

## COMP 282 Project 1

**NOTE:** THE ASSIGNMENTS WILL BE GRADED USING AUTOMATED SCRIPTS SO PLEASE FOLLOW THE SUBMISSION GUIDELINES PROVIDED. PLEASE GO THROUGH THIS ENTIRE DOCUMENT BEFORE CODING AND SUBMITTING THE ASSIGNMENT.

**Problem:** Complete the code to implement a BINARY SEARCH TREE and TRAVERSAL METHODS

Use the classes and interface provided in Project1.zip



Project1.zip

The Project1.zip contains BST.java, TreeInterface.java, TreeNode.java, Test.java and input.txt

**ADD YOUR CODE** to below methods in BST.java

- public boolean search(E e) //TODO
- public void insert(E e) //TODO
- public boolean delete(E e) //TODO **Note: User inorder successor to replace the deleted node**
- public ArrayList<E> postorderNoRecursion() //TODO **Note: Do not use recursion**
- public int getNumberOfNonLeaves() //TODO
- public ArrayList<E> inorderNoRecursion() //TODO **Note: Do not use recursion**

**Note: The generic type E in class BST extends Comparable<E>**

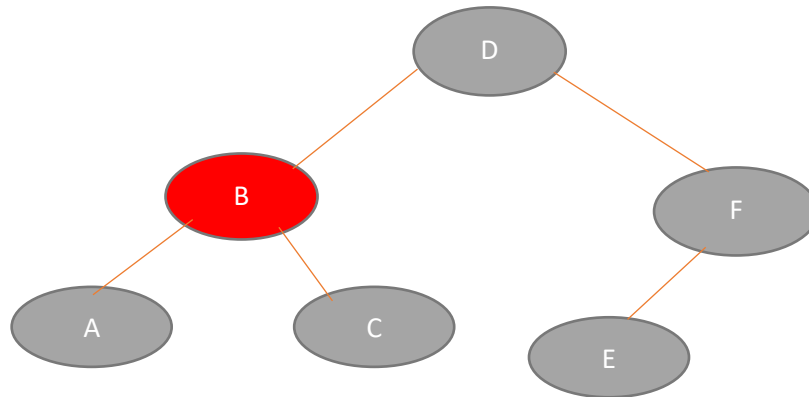
**DO NOT MODIFY ANY OTHER METHODS OR CLASS.**

Sample input provided in input.txt file

EXPLANATION in COMMENTS FOR INPUT in input.txt file BELOW

```
INS D          //INSERT NODE WITH STRING VALUE "D" IN THE TREE
INS F          //INSERT NODE WITH STRING VALUE "F" IN THE TREE
INS B          //INSERT NODE WITH STRING VALUE "B" IN THE TREE
INS A          //INSERT NODE WITH STRING VALUE "A" IN THE TREE
INS C          //INSERT NODE WITH STRING VALUE "C" IN THE TREE
INS E          //INSERT NODE WITH STRING VALUE "E" IN THE TREE
SEARCH F       //SEARCH NODE WITH STRING VALUE "F" IN TREE
DEL B          //DELETE NODE WITH STRING VALUE "B" FROM THE TREE
COUNT NON_LEAF //COUNT NON LEAF NODE IN TREE
GET NON_REC_IN  //GET INORDER TRAVERSAL OF TREE WITHOUT RECURSION
GET NON_REC_POST // GET POSTORDER TRAVERSAL OF TREE WITHOUT RECURSION
```

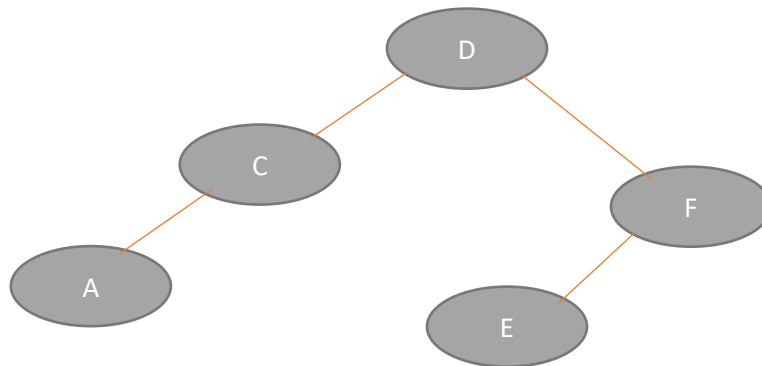
The tree constructed from the above input will be as below: The strings are compared lexicographically  
i.e.  $A < B < C < D < E < F$



Inorder Successor of "B" is "C"

Inorder predecessor of "B" is "A"

On deleting "B" from the tree the resulting tree will be as follows: Note: "B" is replaced with the inorder successor of "B" i.e "C"



The output for the above input in input.txt file should be as below:

true

true

3

A C D E F

A C E F D

**NOTE:** The code should be tested using Test.java class provided in project2.java. DO NOT WRITE ANY METHOD TO PRINT THE OUTPUT. Test.java will call the appropriate method to print the output.

ECLIPSE IDE: If you are using any IDE the input file should be read from path "src/input.txt". If you are using java command from terminal or command prompt read file using only file name "input.txt".

**SUBMISSION DETAILS:** Assignment will be graded using automated scripts so please make sure you follow the below mentioned submission guidelines

- 1) Create a new folder for your group. Example for group 1 create a directory/folder called Group1.
- 2) Copy all the five BST.java, TreeInterface.java, TreeNode.java, Test.java and input.txt in your group directory.
- 3) Make sure you set the input file path in Test.java as "input.txt" as the assignment will be graded through terminal. i.e. below code should be used to read the input file in Test.java

```
File file = new File("input.txt")
```

- 4) **Java version 8** should be used for compiling and executing the code.
- 5) Your code should now compile from command prompt or terminal using **javac \*.java**
- 6) Run your code using **java Test** and check the output in terminal or command prompt
- 7) If everything is working fine zip the directory for your group created in step 1. Example for group 1 a zip archive Group1.zip will be created.
- 8) Submit Group1.zip archive in canvas.