Lab BST

Data Structures CSCI 2320

Lab Objective

Learn how to implement and use a Binary Search Tree (BST) ADT in C++.

Lab Overview

You will receive a partial implementation of a BST in C++. Your mission is to implement some of the methods for the class as well as understand all the other methods.

Lab Tasks

Implement the following methods for the BST class in **`BST.h`**:

- **`void preorder(std::ostream &out);`** This method traverses the BST using preorder and prints out each data value in the tree. Make sure the data values printed are separated by whitespace. See inorder for examples of how you might implement this method. Verify that your order is correct.
- **`void postorder(std::ostream &out);`** This method traverses the BST using postorder and prints out each data value in the tree. Make sure the data values printed are separated by whitespace. See inorder for examples of how you might implement this method. Verify that your order is correct.
- **`bool search(const DataType& item) const;`** This method searches for the specified data value in the tree. The value `true` is returned if the value exists in the tree and `false` is returned if it does not exist.
- **`void BST<DataType>::search2(const DataType& item, bool& found,BST<DataType>::BinNodePointer& locptr,BST<DataType>::BinNodePointer& parent);`** This method searches for the specified data value in the tree. The `found`, `locptr`, and `parent` variables are passed by reference. See the `remove` method to understand how it is used. The `parent` variable should point to the parent node of `locptr`. See `insert` for hints on how to manage the parent pointer.

Final output

Your program must compile and produce output. The final output will not be graded, so you may experiement in your main.

Rubric

Name	Description	Points
Coding Style	Run cpplint on student code	10
BST Empty		0
BST SearchFoundAndNotFound		40
BST Insert		0
BST Delete		0
BST Inorder		10
BST Preorder		20
BST Postorder		20
BST RemoveException		0
BST InsertException		0
Total Points		100

Due Dates and Honor

The due date is specified on Blackboard.

This is an ***independent*** programming project, and it is very important that you understand and abide by the ***academic integrity policy*** concerning programming projects. Remember, your personal honor and integrity is far more important than your grade on the project.

Grading

This lab is available in GitHub Classroom. Accept the URL on Blackboard and then clone your repository to your machine for development. Your lab will be graded automatically via GitHub. Please check the grading results each time you check in your code. Your final grade will be based upon your last sync to GitHub before the deadline.

Project Artifacts

The following should be completed by the due date/time specified on Blackboard.

- Check in all source code changes to your GitHub repository. Please check your URL using a web browser to verify that your changes have been synced.
- Submit the URL for your repository to Blackboard.

© Copyright 2024 by Michelle Talley

You may not publish this document on any website or share it with anyone without explicit permission of the author.