



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

DEPARTMENT OF COMPUTER SCIENCE

COS212: PRACTICAL 3 - FRIDAY

RELEASE: THURSDAY 2 MARCH 2017, 22:00
DEADLINE: FRIDAY 3 MARCH 2017, 22:00

Instructions

Complete the tasks below. Certain classes have been provided for you in the *files* folder of the practical download. You have been given a main file which will test some code functionality, but it is by no means intended to provide extensive test coverage. You are encouraged to edit this file and test your code more thoroughly. Remember to test “corner” cases. Upload **only** the given source files with your changes in a zip archive before the deadline. Please comment your name **and** student number in at the top of each file.

Task 1: Binary Search Trees [25]

Tree traversal (also known as tree search) is a form of graph traversal and refers to the process of visiting (checking and/or updating) each node in a tree data structure, exactly once. Such traversals are classified by the order in which the nodes are visited

You have been provided with a fully implemented threaded tree in the file `BST.java`. Your task is to implement the following methods:

`public Node mirrorTree(Node root)` Write a Java method that takes a reference to the root node of a binary tree and creates a new tree (with its own nodes) that is the mirror image of the original tree. For example: if root is a reference to the root of the tree on the left below, then the return value of `mirror(root)` would be a reference to the root of the tree on the right below.

Hint: This method is much easier to write if you use recursion.

`public void printInorder()` and `private void printInOrderRec(Node currRoot)`: Traverse the Original tree and display the output along side it's Mirror with

`System.out.println();` to print your output as depicted in `main.java` provided as expected output.

`public void printPreorder()` and `private void printPreOrderRec(Node currRoot)`: Traverse the Original tree and display the output along side it's Mirror with

`System.out.println();` to print your output as depicted in `main.java` provided as expected output.

`public void printPostorder()` and `private void printPostOrderRec(Node currRoot)`: Traverse the Original tree and display the output along side it's Mirror with

`System.out.println();` to print your output as depicted in `main.java` provided as expected output.

Note: You can use only one function or method to implement In-order, Pre-Order and Post-Order. But should have same names as stated above

`(printInorder(), printPreorder() and printPostorder())`

Submission

Submit your source files on the CS Website. Place all the files in a zip archive named as uXXXXXXXX.zip where XXXXXXXX is your student number. Please make use of the submission slot that corresponds to your practical session (Practical 3 Tuesday). Submit your work before the deadline. No late submissions will be accepted.