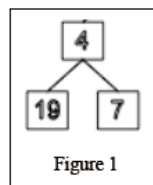


# CSCI 1112 Algorithms and Data Structures

## Lab 10 - Trees

### Part 1: Binary trees (10 points)

- Create a class called `BTNode`, which represents a node in a binary tree. The class should have the following private fields: a data element, a left child, and a right child.
- Add getter and setter methods for the three private fields.
- Add a constructor for the `BTNode` class. The constructor should take arguments for the data value, as well as the left and right children of the node.
- Write a Boolean method `isLeaf()` which returns true if the node is a leaf, false otherwise.
- Add two recursive methods, `getLeftMost()` and `getRightMost()`. These methods should return the data values of the left-most and right-most leaves in the tree, respectively. (The code for `getLeftMost()` is given in the slides)
- Add three recursive printing methods: `preorderPrint()`, `inorderPrint()`, and `postorderPrint()`. These methods should print the values of the nodes using the corresponding tree traversal techniques.
- Add a main method to test your `BTNode` class by constructing the tree in figure 1.



### Part 2: Binary search trees (10 points)

- Create a Java class called `BSTNode` which extends `BTNode`. Implement `search(int k)` and `insertValue(int k)` methods for this class.
- In the main method of your Test class, create a `BSTNode` object with data value 14. This is the root of the BST. Add the remaining values in figure 2 using the `insertValue()` method. Note that the order of the nodes in your binary search tree will be different from the structure shown in the figure. Output the values of the tree using the inorder printing method.

