Assignment 5: Treebook 🌲

*Due **May 17th at 11:59pm***

Table of Contents

Overview

For this assignment, you will be implementing parts of a class to allow for operator overloads, as well as to remove some aspects of the special member functions.

Download the starter code here.

Part 1: Make a profile

The newest Stanford social media startup is Treebook, and you're a founding member of the team! To get the product off the ground and compete with an unnamed, completely legally unaffiliated app from Harvard, you've been assigned the task of implementing user profiles.

You will be writing the User class into the **user.cpp** file. The header **user.h** has been provided and started for you, but you will need to add special member functions to it.

- **Special member functions**: You will need to declare each special member function in the header file, including the default constructor. The default constructor should create a user profile with no name (an empty string).
    - Please also create an additional constructor that takes in a string name and uses this to change the name private variable.
    - The class should be move assignable and move constructible and copy constructible, but not copy assignable.

Part 2: ABF: Always Be Friending

Now that the User class has been created, you've moved on to creating friends for them! You'll be implementing two operators: the + and the < to be able to build up a user's friend list.

- The + operator will representing adding a user to another user's friend list. This should be symmetric, meaning that adding, for example, Charlie to Alice's friend list should cause Alice to also be in Charlie's list. This should be implemented as a non-member function operator overload in main.cpp. **Tip:** Look at the operator's usage in main.cpp to see what we expect the return to be.
- The < operator is necessary to store Users in a set, as they require comparison operators. For this, we will base comparison on alphabetical sorting of the User's name. This should be implemented as a member function.

That should be all! You can run the code in main.cpp to make sure you don't have compiler errors. Then, add your files to Paperless.

Feedback Survey

Please fill out this anonymous feedback form to help us improve these short assignments in the future!

Submitting Instructions

When you have completed this assignment, upload all of the files to Paperless under the correct assignment heading.
Your deliverables should be:

- **main.cpp**
- **user.cpp**
- **user.h**

You may resubmit as many times as you'd like before the deadline. Please let us know if you have any questions as you work on the assignment!