Project #6
CS 2210 – Fall 2022
Christopher LaFave

I. Requirements: Create methods to a Binary Search Tree with Sibling nodes and methods to traverse that tree with pre-order, in-order, and post-order methods.

II. Design: The binary tree methods are just creating the implementation of BinaryTree.java, but I added a posToSTNode() method to help me convert between the Position abstraction and STNodes.
For the walks, I only had to implement each respective visit method and the computeResult method.

III. Security Analysis: I did not do very much exception handling and I only tested it on the fill-tree method, so if the elements are other than numbers or if somebody makes a crafty tree it could mess with my program in weird ways.

IV. Implementation: The implementation of ListBinaryTree was just turning the pos into an STNode and then doing the right operations to it.
The implementation of the walks was similarly simple. All I did was override the performTour() method and comment out the visits that didn't apply to the respective walk, override the respective visit method to print to result.nodeResult, and the computeResult() method to add up the results correctly for each different walk.

V. Testing: The only testing I did was on the fillTree() method.

VI. Summary/Conclusion: My program runs properly, gets the right answers, and follows all of the instructions.

VII. Lessons Learned: I learned a lot more about tree walking, how to deal with abstractions in Java code, how trees work, and how to implements abstract classes.