```
14          if(root == null){
15              root = newNode;
16              return;
17          }
18
19          TreeNode current = root;
20
21          while(current != null){
22              if(value < current.data){
23                  if(current.leftChild == null){
24                      current.leftChild = newNode;
25                      return;
26                  }
27                  current = current.leftChild;
28              }
29              else{
30                  if(current.rightChild == null){
31                      current.rightChild = newNode;
32                      return;
33                  }
34                  current = current.rightChild;
35              }
36          }
```

```java
    }

    public void displayTree(TreeNode node, int depth) {
        if (node == null) {
            return;
        }
        displayTree(node.leftChild, depth + 1);
        for (int i = 0; i < depth; i++) {
            System.out.print(s: " ");
        }
        System.out.println(node.data);
        displayTree(node.rightChild, depth + 1);
    }

    Run | Debug
    public static void main(String[] args){
        UniqueTree uniqueBinaryTree = new UniqueTree();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.print(s: "Enter a number: ");
            int value = scanner.nextInt();
            if (value == 0) {
                break;
            uniqueBinaryTree.insertNode(value);
            uniqueBinaryTree.displayTree(uniqueBinaryTree.root, depth: 0);
        }
    }

    public class TreeNode {
        int data;
        TreeNode leftChild;
        TreeNode rightChild;

        public TreeNode(int value) {
            this.data = value;
            this.leftChild = null;
            this.rightChild = null;
        }
    }
}
```

UniqueTree.java 1, U

Lab1Submission > UniqueTree.java > ...

```java
import java.util.Scanner;

public class UniqueTree {

    public TreeNode root;

    public UniqueTree(){
        this.root = null;
    }

    public void insertNode(int value) {
        TreeNode newNode = new TreeNode(value);

        if(root == null){
            root = newNode;
            return;
        }

        TreeNode current = root;

        while(current != null){
            if(value < current.data){
                if(current.leftChild == null){
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL GITLENS Code - Java

```
(base) sethstephens@Seths-MBP Java % cd "/Users/sethstephens/Desktop/Programming-Fundamentals/Lab01/Java/" && javac UniqueTree.java && java UniqueTree
Enter a number: 5
5
Enter a number: 3
 3
5
Enter a number: 1
  1
 3
5
(base) sethstephens@Seths-MBP Java % cd "/Users/sethstephens/Desktop/Programming-Fundamentals/Lab1Submission/" && javac UniqueTree.java && java UniqueTree
Enter a number: 5
5
Enter a number: 4
 4
5
Enter a number: 3
  3
 4
5
Enter a number: 1
   1
  3
 4
5
Enter a number: 6
   1
  3
 4
5
 6
Enter a number: 4
   1
  3
 4
  4
5
 6
Enter a number:
```