**CS 248 – Object-Oriented Programming and Data Structures**

**HW7: 100 points**

**Objective:**

The objective of this homework is to learn about:

1. Develop knowledge regarding different operations of binary search tree (BST)
2. Learn and implement different operations and functions in BST

**Problem Statement:** In class, we have learned about the process of deleting a node in BST. Here is a summary of that:

**1) Node to be deleted is leaf:** Simply remove that node from the tree

50 50

/ \ delete(20) / \

30 70 ---------> 30 70

/ \ / \ \ / \

20 40 60 80 40 60 80

**2) Node to be deleted has only one child:** Copy the child to the parent node and delete the child

50 50

/ \ delete(30) / \

30 70 ---------> 40 70

\ / \ / \

40 60 80 60 80

**3) Node to be deleted has two children:**Find inorder successor of the node. Copy contents of the inorder successor to the node and delete the inorder successor. Note that inorder predecessor can also be used.

50 60

/ \ delete(50) / \

40 70 ---------> 40 70

/ \ \

60 80 80

**Problem Statement:** Complete the delete function for BST by implementing the functions given in the starter code. Submit the BST.java file only through Canvas.

**Sample input/output:**

(following is the content of the file. It shows 11 inserts followed by 3 delete operations)

11

43

31

20

40

28

33

64

56

89

47

59

3

43

20

89

After all insertions:

43 31 64 20 40 56 89 28 33 47 59

After every delete operation:

47 31 64 20 40 56 89 28 33 59

47 31 64 28 40 56 89 33 59

47 31 64 28 40 56 33 59

**(This sample output is generated by replacing the node to be deleted by the minimum value to the right subtree)**

**HW Grading:** (100 points).

1. 5% - Follows style guidelines, including header and correct filename.
2. 10% - No warnings
3. 30% - Compiles without errors
4. 55% - Correct implementation of all functions