Lab LinkedList

Data Structures CSCI 2320

Lab Objective

Learn how to implement and use a LinkedList ADT in C++.

Lab Overview

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

Lab Tasks

Task 1

Implement the following methods for the LinkedList class in `LinkedList.h`:

- `void insert(int index, const T& value)` This method inserts an item in the list at the specified index, which starts at zero. This method should throw an `out_of_range` exception.
- `void push_back(const T& value)` This method places an item at the back of the list.
- `void push_front(const T& value)` This method places an item at the front of the list.
- `void pop_back()` This method removes the last item in the list. This method should throw an `out_of_range` exception.
- `void pop_front()` This method removes the first item in the list. This method should throw an `out_of_range` exception.

Task 2

Implement the following in your main driver `main.cpp`.

- Print this exact message at the beginning of your main before any output.
  `Welcome to the LinkedList test program.`
- Print this exact message at the end of your main after all other output.
  `Goodbye!`

Final output

Your output must be ***exactly*** the following to receive the points allocated for output.

```
Welcome to the LinkedList test program.

Test some integer lists...
Integer list: 5 10 20 30
Front element: 5
Back element: 30
Pop 5
Integer list: 10 20 30
Pop 10
Integer list 1: 20 30
Length of list 1 is 2
List 1 is not empty
Pop 20
Pop 30
Integer list::
List 1 is empty
Integer list 1: 10 20 30
Copy integer list 1 to integer list 2.
Integer list 2: 10 20 30
Integer list 1:30:
Integer list 2:10 20 30:
Integer list 2 via std cout: 10 20 30:

Test some exceptions...
List 1 is empty
Front element: Exception: LinkedList is empty
Back element: Exception: LinkedList is empty

Test some string lists...
String list 1: Please, may I have some spam?
String list 1: have some
Goodbye!
```

Rubric

Name	Description	Points
Main Output	Run the student main and compare output required output	5
Coding Style	Run cpplint on student code to check coding style	5
Empty	Test Method	1
Size	Test Method	1
PushBackAndBack	Test Method	15
PopBack	Test Method	10
PopFront	Test Method	10
PushFrontAndFront	Test Method	15
Insert	Test Method	20
Erase	Test Method	1
FrontException	Test Method	1
BackException	Test Method	1
PopBackException	Test Method	5
PopFrontException	Test Method	5
InsertException	Test Method	5
Print	Test Method	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.