

# ENSF 594 – Principles of Software Development II

Summer 2021

Lab Assignment # 5

Binary Search Tree

**Total Marks: 25**

**Due Date: Wednesday, Aug 11, by 11:59 PM using D2L**

You are given an input ASCII text file (input.txt) containing an arbitrary number of student records in the following format:

Operation code: 1 character ('I' for insert)

Student number: 7 characters

Student last name: 25 characters

Home department: 4 characters

Program: 4 characters

Year: 1 character

Each record is stored as one line in the text file; i.e. there is a newline character immediately following the year. Create a Java program that does the following:

Create a Java program that does the following:

1. Build a binary search tree using the data from the input file. The tree should be ordered by student last name (use a case insensitive comparison). There are only unique records in the input file. Each node must contain the student data (exclude the operation code), a left child pointer, and a right child pointer. Submit a picture of the binary search tree named BST.jpeg.
2. Traverse the binary search tree recursively, printing out the nodes in ascending logical order; i.e. do a depth-first, in-order tree traversal. Print the node data to a text file named (output1.txt).
3. Traverse the binary search tree, starting at the top level (the root node), proceeding downwards level-by-level. At each level print out the nodes from left to right. In other words, do a breadth-first traversal. You may have to use a queue to implement this. Print the node data to a text file. (output2.txt)

Be sure to use proper headings and fixed-width columns for both of the output files.

Create your own input files to test your code. Start with a file that specifies insertions of data into the tree. An official input file is available on D2L; use this file to create your output files.

You must program all the data structures for this assignment from scratch; in other words, do not use any classes from the Java libraries to implement the binary search tree or queue. Also draw a picture of the binary search tree, as it appears after processing all the insertions specified in the input file. Within each node, only show the data that is used to order the tree. The picture can be hand drawn, or you may use a graphics application to create it.

### Bonus (Optional)

Create a second version of your program, called Lab05b, which uses an AVL tree to store the input data. This version of the program should do the appropriate rotations when inserting data into the tree. Only insertions into the tree need to be handled (i.e. use an input file that specifies only insertions, and no deletions)

### Submit electronically via D2L:

1. Your source code files. Your TA will run your program to verify that it works correctly. Make sure your Java program compiles and runs without issues.
2. The two output files named output1.txt and output2.txt
3. Your picture of the Binary Search Tree

### Grading Rubric:

**Functionality:** produces correct output:

- Correct Binary Search Tree generated (8 marks)
- In-order traversal (9 marks)
- Breadth First Traversal (8 marks)