# Programming Assignment 5

## Instructions

Write a program that allows the user to enter and search for strings. New strings are added to the tree in their correct position using alphabetical sorting (If the word “Ball” is at the root, the word “Apple” would go left, but the word “Orange” would go right. When strings are added to the tree, they should be wrapped inside a node object that holds the string, the frequency (number of times) with which that string has been added to the tree, and references to two other nodes (children). The strings must be stored in a Binary Search Tree. You will need to implement your own Binary Search Tree (you may use the one in zybooks). A driver has been provided for you (`TreeDemo.java`). You **must** use that class as your driver, without alterations. Be sure to analyze the driver code (specifically the methods and manner in which the methods are called) to be sure that your program functions properly.

In addition to the provided `TreeDemo` class, your project will also need two other classes:

1. BinarySearchTree – This will serve as your container class. It needs to have functionality for adding nodes and searching (traversing) the tree.
2. Node – Your Binary Search Tree will be made up of Node objects. Each node object must reference two children and also contain the string entered by the user and the frequency of that string.

**To submit your program, zip all files together and upload to Canvas.**

The rubric below will be used in the grading of your program. Partial points may be awarded for each category.

| **Category** | **Value** |
|---|---|
| Program is free of errors and is well-commented | **/ 5** |
| Node class contains references to two children, the string, and the frequency of that word | **/ 5** |
| Successfully implement and instantiate a Binary Search Tree that holds Node objects | **/ 5** |
| New strings are successfully added to the tree in their correct position using alphabetical sorting. (If the word “Ball” is at the root, the word “Apple” would go left, but the word “Orange” would go right.). Strings already in the tree should not be added again; instead, the frequency should be increased | **/ 25** |
| Program correctly searches for and displays a node | **/ 15** |
| Program correctly uses Pre-Order traversal to print all the nodes in the tree | **/ 15** |
| Program correctly uses In-Order traversal to print all the nodes in the tree | **/ 15** |
| Program correctly uses Post-Order traversal to print all the nodes in the tree | **/ 15** |
| **Total Points** | **/ 100** |