

Lab 4 – due Wednesday, 9/28 at 11pm

Symbol tables

1. (30 pts) Starting with an empty Binary Search Tree, insert each letter (don't worry about the value) into the tree. The Letters are V E R Y E A S Y Q U E S T I O N. Submit all the steps together (in order) in a pdf into gradescope. You cannot just show the final tree – you must show each step.
2. (70 points) Create a generic binary search tree data structure and a driver to test it (can be in same file as data structure). You can get this data structure from the internet, the book or write your own. Write a driver that inserts V E R Y E A S Y Q U E S T I O N in the BST. The value of each will be its place in input * 5. (V is 5, first E is 10, R is 15, etc). Print out the tree (key and value) in inorder traversal (for key) and in level order (for key). You must comment your code such that you show that you understand everything this code is doing (even if you didn't write it). There must be a comment for each task or set of computations that shows this. Hint: the output from level order (for keys) should have the same levels/keys in each level as your answer to question 1. If not, something is wrong.

Submission:

Rubric for question 1

- 10 points - Must show each step in the insertion process in a separate tree picture
- 10 points – correct insertion for each step
- 10 points – Correct resulting tree

Rubric for question 2

- 20 points – comments explaining what every task or set of computations is doing – this is the proof that you understand the code you picked up or wrote.
- 10 points – program runs correctly
- 10 points – output shows the correct inorder traversal given the required input.
- 10 points – output shows the correct level order traversal given the required input.
- 5 points – output shows correct value for each key in the output
- 5 points – Comment that shows where the BST came from
- 10 points – driver (main) you wrote from scratch
- You will get a 0 if you:
 - a) Did not submit the source code
 - b) Did not make your binary search tree generic