Lab 3 consisted of of just five objectives first drawing a binary search tree. Second objective was doing a search through a binary search tree, only using a iterative call meaning no recursive.Third objective was to buiuld a binary search tree from a in order list, while the fourth was the oppisite of the previous objective. The last objective was to print all nodes in a certain depth.

For objective one my thoughts where to use what I made in lab 1 the draw circles and the line ones to make the binary tree and just place the item with in the circle. For objective 2 i thought of using the finding method’s logic and instead of doing it recursive call change the loop as a while loop. For objective three I thought about how the binary search tree should be build and balanced, i came up with thinking of the middle being key and just running it recursively.

Objective 4 I first needed to travers to the lowest value then add the left, root and then right into the list. For objective 5 was to just have the recursive call do most of the work along by every recursive call decreasing by 1 until reaching zero and then at that point the recursive calls should of grew as they need to be to find and print the nodes.

Starting off the biggest struggle I got struggled with is drawing the binary search tree with code. I spent a bit too much time on it. I kept on trying to combine draw circles with the other one but i just kept on messing it up and my program not running i got backed up, until a classmate told me to do at least the others, wich for me where alot more easier. Then on the second problem the same friend adviced to use similar logic to the find method previously availble on the code gifted to us. The third objective i had to figure out how to balance a tree i thought of using the mid element as the root and from there try to balance it out so i went a head and did so by having nodes created and putting them in order as they should be.On problem 4 i went ahead and traversed the binary tree to the left, append each item in the order of left root then right then headed to the right. Lastly for problem 5 it was pretty straight foward to just keep on calling the children and if d reaches 0 print those.

I struggled with the beginning of lab 3 but once past that it was alot more easier, it took too long to finished but at least i finished what i wanted