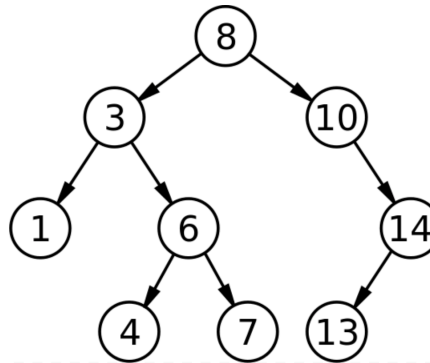


CSCD 320 Homework Two –Second choice

This is a programming assignment. Javadoc Standard Documentation *NOT* Required -- basic documentation is (all source files must include author name and a description of the class(es) contained in the file; any code another developer might not understand at first look should also be documented). Also, follow Java's naming conventions, utilize whitespace/indentation liberally in your code.

Requirements

- 1) As we learned in class, please implement a Binary Search Tree(BST) and its nested tree Node class, as well as their constructors. Data type in the tree Node is Object.
- 2) As we learned in class, please implement the insert() method that performs insertion into the tree.
- 3) As we learned in class, please implement printAllPath() method for the tree.
- 4) Implement a Tester class to test your BST implementation. In the Tester class, instantiate a BST object myBST, then using insert() method to insert data items into myBST so that it will contain the data shown as below.



Hint: thinking about in what order you will insert the sequence of these integers to form this tree in the main memory?

5)After you created completely myBST object following step 4, you call myBST.printAllPath() method. You are supposed to see the results below are displayed on the standard output, if your implementation is correct.

path 1: 8->3->1

path 2: 8->3->6->4

path 3: 8->3->6->7

path 4: 8->10->14->13

6)At the bottom of your Tester.java file, please use java comments to briefly explain why you can get these results from step 5? Hint: explain as if you are the computer that executes the code step by step?

Java inline comment format is shown in the below, each line starting with double slashes.

```
// this is a java code comments  
// This is explains blab ..  
//.....
```

Please be noted that printAllPath() method should be able to work for all kinds of binary trees, not limited to BST tree. Again, here it is easier to construct a BST tree in memory, therefore we use BST tree to practice the printAllPath() method.

BONUS

If you can solve the same problem of printAllPath(), by using a **stack** in an iterative program (not recursive), you get 10 extra bonus points. Please clearly tell the grader on top of your Tester file, that you have done the bonus part.

To Turn In:

Turn in your solution on the **EWU Canvas** by going to CSCD320-01 course page on Canvas, then clicking Assignments→hw2->submit.

This assignment must be submitted in working order by the due data on the top of this assignment. Submit a zip file with your source files only. Source files are those that end in .java. Check your zip file before submitting and make sure it has only your source files. Do not submit .class files; they do the grader no good. You will not receive credit if your submission contains only .class files. Name your zip file with your last name, followed by the first initial of your first name, followed by hw2. For example, if you are John Smith, name you file as smithjhw2.zip.

The grader should be able to open your zip file, compile your code, and run your program from **the command line using javac and java** version 1.6 or greater (so make sure you try this yourself before you submit).

Get started right away on this assignment or you WILL NOT finish :-(

If your code has compile-time error when USING COMMAND LINE tool to compile, you get a ZERO for this homework! That is, we use the command javac *.java to compile your source code and use java Tester to run your program.