**CSc 2720 - Data Structures: Lab 7**

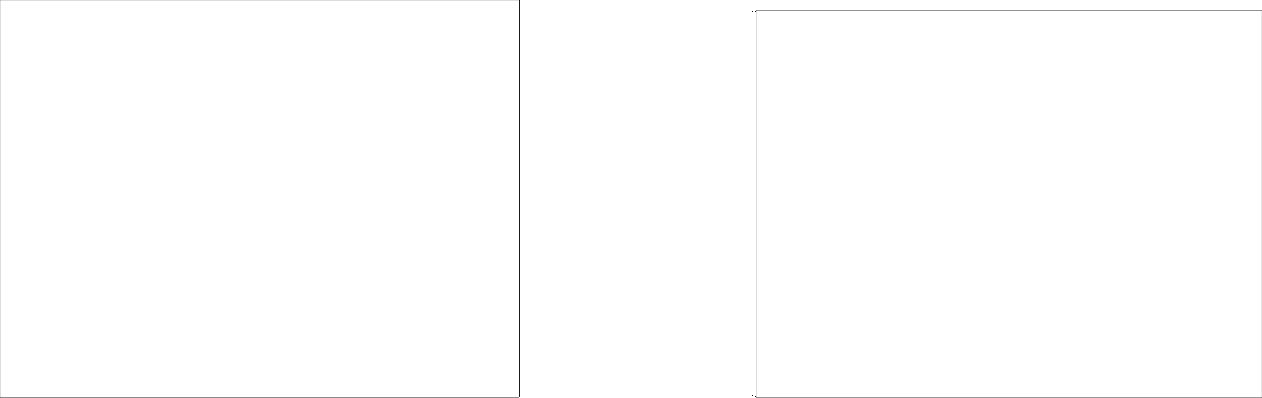
**Deadline to Submit: [2/26/2021] [11:00pm] ET(US)**

Failure to submit will result in a zero for this lab.

**#Note : Helper Code in next page.**

**Problem: [100 points]**

1. Import the java project BinaryTrees to your IDE and create a new class called **TreeTraversal.java**
2. Create the binary tree tree2 (tree1 already exists in the code template provided).



tree1 tree2

1. Write a function called checkSkipped that takes a tree as an argument traverses it in an Inorder traversal and prints out the traversed nodes. The function should return true if there is a skipped alpha character in the sequence and false if the tree contains a perfect sequence of alphabets.

**For example:**

**Input:** tree2

**Output:** False

**Explanation:** The inorder traversal is ABCDEFGHI and there is no skiped character

**Input:** tree1

**Output:** True

**Explanation:** The inorder traversal is BDEFGHXY and there are many skiped characters.Between B and D there is no C and there are many missing characters between H and X.



-------------------------------------- Class Template --------------------------------------------

**public class** TreeTraversal {

**public static void** main(String[]args){

* Create a tree called tree1 BinaryTree<Character> tree1 = **new** BinaryTree("H");

BinaryTree<Character> rightsubtree1 = **new** BinaryTree("X"); rightsubtree1.attachRight('Y');

BinaryTree<Character> leftsubtree1 = **new** BinaryTree("D"); leftsubtree1.attachLeft('B');

BinaryTree<Character> leftsubtree2 = **new** BinaryTree("F"); leftsubtree2.attachRight('G'); leftsubtree2.attachLeft('E');

leftsubtree1.attachRightSubtree(leftsubtree2); tree1.attachLeftSubtree(leftsubtree1); tree1.attachRightSubtree(rightsubtree1);

BinaryTree<Character> tree2 = **new** BinaryTree("F");

//BUILD tree2 HERE

System.***out***.println(*checkSkipped*(tree1)); // Should return true System.***out***.println(*checkSkipped*(tree2)); // Should return false

}

**public static boolean** checkSkipped(BinaryTree<Character>tree){

/\* To Iterate a tree using Inorder Traversal

TreeIterator<Character> iter = new TreeIterator<Character>(tree);

iter.setInorder();

while (iter.hasNext()){

System.out.print (iter.next());

}

INSERT CODE HERE

\*/

**return false**;

}

}