CST 8130 Data Structures

Lab Exercise #5

Purpose demonstrate use of:

- multiple classes organized in multiple packages (industry best practice)
- a Binary Search Tree including example operations/features

Task Using sample code provided, code additional methods to add functionality

---

Detailed requirements:

- You are provided the sample code for the following classes:
  - BinaryTreeNode
  - BinaryTree
  - BinaryTreeTest

- Your tasks are to:
  - Create an array to store these int values:

```
26 38 34 33 70 11 25 72 14 51
13 77  6 40 95 84 50 35 31 54
88 74 46 86 57 39 85 80 19 92
```

  - Start with an empty Binary Search Tree (BST) and use the array above along with a loop to insert the values into the BST.
    Do not use dozens of lines containing insertInTree() !!

  - Display the contents of the BST using in-order and pre-order traversal
    - Hint: use the sample in-order traversal method
    - and write your own method to perform pre-order traversal
    - Your output must match the ***Sample Output***

  - Create a method using **iteration** to search for a specific value following this method signature: `public boolean search(int key)`
    - Test your method using the values seen in the ***Sample Output.***

        - Your results and format of output must match the ***Sample Output***
    - Notes:
        - You may alter main()
        - And you may make minor changes to the sample code.

Sample Output: see .jpg from on Brightspace

Coding & Submission Requirements:

- follow the Java Coding Conventions as your "style guide"
- create at least 2 classes
    - name of the class with the main() method must end in "Test"
- create at least 2 packages
- export your Eclipse **project** as a .zip
    - name your file Lab5_Lastname_Firstname.Lab.zip
    - also, be sure to name your Eclipse project according to the form: Lab5_Lastname_Firstname
    - (substitute your own first and last names, of course)

The *Submission Requirements* have been **updated** since your previous *Assignment* & *Labs* !!

Note:

- You will lose marks if you do not:
    - follow the instructions under ***Detailed Requirements***
        - including any algorithms described
    - meet the ***Coding requirements***
    - ensure there are complete Javadoc comments for all public constructors and methods.
    - produce the same output as the ***Sample output***