Data Structures and Algorithms

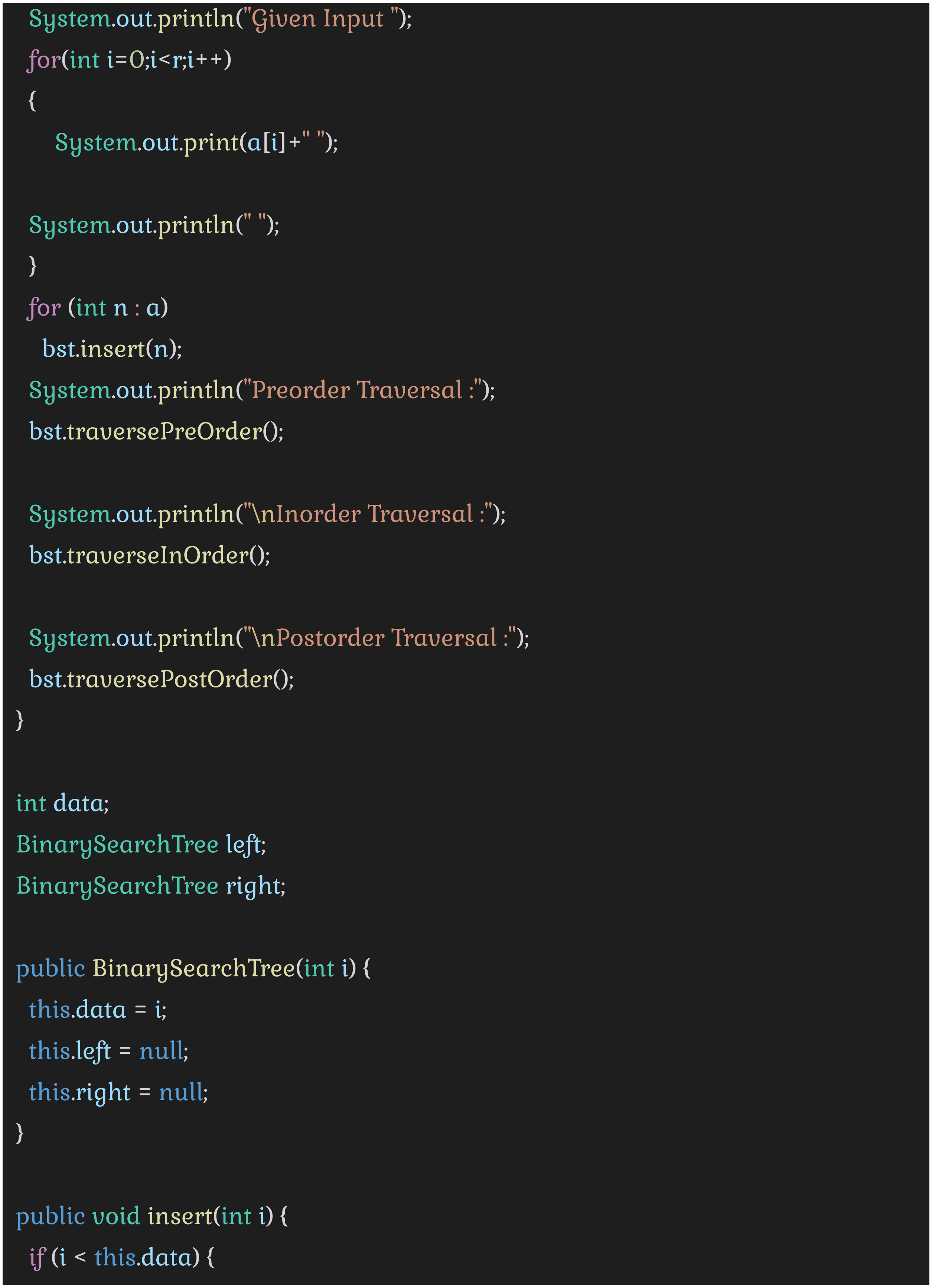
CSE2001

Lab - 3 - Assignment - 1

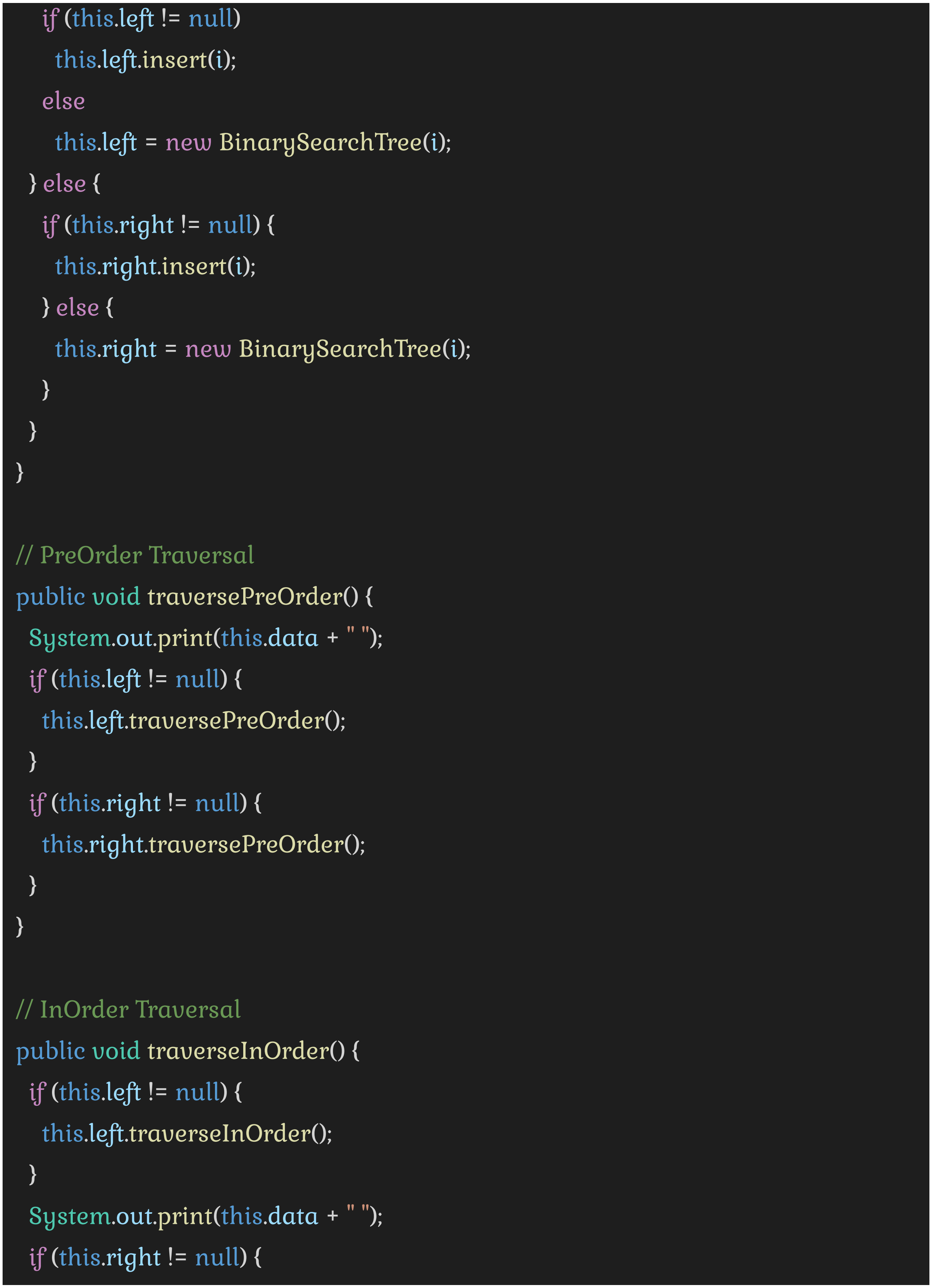
Yashwanth Reddy
19BCE7362
Date- 6thJuly2021

Problem : Binary Tree to Inorder, Post-Order and Pre-Order

```
import java.util.*;

public class BinarySearchTree {

 public static void main(String[] args) {

  Scanner sc = new Scanner(System.in);

  BinarySearchTree bst = new BinarySearchTree(36);

  System.out.println("Enter Number of nodes");
  int r = sc.nextInt();
  int[] a = new int[r];
  System.out.println("Enter the nodes to Insert ")
  for(int i=0;i<r;i++)
  {
    a[i]=sc.nextInt();

  }
```

