Last name First name	
class	cs3100
Assignment name/number	hwk4
Date submitted	“2014/05/02
Cs- login	
How many hours did your program since semester start	

This is type T2 hwks. Read on my website how to submit hwks. Ask me if something is not clear. Read this document carefully twice from start to end before doing any programming. All programs in this course use OOP. Read the programs on help 1500 and help2500 and help3100 and write some of those programs by yourself. Use 7SP.

For all hwk:
You must have all the classes, attributes, and methods I ask for. But, you can make additional attributes and methods if you feel you need them. No additional classes.
If I give you name of a class, signature of a method, or name of a variable, you must use the name exactly. Signature of method means it's name and type of any arguments.
If I let you select the arguments for your method, I will say doFunction(…). You can then pick any arguments you want or have no arguments if you choose so.
Similarly for return type, sometimes I will specify, sometimes I will let you choose.

In this course we may sometimes specify packages. If I don't specify package you use no package. (That is called in Java using default package.)

You will be graded on following the given requirements, technical elegance of your solution, satisfaction of requirements, program style and correctness.

Look at program ProgramStyle2500.java that does not do much but shows the program style we will be using in this course.

What is unspecified you can choose.

Any questions … ask in the class if you think it may benefit others, if it is specific to you – you can use the email or my office hours.
Hint:
Start early.

Do not submit data files, I will use my own.

Program 1:

Write a program. It will have two classes. Node and BSTRec (the main class). The two classes will be in no package. (aka default package)

Write class Node for BST.
It will look like usual BST Node and will have data of type int.

Class BSTRec, in no package.
It will have data root to reference the root node.
It will NOT have any variable counting nodes.

It has a method:
insertR1(..., int val), insertR2(...) //will insert element val; it will reject duplicates
countTreeR1(....), countTreeR2(....) //will compute how many nodes you have in the tree.
deleteR1(...., int val), deleteR2(...) //will delete element val
printLDRR1(), printLDRR2() //will print the tree in LDR order on a single line with 2 spaces between values
Ex: >33 44 55
addR1(...), addR2(...) //this will add 1 to all values in the tree.
Whatever else you need.

There will be data file data.txt. It will have the following format. Data will be integer. Fields will be delimited by blank. The data.txt will have the correct format as specified, no need for defensive programming here.
ins 77
ins 65
ins 128
printLDR
del 65
printLDR
add
countNodes //and print

In main.
Do whatever you need to do.
Read the data.txt do the operations it lists.