

Learning Objectives:

- Deepen your understanding of BinarySearchTrees
- Gain experience using recursion
- Write junit tests to unit test your code
- Understand preorder, postorder and inorder traversal

Turning In:

Make sure to use the same names and signatures that are specified in the instructions.

Submit a jar file that **includes all the java files** via Canvas.

Description:

Use Sedgewick's code in BST.java as a startingpoint for this assignment.

Do NOT make any changes to the existing code in BST.java except as described below.

Add three methods: **preOrder**, **inOrder**, and **postOrder** and the corresponding junit tests (use a test folder)

In addition write a public class called BstPrinter.java. It includes a method **printBST** that prints the binary tree

3 methods: preOrder inOrder postOrder

Learn about the different ways to traverse a binary tree (<https://www.youtube.com/watch?v=gm8DUJJhmY4>)

Then implement the three depth-first traversals.

- `public Iterable<Key> preOrder()`
- `public Iterable<Key> inOrder()`
- `public Iterable<Key> postOrder()`

All three methods are public and return a value of type `Iterable<Key>`

In case that the tree is empty all three methods should behave like the method **levelOrder**

Write junit tests to test the new methods. There should be multiple tests for each of the three traversal methods. They should test representatives from the different equivalent partitions.

A class: BstPrinter:

Class BstPrinter has 2 public methods: `printBST` and `main`.

```
public static void printBST(BST<Integer, ? > tree)
```

It prints a BinarySearchTree, whose keys are of type Integer.

You can assume that each of the keys is less than 100.

That means that each key can be formatted to 2 digits.

If a number is less than 10 leading zeros should be added

Note: printBST needs to be able to print BST of any size.

The output should look like shown on the right.

It grows and shrinks proportional to the size of the tree.

Class BstPrinter also includes a main method to test the `printBST`.

