

**LAB-10**

**Title: Algorithm to implement tree traversal techniques**

**Name: Azizul Abedin Azmi**

**ID: 2022-1-60-130**

**Section: 03**

**Course Code: CSE207**

**Course Title: (Data Structures)**

**Date: 13/05/2024**

**Course Instructor:**

**Dr. Anup Kumar Paul**

**Associate Professor**

**Department of Computer Science and Engineering**

**Source Code:**

**TreeTraversal.java:**

package Lab10;

public class TreeTraversal {

    // Inorder Traversal

    public static void inorder(Node root) {

        if (root != null) {

            inorder(root.lchild);

            System.out.print(root.data + " ");

            inorder(root.rchild);

        }

    }

    // Preorder Traversal

    public static void preorder(Node root) {

        if (root != null) {

            System.out.print(root.data + " ");

            preorder(root.lchild);

            preorder(root.rchild);

        }

    }

    // Postorder Traversal

    public static void postorder(Node root) {

        if (root != null) {

            postorder(root.lchild);

            postorder(root.rchild);

            System.out.print(root.data + " ");

        }

    }

}

**Node.java:**

package Lab10;

public class Node {

    int data;

    Node lchild;

    Node rchild;

    public Node(int data) {

        this.data = data;

        this.lchild = null;

        this.rchild = null;

    }

}

**Main.java:**

package Lab10;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Manual input of tree data

        System.out.println("Enter the root value:");

        int rootValue = scanner.nextInt();

        Node root = new Node(rootValue);

        System.out.println("Enter the left child value of the root (or -1 if none):");

        int leftValue = scanner.nextInt();

        if (leftValue != -1) {

            root.lchild = new Node(leftValue);

        }

        System.out.println("Enter the right child value of the root (or -1 if none):");

        int rightValue = scanner.nextInt();

        if (rightValue != -1) {

            root.rchild = new Node(rightValue);

        }

        // Inorder Traversal

        System.out.println("Inorder Traversal:");

        TreeTraversal.inorder(root);

        System.out.println();

        // Preorder Traversal

        System.out.println("Preorder Traversal:");

        TreeTraversal.preorder(root);

        System.out.println();

        // Postorder Traversal

        System.out.println("Postorder Traversal:");

        TreeTraversal.postorder(root);

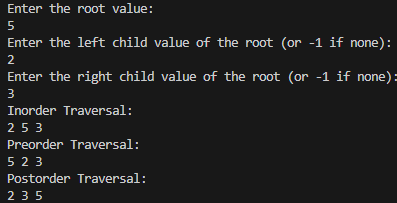
        System.out.println();

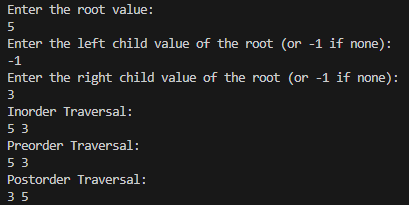
        scanner.close();

    }

}

**Output:**

****

****