BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Munshi nagar, Andheri (W) ,Mumbai - 400058
DEPARTMENT OF MASTER OF COMPUTER APPLICATION

Name: Snehal Jayprakash Borji
UID: 2023510008
Course: F.Y.M.C.A.
Subject: Data Structure

Practical 05

Aim: Create a tree data structure which will hold employee records. write functions for insertion and traversal of tree.

Code :

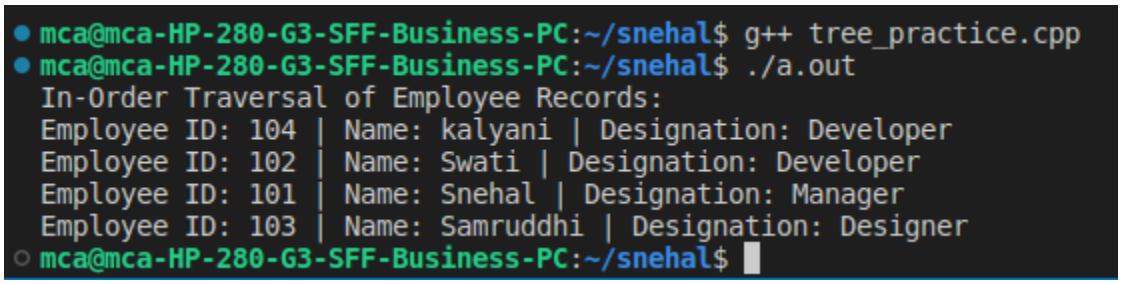
```
#include <iostream>

#include <string>

using namespace std;



struct Employee {

   int empID;

   string ename;

   string desg;

};



class EmployeeTree {

public:
```

```
    EmployeeTree(int capacity) {
        maxCapacity = capacity;
        tree = new Employee[capacity];
        for (int i = 0; i < capacity; i++) {
            tree[i].empID = -1;
        }
        size = 0;
    }

// function to insert employee in tree
    void insert(Employee employee) {
        if (size < maxCapacity) {
            tree[size++] = employee;
        } else {
            cout << "Tree is full. Cannot insert more employees." <<
endl;
        }
    }

    // function to perform traversal
    void inOrderTraversal() {
```

```
        inOrderTraversalRecursive(0);
    }

private:
    Employee* tree;
    int maxCapacity;
    int size;

    void inOrderTraversalRecursive(int index) {
        if (index < size) {
            inOrderTraversalRecursive(2 * index + 1); // Left child
            if (tree[index].empID != -1) {
                cout << "Employee ID: " << tree[index].empID << " |
Name: " << tree[index].ename << " | Designation: " <<
tree[index].desg << endl;
            }
            inOrderTraversalRecursive(2 * index + 2); // Right child
        }
    }
};
```

```
int main() {
    int maxEmployees = 10;
    EmployeeTree tree(maxEmployees);

    Employee emp1 = {101, "Snehal", "Manager"};
    Employee emp2 = {102, "Swati", "Developer"};
    Employee emp3 = {103, "Samruddhi", "Designer"};
    Employee emp4 = {104, "kalyani", "Developer"};

    tree.insert(emp1);
    tree.insert(emp2);
    tree.insert(emp3);
    tree.insert(emp4);

    cout << "In-Order Traversal of Employee Records:" << endl;
    tree.inOrderTraversal();

    return 0;}
```

OUTPUT :


mca@mca-HP-280-G3-SFF-Business-PC:~/snehal$ g++ tree_practice.cpp
mca@mca-HP-280-G3-SFF-Business-PC:~/snehal$ ./a.out
In-Order Traversal of Employee Records:
Employee ID: 104 | Name: kalyani | Designation: Developer
Employee ID: 102 | Name: Swati | Designation: Developer
Employee ID: 101 | Name: Snehal | Designation: Manager
Employee ID: 103 | Name: Samruddhi | Designation: Designer
mca@mca-HP-280-G3-SFF-Business-PC:~/snehal$