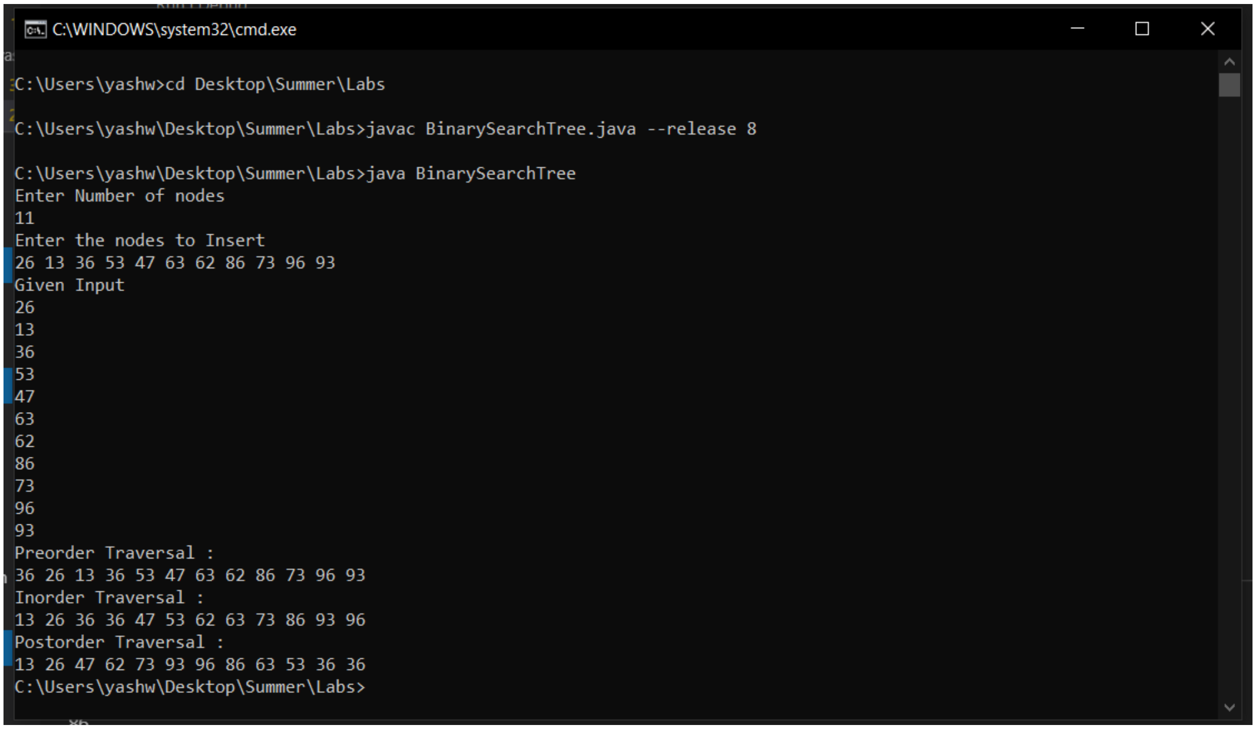


```
  System.out.println("Given Input ");
  for(int i=0;i<r;i++)
  {
      System.out.print(a[i]+" ");

  System.out.println(" ");
  }
  for (int n : a)
    bst.insert(n);
  System.out.println("Preorder Traversal :");
  bst.traversePreOrder();

  System.out.println("\nInorder Traversal :");
  bst.traverseInOrder();

  System.out.println("\nPostorder Traversal :");
  bst.traversePostOrder();
}

int data;
BinarySearchTree left;
BinarySearchTree right;

public BinarySearchTree(int i) {
  this.data = i;
  this.left = null;
  this.right = null;
}

public void insert(int i) {
  if (i < this.data) {
```



```
    if (this.left != null)
      this.left.insert(i);
    else
      this.left = new BinarySearchTree(i);
  } else {
    if (this.right != null) {
      this.right.insert(i);
    } else {
      this.right = new BinarySearchTree(i);
    }
  }
}

// PreOrder Traversal
public void traversePreOrder() {
  System.out.print(this.data + " ");
  if (this.left != null) {
    this.left.traversePreOrder();
  }
  if (this.right != null) {
    this.right.traversePreOrder();
  }
}

// InOrder Traversal
public void traverseInOrder() {
  if (this.left != null) {
    this.left.traverseInOrder();
  }
  System.out.print(this.data + " ");
  if (this.right != null) {
      this.right.traverseInOrder();
    }
  }
  // PostOrder Traversal
  public void traversePostOrder() {
    if (this.left != null) {
      this.left.traversePostOrder();
    }
    if (this.right != null) {
      this.right.traversePostOrder();
    }
    System.out.print(this.data + " ");
  }
}
```

Output



```
C:\Users\yashw>cd Desktop\Summer\Labs

C:\Users\yashw\Desktop\Summer\Labs>javac BinarySearchTree.java --release 8

C:\Users\yashw\Desktop\Summer\Labs>java BinarySearchTree
Enter Number of nodes
11
Enter the nodes to Insert
26 13 36 53 47 63 62 86 73 96 93
Given Input
26
13
36
53
47
63
62
86
73
96
93
Preorder Traversal :
36 26 13 36 53 47 63 62 86 73 96 93
Inorder Traversal :
13 26 36 36 47 53 62 63 73 86 93 96
Postorder Traversal :
13 26 47 62 73 93 96 86 63 53 36 36
C:\Users\yashw\Desktop\Summer\Labs>
```