Practical

#Implement traversal of binary tree.  
1>preorder  
2>inorder  
3>postorder

**Code:**

// Tree traversal in Java

class Node

{

int item;

Node left, right;

public Node(int key)

{

item = key;

left = right = null;

}

}

class BinaryTree

{

// Root of Binary Tree

Node root;

BinaryTree()

{

root = null;

}

void postorder(Node node)

{

if (node == null)

return;

// Traverse left

postorder(node.left);

// Traverse right

postorder(node.right);

// Traverse root

System.out.print(node.item + " ");

}

void inorder(Node node)

{

if (node == null)

return;

// Traverse left

inorder(node.left);

// Traverse root

System.out.print(node.item + " ");

// Traverse right

inorder(node.right);

}

void preorder(Node node)

{

if (node == null)

return;

// Traverse root

System.out.print(node.item + " ");

// Traverse left

preorder(node.left);

// Traverse right

preorder(node.right);

}

public static void main(String[] args)

{

BinaryTree tree = new BinaryTree();

tree.root = new Node(1);

tree.root.left = new Node(12);

tree.root.right = new Node(9);

tree.root.left.left = new Node(5);

tree.root.left.right = new Node(6);

System.out.println("Inorder traversal");

tree.inorder(tree.root);

System.out.println("\nPreorder traversal ");

tree.preorder(tree.root);

System.out.println("\nPostorder traversal");

tree.postorder(tree.root);

}

}

**Output:**

