Assignment 0

TREES

```
import java.util.*;
class Node
{
        int data;
        Node left,right;

        Node(int d){
          data=d;
        }
}
class BinaryTree {
   Node root;
   BinaryTree()
   {
                root=null;
   }
    void create(Node n)
   {
        Node temp=root;
     Scanner sc=new Scanner(System.in);

        if(root==null)
                root=n;
```

```
else {

        while(temp!=null)
        {
          System.out.println("Enter Direction : 1.left 2.right");
          char d=sc.next().charAt(0);

          if(d=='l')
          {
            if(temp.left==null) {
               temp.left=n;
                break;
        }
             else {
              temp=temp.left;
            }
          }
          else if(d=='r') {
                    if(temp.right==null) {
               temp.right=n;
               break;
               }
              else {
               temp=temp.right;
              }
```

```
                }
            }//close while
        }
}//end create

void printInorder(Node root) {

    if(root==null)
    return;

    printInorder(root.left);
    System.out.print(root.data+" ");
    printInorder(root.right);
}

void printPreorder(Node root) {

    if(root==null)
    return;

    System.out.print(root.data+" ");
    printPreorder(root.left);
    printPreorder(root.right);
}

void printPostorder(Node root) {
```

```
        if(root==null)
                return;

        printPostorder(root.left);
        printPostorder(root.right);
        System.out.print(root.data+" ");
}

void printLevelOrder() {

        int h=height(root);

        for(int i=1;i<=h;i++){
                printCurrentLevel(root,i);
        }

}

int height(Node root) {

        if(root==null)
                return 0;

  else {
```

```
        int lh=height(root.left);
        int rh=height(root.right);

        if(lh>rh)
        return (lh+1);

        else
        return (rh+1);
   }
}

void printCurrentLevel(Node root, int level) {

      if(root==null)
      return;

      if(level==1)
            System.out.print(root.data+" ");


      if(level>1)
      {
            printCurrentLevel(root.left,level-1);
            printCurrentLevel(root.right,level-1);
      }
}
```

```
}
public class Main{
   public static void main(String[] args) {

       Scanner s=new Scanner(System.in);

         BinaryTree bt=new BinaryTree();
         int ch=1;

        System.out.println("Create a binary tree.");
        while(ch==1)
        {
          System.out.println("Enter node data : ");
                int d=s.nextInt();
                Node n=new Node(d);
                bt.create(n);
                System.out.println("Press 1 to continue adding nodes");

                ch=s.nextInt();
                }

        System.out.println("Inorder traversal:");
          bt.printInorder(bt.root);

        System.out.println("\nPreorder traversal:");
          bt.printPreorder(bt.root);
```

```
        System.out.println("\nPostorder traversal:");
          bt.printPostorder(bt.root);

        System.out.println("\nLevel order traversal of binary tree:");
          bt.printLevelOrder();
    }
  }
```

OUTPUT

```
Create a binary tree.
Enter node data :
1
Press 1 to continue adding nodes
1
Enter node data :
2
Enter Direction : 1.left 2.right
l
Press 1 to continue adding nodes
1
Enter node data :
3
Enter Direction : 1.left 2.right
r
Press 1 to continue adding nodes
1
```

Enter node data :

4

Enter Direction : 1.left 2.right

l

Enter Direction : 1.left 2.right

l

Press 1 to continue adding nodes

1

Enter node data :

5

Enter Direction : 1.left 2.right

r

Enter Direction : 1.left 2.right

r

Press 1 to continue adding nodes

2

Inorder traversal:

4 2 1 3 5

Preorder traversal:

1 2 4 3 5

Postorder traversal:

4 2 5 3 1

Level order traversal of binary tree:

1 2 3 4 5

...Program finished with exit code 0