**King Fahd University of Petroleum & Minerals**

**College of Computer Science and Engineering**

**Information and Computer Science Department**

**ICS 202 – Data Structures**

# Lab07 - Binary Search Trees

**Objectives**

The objectives of this lab is to design, implement and use binary search trees. This lab is a continuation of the previous lab work.

**Notes**

For the purpose of this lab, you may download the attached programs.

**Lab Exercises**

**NOTE: For each of the tasks you are only allowed to use the classes given in this lab.**

1. **Given a binary (search) tree, design and implement the following methods in the class BST<T>.** [Your methods should not invoke other methods, except helper methods if needed]

(a) **public String getPath(T t1)**

This method finds and returns the path from the root to a given node (value) t1. This should be recursive.

(b) **public int getRightLeafCount()**

This method finds and returns the number of leaves that are right children. This should be recursive. For example, if your given binary tree is:

|  |  |
| --- | --- |
|  D / \ B F / \ \A C H  | * getPath("C")

 This will return "D B C" which is the path from the root to a given node t1, in this case “C”.* getRightLeafCount() returns 2
 |

2. Write a recursive instance method: **public static int rangeCounter(T t1, T t2)** in the BST class that counts the number of elements that are greater than t1 and less than t2 in the BST. Assume that the invoking BST object has distinct keys, and t1 < t2 (t1.compareTo(t2) < 0).

3. Write the following methods:

**public T successor(T t1)** that finds and returns the successor of a given node.

**public T predecessor(T t1)** that finds and returns the predecessor of a given node t1.

**Note**:

* The successor of a node x is the node that is visited just after node x is visited in an inorder traversal.
* The predecessor of a node x is the node that is visited just before node x is visited in an inorder traversal.

4. Use the given test class to test for these methods.