This is an individual assignment. Seeking direct help from students, tutors, and websites such as chegg or stack overflow will be construed as a violation of the honor code.

Semester Project Part 4: The AVL Tree

Data Structures and Analysis of Algorithms, akk5

Objectives

- To strengthen student's knowledge of C++ programming
- To give the student experience in writing Data Structures for data types
- To give the student experience writing a non-linear data structure
- To give the student experience implementing an AVL tree

Instructions

Attention!!! You will be using your BST tree or the BST code base I have provided for this project.

For this assignment you must implement an AVL which stores strings. Your AVL must implement all of the basic operations we have talked about in class.

You should then write a program that allows a user to interact with an instance of the AVL you have implemented. This program should implement a text-based interface that allows the user to:

1. Create an empty AVL. This should warn the user they are deleting the existing AVL and ask them if they wish to proceed. Remind the user they can save the contents of their AVL to a file.
2. Insert a string into the current AVL.
3. Search for a string in the current AVL.
4. Remove a string from the current AVL.
5. Output the in-order traversal of the current AVL.
6. Output the pre-order traversal of the current AVL.
7. Output the post-order traversal of the current AVL.
8. Save the post-order traversal of the current AVL to a user specified filename.
9. Exit.

Make certain you inform the user of the available commands and any information pertaining to the state of the system such as whether an AVL has been created, the number of nodes in the AVL, etc.

The BST and the AVL share lots of code in common. It would be wise to use your BST as a starting point for implementing the AVL. I will provide you with a working BST code base, but not a working user interface to make this easier.

Students who remember inheritance and are comfortable with inheritance might find implementing the AVL as a subclass of the BST an easy way to prevent copy/paste issues. The use of inheritance is not a requirement for this project.

This is an individual assignment. Seeking direct help from students, tutors, and websites such as chegg or stack overflow will be construed as a violation of the honor code.

Grading Breakdown

Point Breakdown	
Structure	12 pts
The program has a header comment with the required information.	3 pts
The overall readability of the program.	3 pts
Program uses separate files for main and class definitions	3 pts
Program includes meaningful comments	3 pts
Syntax	18 pts
Implements Class AVL correctly	9 pts
Implements Class Node correctly	9 pts
Behavior	70 pts
Program handles the following correctly	
• Provide the user with a menu of choices	7 pts
• Create a new AVL … must perform cleanup	7 pts
• Insert a string	7 pts
• Remove a string	7 pts
• Find a string	7 pts
• Output the pre-order traversal	7 pts
• Output the post-order traversal	7 pts
• Output the in-order traversal	7 pts
• Save the post-order traversal to file	7 pts
• Exits the program … must perform cleanup	7 pts
Total Possible Points	**100pts**
Penalties	
Program does NOT compile	-100
Late up to 24 hrs	-30
Late more than 24hrs	-100

This is an individual assignment. Seeking direct help from students, tutors, and websites such as chegg or stack overflow will be construed as a violation of the honor code.

Header Comment

At the top of each program, type in the following comment:

/*

Student Name: <student name>

Student NetID: <student NetID>

Compiler Used: <Visual Studio, GCC, etc.>

Program Description:

<Write a short description of the program.>

*/

Example:

/*

Student Name: John Smith

Student NetID: jjjs123

Compiler Used: Eclipse using MinGW

Program Description:

This program prints lots and lots of strings!!

*/

Assignment Information

Due Date: 11/4/2019 (Section 1), 11/3/2019 (Section 3)

Files Expected:

1. Main.cpp – File containing function main
2. AVL.h - File containing class Node and class AVL.
3. AVL.cpp - File containing the implementation for the AVL