

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 1: Implementing a Doubly Linked List

Data Structures and Analysis of Algorithms, akk5

Objectives

- To strengthen student's knowledge of C++ programming
- To give student experience reading and parsing strings of commands
- To give student experience in writing Data Structures for data types

Instructions

For this assignment you must write a program that implements and manages a list of information. The list must store entries comprised of a string representing a user id and an integer representing the user's security level; for simplicity, this program doesn't concern itself with duplicate uid's. The linked list should be implemented as a doubly linked list.

Your program should implement a command line (text-based interface) capable of handling the following commands:

exit – exits the program

load <file> - parses the contents of the file as if they were entered from the command line

display – displays the contents of the double linked list

find <uid> - finds the specified uid and displays its security level. Should inform the user on a failure.

remove <uid> - Removes the specified uid from the list. Should inform the user on a failure.

insert <uid> <security level> - Inserts the specified uid, security level pair at the tail of the doubly linked list.

insert <uid> <security level> before <uid2> - Inserts the specified uid, security level pair immediately before uid2. Should inform the user if the insert fails because uid2 isn't in the list.

insert <uid> <security level> after <uid2> - Inserts the specified uid, security level pair immediately after uid2. Should inform the user if the insert fails because uid2 isn't in the list.

Guidance

Parsing text can be a frustrating part of any programming assignment and is probably more challenging than implementing the actual doubly linked list class and its methods. Although C++ supports multiple approaches to handling this challenge, I suggest the following approach.

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.

First, forego the use of the >> operator in conjunction with any istream you might consider using (this would be cin and an input file stream for this program); instead, read the entirety of each line using a call to getline. We will break the data into its different parts afterwards using a process known as tokenization.

Once you have successfully read a command in a string, convert the string in a stringstream for further processing. The stringstream is probably new to most of you, but if you are comfortable working with streams it is easy enough to understand.

Once you have successfully read a command in a string, convert the string in a stringstream for further processing. The stringstream is probably new to most of you, but if you are comfortable working with streams it is easy enough to understand.

Stringstream is accessed by including sstream (#include <sstream>); since stringstream is in the std namespace, make certain you place a using clause in your code as well (using std::stringstream;).

You can convert a string to a stringstream as part of declaring the variable; as an example, the line of code below creates a stringstream labeled ss containing the contents of the string variable str:

```
stringstream ss(str);
```

At this point, any function, method, or operator that works with an istream will work with the stringstream ss. Of interest is a variation of the getline function. When tokenizing strings, it is necessary to break them apart based on a given delimiter character. The getline function supports a third argument that is often used to specify such a delimiter. As an example, the line of code below reads the first word of the stringstream labeled ss into the string variable cmd:

```
getline(ss,cmd, ' ');
```

For this assignment, it will also be necessary to convert a string into an integer. Since C++ 2011, this has been accomplished using the function stoi. The stoi function attempts to convert a string into an integer, returning the integer or throwing an exception if the conversion fails. The following block of code is an example of using stoi to convert the contents of the string str into an integer name sec:

```
try {
    sec = stoi(str);
} catch (...)
{
    // failed to convert the string to an integer, handle that failure here
}
```

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.

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	24 pts
Implements Class Node correctly	9 pts
Implements Class List correctly	15 pts
Behavior	64 pts
Program handles all command inputs properly	
<ul style="list-style-type: none"> Exit the program 	8 pts
<ul style="list-style-type: none"> Display the list correctly 	8 pts
<ul style="list-style-type: none"> Load a valid file 	8 pts
<ul style="list-style-type: none"> Find a value in the list 	8 pts
<ul style="list-style-type: none"> Remove a value from the list 	8 pts
<ul style="list-style-type: none"> Insert a uid, security pair at the tail of the list 	8 pts
<ul style="list-style-type: none"> Insert a uid, security pair before another uid 	8 pts
<ul style="list-style-type: none"> Insert a uid, security pair after another uid 	8 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: 1/26/2020

Files Expected:

1. Main.cpp – File containing function main
2. List.h - File containing the List and Node class definitions.
3. List.cpp - File containing the code for the List and Node methods.

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.