

# CSCI 230 Homework 6

## Spring 2017

Collaboration: None

Due Date: 9:30/10:30 am Apr 5

Total Points: 100

## 1 Source Code

The Java classes provided in the zip file attached to this Dropbox assignment are:

`BinarySearchTree.java`

Java classes reused (with no modifications) from homework assignment 5 are:

`BinaryNode.java`

`EmptyBSTException.java`

`NullBinaryNodeException.java`

`DuplicateElementException.java`

Under no circumstances are you allowed to modify or create new *BinaryNode*, *DuplicateElementException*, *NullBinaryNodeException*, or *EmptyBSTException* classes. You must use these files **as is**.

You **may only** modify the *BinarySearchTree* class. In particular, in this class you may **only** modify the methods listed in Part 1, and under no circumstances are you allowed to remove, add, or modify any other line of code in this class (this include instance variables, class variables, constants, etc.).

Lastly, you **may not** change the package structure! Specifically, `csci230.hwk6` cannot be removed or modified. If a solution is submitted with a different package structure, it will not be graded, no exceptions.

## 2 Part 1

In the *BinarySearchTree* class please fully implement the methods listed below:

```
public void delete(AnyType element) throws EmptyBSTException, NullBinaryNodeException
private void delete(BinaryNode<AnyType> node, AnyType element)
public String postOrder() throws EmptyBSTException
private String postOrder(BinaryNode<AnyType> node)
public String preOrder() throws EmptyBSTException
private String preOrder(BinaryNode<AnyType> node)
public String inOrder() throws EmptyBSTException
private String inOrder(BinaryNode<AnyType> node)
```

In each method you will see a TODO comment - this is where you add your code. In the provided source code, numerous comments are given; please ensure you read them carefully. Additionally, I posted the BST slides on OAKS, and sections 12.3 in your course textbook discusses BSTs including its operations.

## 3 Part 2

The provided *BinarySearchTree* class has a main method. In the main please add test cases that demonstrate you have fully evaluated the operational correctness of the methods you implemented in Part 1. To receive full credit, these test cases must be included.

## 4 Submission

Create a zip file that only includes the completed `BinarySearchTree.java` file. The name of the zip file must be your last name in lower case. For example, `ritchie.zip` would be correct if the original co-developer of UNIX (Dennis Ritchie) submitted the assignment. Only assignments submitted in the correct format will be accepted (no exceptions). Please submit the zip file (via OAKS) to the Dropbox setup for this assignment by the due date. You may resubmit the zip file as many times as you like, Dropbox will only keep the newest submission.

Per the syllabus, late assignments will not be accepted — no exceptions. Please do not email Jefferson or I your assignment after the due date, we will not accept it.

## 5 Grading Rubric

For programs that at least compile and run, the following rubric will be used:

<i>BinarySearchTree</i> Compiles	10 points
<i>BinarySearchTree</i> Runs	5 points
Thoroughness of your test cases	5 points
Instructor test cases (8 cases 10 points each)	80 points
TOTAL	100 points

For programs that don't compile you could lose 10 to 100 points depending on the number and severity of the syntax errors. For programs that compile but either don't run or posses runtime errors, you could lose 15 to 90 points, depending on the number and severity of the runtime errors. Your test cases are only worth 5 points; however, if you don't thoroughly test your code, you will likely lose many of the 80 points on the instructor test cases. In short:

- make sure your code compiles and runs
- follow the directions in this handout
- **test your code**