BST

In this project I have made a class of binary tree node and a class of book. Class BinaryTree have all the functions and methods. Whereas Struct Treenode have parents and children's , book have all the Book information that should be inside the Book which is basically ISBN, Title, Author's name, Quantity, on sale and price.

I'm creating a simple inventory for Sales Associate associate or a manager for retail store and using a class and a struct for this tree. Its very easy to organise the methods and declaration in class and it's good practice to have clean and organise data structure. Basically when we declare a class or a struct the main purpose is finding the connection between different kind of relation between these elements.

I'm adding a every single element in a binary search tree in order reading from a txt file in the first case. Add function is used for this method, basically it finds/searches if there is a spaces(free place) in our created binary search tree,and if there is it add the book to the possible place, and organizes it by the order.

In case two I'm adding a new book from a user that is not in the .txt file, so the new book will be added in order. Basically,adding a single data with a element in a binary tree. Deleting a selected item if it exist from the file.

In part 3 I'm,finding a data and printing out and printing everything in the order. Basically I'm searching a book with ISBN or TITLE which is integer or string, printing it and asking user if they want to delete it, if they want then delete it. I have done this part separately for string and integer, so when user have ISBN I search for ISBN and compare if it exist or not,if does then delete that book if user wants to. Same with string.There might be a better way to do it, without

creating two different function for ISBN and TITLE but I cannot think of anything then template, whereas I wasn't able to use a template class in this project since there is a header and .cpp file.

In part 4 I'm comparing the price of book and printing that book and the book that cost less price than that. Basically I'm doing the same workload as part 3 that I am asking if user have either a string or integer and search for it and if it exit print but instead of deleting it, I'm modifying the file as user wishes.

Here in part 5 I'm again doing basically the same thing I did in part 3 and 4, I'm searchin for a match called price but instead of printing out one single book I'm printing out every book of price below that price. We need to have a extra for loop for this case. I was guessing the way to do this method by using a successor function and setting my successor as a user entered price but I needed to do other extra function which I would basically won't need for this project, so i went the easy way.

In part 6 I'm Printing the compared onsale book. Here I'm printing out everything from my tree in order this line of codes basically helps to see if we actually added a book in number 2, delete a book from number 3 and modified a book from number 4.

Overall in part3,4,5 and 6, I could have just made a single search function and and print as well add delete function. But I wanted to have a different methodes and more practice doing all of this code that's why I have similar function in these methodes, which is costly in a real world but it's efficient because if something goes wrong in a single function, changing one file wouldnt be more costly than changing all files.

In part7 , I'm comparing the book,if they are in sale or not,and if they are then printing them out.

lastly, I’m deleting everything from my tree and exiting the program.