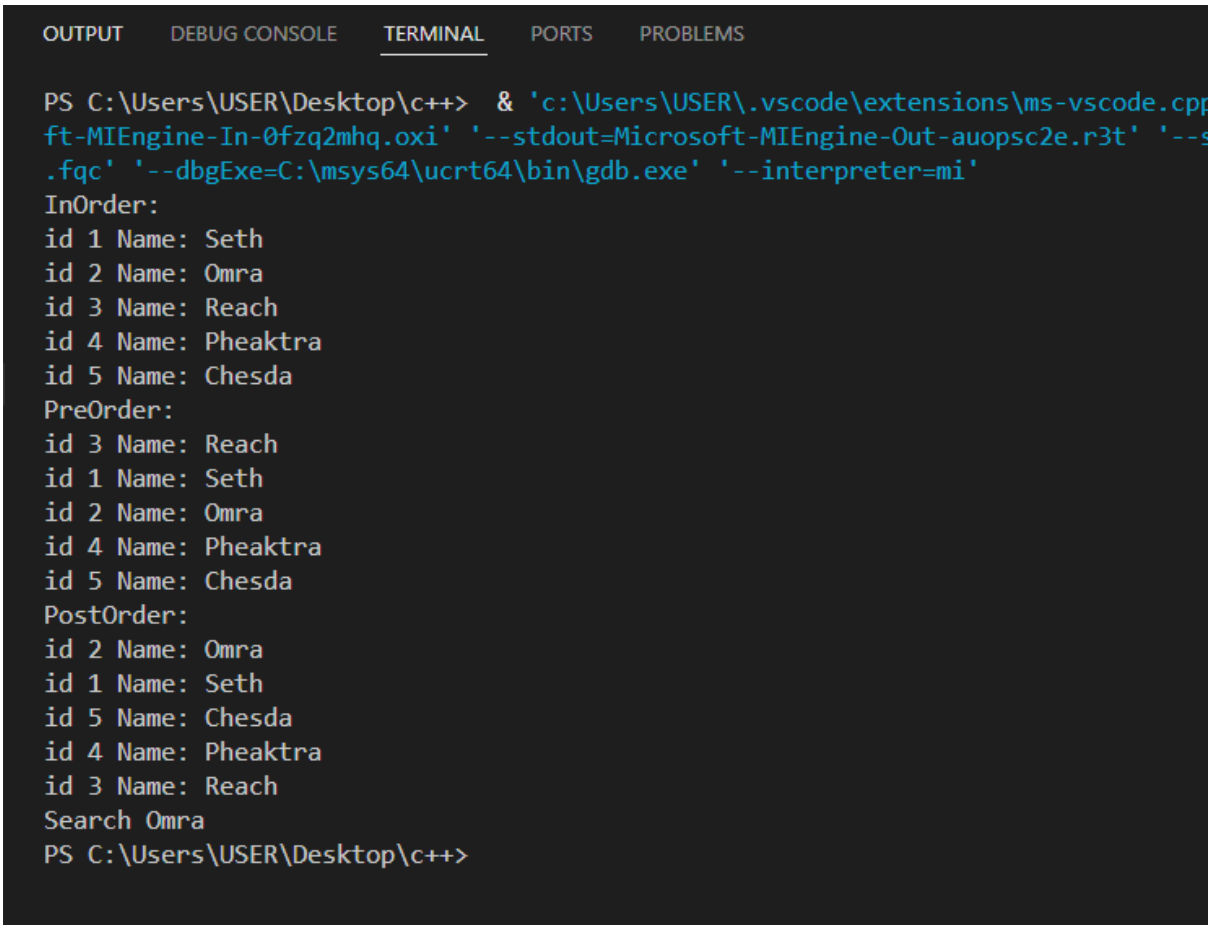
```cpp
#include<iostream>
using namespace std;
struct Customer
{
    int id;
    string name;

};
class node{
public:
    Customer* data;
    node* right;
    node* left;
    node(int id, string name){
        data = new Customer();
        data->name = name;
        data->id = id;
        right = left = nullptr;
    }
};
class Tree{
private:
    node* root;
public:
    Tree(){
        root = nullptr;
    }
    node* insertNode(node* root, Customer s){
        if (root == nullptr) {
            return new node(s.id, s.name);
        }
        if(s.id<root->data->id){
            root->left = insertNode(root->left,s);
        }
        if(s.id > root->data->id){
            root->right = insertNode(root->right, s);
        }
        return root;
    }
    void insert(Customer a){
        root = insertNode(root,a);
    }
```

```
43      void inOrder(node* root){
44          if(root==nullptr) return;
45          inOrder(root->left);
46          cout<<"id "<<root->data->id <<" Name: "<< root->data->name<<endl;
47          inOrder(root->right);
48      }
49      void preOrder(node* root){
50          if(root==nullptr) return;
51          cout<<"id "<<root->data->id <<" Name: "<< root->data->name<<endl;
52          preOrder(root->left);
53          preOrder(root->right);
54      }
55      void postOrder(node* root){
56          if(root==nullptr) return;
57          postOrder(root->left);
58          postOrder(root->right);
59          cout<<"id "<<root->data->id <<" Name: "<< root->data->name<<endl;
60      }
61      void displayPreOrder(){
62          preOrder(root);
63      }
64      void displayInOrder(){
65          inOrder(root);
66      }
67      void displayPo    node *Tree::root
68          postOrder(root);
69      }
70      node* searchNode(node* root, int id){
71      if(root==nullptr){
72          return nullptr;
73      }
74      if (id == root->data->id) {
75          return root;
76      }
77      else if (id < root->data->id) {
78          return searchNode(root->left, id);
79      }
80      else {
81          return searchNode(root->right, id);
82      }
83  }
```

```
84
85  node* search(int id) {
86      return searchNode(root, id);
87  }
88  };
89
90  int main(){
91      Tree tree;
92      Customer c1 = {3, "Reach"};
93      Customer c2 = {1, "Seth"};
94      Customer c3 = {4, "Pheaktra"};
95      Customer c4 = {2, "Omra"};
96      Customer c5 = {5, "Chesda"};
97
98      tree.insert(c1);
99      tree.insert(c2);
100     tree.insert(c3);
101     tree.insert(c4);
102     tree.insert(c5);
103     cout << "InOrder: ";
104     cout<<endl;
105     tree.displayInOrder();
106     cout << "PreOrder: ";
107     cout<<endl;
108     tree.displayPreOrder();
109     cout << "PostOrder: ";
110     cout<<endl;
111     tree.displayPostOrder();
112     cout<<"Search "<< tree.search(2)->data->name<<endl;
113     return 0;
114 }
```

Output :


OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PROBLEMS
PS C:\Users\USER\Desktop\c++> & 'c:\Users\USER\.vscode\extensions\ms-vscode.cpp
ft-MIEngine-In-0fzq2mhq.oxi' '--stdout=Microsoft-MIEngine-Out-auopsc2e.r3t' '--s
.fqc' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
InOrder:
id 1 Name: Seth
id 2 Name: Omra
id 3 Name: Reach
id 4 Name: Pheaktra
id 5 Name: Chesda
PreOrder:
id 3 Name: Reach
id 1 Name: Seth
id 2 Name: Omra
id 4 Name: Pheaktra
id 5 Name: Chesda
PostOrder:
id 2 Name: Omra
id 1 Name: Seth
id 5 Name: Chesda
id 4 Name: Pheaktra
id 3 Name: Reach
Search Omra
PS C:\Users\USER\Desktop\c++>