Project 1: Binary Tree Printing

Due date: Friday July 28, 11:55PM EST.

### You may discuss any of the assignments with your classmates and tutors (or anyone else) but all work for all assignments must be entirely your own . Any sharing or copying of assignments will be considered cheating (this includes posting of partial or complete solutions on any public forum). If you get significant help from anyone, you should acknowledge it in your submission (and your grade will be proportional to the part that you completed on your own). You are responsible for every line in your program: you need to know what it does and why. You should not use any data structures and features of Java that have not been covered in class (or the prerequisite class). If you have doubts whether or not you are allowed to use certain structures, just ask your instructor.

In this project you will populate a Binary Search Tree (BST) and print out its contents. For example, a tree built from the input: DBACGHJK will produce an output similar to the tree below:

D

/ \

B G

/ \ \

A C H

\

J

\

K

Notice the tree display is not exact , as shown by the placement of the node “C” . The goal of the exercise to produce a reasonable facsimile of the tree. Since we are limited by output to the console line, the actual display can not be a perfect graphical representation of the tree. Three sample inputs and their respective outputs are provided for you to duplicate or improve upon.

# Objectives

The goal of this programming project is for you to master (or at least get practice on) the following tasks:

* understanding recursion and / or iterative calls within a BST
* debugging and checking results
* Analyzing a problem to determine the best approach
* working with existing code

**Start early!** This project may not seem like much coding, but debugging always takes time. Analyze and plan now so questions are not being asked a day before the due date.

# Helpful code:

# The main class that will call methods the BST class are provided. This class (ProgProject3.java) defines the inputs, creates the BST and calls the print method (which you will write), in order to display the tree. You should use this code “as is”, and put your work into the printTree method of the BinarySearchTree class.

## One Approach

Your solution should be able to print all the nodes of the tree, without any collisions (nodes printing over other nodes), and display each node on its appropriate level and traversal order. Some arrows (\ or /) should be included to help read your tree. Arrows do **NOT** have to start at one node and end at another node. This is not possible within the structure of using the Console line. View the expected outputs below to see the type of output required.

You are welcome to use any approach, however, I provide my approach below to assist you.

Print Tree:

Initialize a 2 dimension array that hold the image or your output. You should only need a 100 by 100 array.  
 call PostArray with the root, and some initial positioning information  
 Print the 2D array. In the print loop, ignore blank rows and columns

PostArray

Perform some type of traversal (Preorder, Inorder, Postorder) that will visit a left and right subtrees, and the root.  
 For the left and right subtrees, recursively call this method  
 The method adds the root info (the input letter) to the array as well as / \  
 In my approach I passed the row and column for the root of this subtree, and returned the width of the tree for this method. This information was useful in order to maintain where the next node needed to be placed in the 2D array.  
  
I viewed the output as a 2D array where every cell was one character to print out.

## Expected Results:

: Tree for input: DBACGHJK

D

/ \

B G

/ \ \

A C H

\

J

\

K

Tree for input: DACBEFMLGHJK

D

/ \

A E

\ \

C F

/ \

B M

/

L

/

G

\

H

\

J

\

K

Tree for input: JABCDEFISRQPON

J

/ \

A S

\ /

B R

\ /

C Q

\ /

D P

\ /

E O

\ /

F N

\

I

## Running the program

The zipped java project file, which contains all your source code in your Eclipse project. This code should include all the code in the attached file, with the BST class have the additional printTree method and other supporting methods. Provide the zip file of the entire Java project provided to you. If I need to reconstruct your project because you only provided source files, you will lose 20% of the project points. If this is not clear to you , see me before or after class.

Since you will be using the ProgProject3 class provided, running your program should be produce a series of tree input statement and their representation using the printTree method.

## Working on This Assignment

You should start right away! My effort was about 2 hours, and the total lines of code approximately 50. Your solution may be longer, or perhaps shorter. You should plan on spending at least 4-8 hours on the project.

**Grading Criteria – 20 points**

1. (5) Program provide some display of all the nodes
2. (5) Nodes are displayed in the proper order and level
3. (5) Results are totally accurate. , meaning all nodes are in the right order and shown on the correct level. There are not overlaps of nodes
4. (5) Display is compact (no large amounts of unnecessary white space). Three of your sample input demonstrating your approach handles many cases. These inputs are additional elements of the inputs array within the ProgProject3 class. All nodes are unique, no duplicate keys.

***Remember arrows between nodes DO NOT have to be complete.***