CSUC CSCI-311 - Algorithms and Data Structures

Lab Assignment 4

Goal: 1) continue practice using BST.

In this lab we will continue the process of building our own binary search tree class by implementing the other methods in the BST_skelenton.cpp file from your lab3. **You should do this lab by using your solution "BST.cpp" file from your lab3 as the starting point.**

Submission: C++ solutions to these problems should be written in a file called BST.cpp (that builds on your lab3 solution code file) Other related files(BST.h, Node.h, Node.cpp, and BSTDriver.cpp as well as the test cases) are posted under the corresponding link on Canvas. Remember to submit your solution file "BST.cpp" to inginious. (inginious.csuchico.edu)

Coding Style: Note that your submission should use good C++ coding style and should include appropriate comments (for readability and maintainability). Specifically, your code must follow common C++ coding conventions for Naming, Indentation, and Comments. Points will be deducted if these are not present.

Collaboration: There will be time in lab to discuss these problems in small groups and I highly encourage you to collaborate with one another outside of class. However, you must write up your own solutions **independently** of one another. In addition, do not post solutions in any way. Also, please include a list of the people you work with in a comment section at the top of your submission.

Have fun!

Assignment Date:	**Oct 7, 2025**
Due Date:	**11:59pm on Oct 15, 2025**
Grace Due Date:	**11:59pm on Oct 18, 2025**

Grading Notes: 1) Assignment is due on the Due Date. 2) For each day late after the Due Date, there will be 10% penalty on the assignment's grades. 3) Submission is not accepted after the Grace Due Date. In other words, you will receive zero pts if your submission is not received by the Grace Due Date.

Special Nodes: **You are supposed to continue to work on your BST.cpp file produced from your lab3.**

Grading: Coding Style 5 pts, Test Cases 95 pts, Total 100 pts.

Assignment Questions (95 pts)

1. Implement the minimum method for binary search trees.
2. Implement the maximum method for binary search trees.
3. Implement the delete method for binary search trees. When the node to be removed has two children, search the right subtree for a replacement value. Do not forget to update the size of the tree when appropriate (including for insertion).
4. Implement the in-order traversal method for binary search trees.
5. Implement the post-order traversal method for binary search trees.