COMP 3522 Lab 8

Object Oriented Programming with C++

Due Nov 22 11:59PM

1 Instructions

Today you will design and implement a new template class **MyLinkedList** which simulates some of the functionality of a singly linked list. You will then use a template function, **print** to print out the contents of your linked list

2 Set up your lab

Start by creating a new project:

1. Create a new project in CLion. Let's call it lab8. Choose C++ Executable for the project type and ensure you select C++14 as the Language Standard.
2. Add this project to version control. From the VCS menu in CLion select Import into Version Control | Create Git Repository... to convert the project folder to a git repository.
3. Add the project to GitHub by returning to the VCS menu and selecting Import into Version Control | Share Project on GitHub. Call it lab8, make sure it's private, and add a first commit comment. It's fine to add all the files for your first commit.
4. Visit GitHub and ensure your repository appears. Add me as a collaborator. In GitHub, I am known as jeffbcit. You'll recognize my avatar.

3 Requirements & Grading

A singly linked list is a linked data structure that consists of sequentially linked records called nodes. Each node contains data and a field that references the next node. You will write code to implement the node, and singly linked list of nodes.

A linked list is visually represented as follows:

1. (3) Implement a node using a template class named "Node." This template is parameterized over typename T. A node contains member variables for:
   - Data of type T
   - A node pointer to the next node

Implement a template class named “MyLinkedList.” This template is parameterized over typename T. This class contains the following:

- A pointer to the head node named “head”
- A function that pushes a new node with its data to the front of the linked list. Name the function “pushFront” it takes a parameter “data” of type T.
- A function that creates a new node with its data to the back of the linked list. Name the function “pushBack”, it takes a parameter “data” of type T.
- A function that creates a new node with its data and inserts it after a given node. Name the function “insert” it takes two parameters: a pointer to the prevNode, and “data” of type T

You may NOT use STL containers to assist in creating the linked list. The MyLinkedList class will be written over the next several sections.

2. (2) Declare the template class MyLinkedList and the instance variable “head” which is a pointer to the first node in the list.

3. (3) Write a function that creates a new node with its data, and adds it to the front of the linked list. Name the function “pushFront.” It takes a parameter “data” of type T.

4. (5) Write a function that creates a new node with its data, and adds it to the back of the linked list. Name the function “pushBack.” It takes a parameter “data” of type T.

5. (5) Write a function that creates a new node with its data, and inserts it after a given node. Name the function “insert.” It takes two parameters: a pointer to the Node “prevNode”, and “data” of type T

The MyLinkedList class is now complete.

6. (3) Outside of the LinkedListClass, implement a template print function that will iterate over any type of linked list and data, and print out the contents of the linked list. This template is parameterized over two typenames, DLL and T
    - The print function itself accepts one parameter of type DLL

7. (4) We will test your linkedlist class and print function in the main function
    - Create a linked list parameterized with int named intList
    - Push 3 to the back of the list
    - Push 1 to the front of the list
    - Insert 2 into the middle of the list
    - Print out your intList using your template print function
    - Create a linked list parameterized with string named stringList
    - Push “c” to the back of the list
    - Push “a” to the front of the list
    - Insert “b” into the middle of the list
    - Print out your stringList using your template print function
    - Ensure your output is similar to: 1 2 3 a b c

That’s it. Check your work. After you submit your Lab to GitHub, send me a private message on Slack telling me you're finished, along with your student number, gitName, and collaboration url. ie: "Jeff I uploaded my work to gitHub. My student number is A00XXXXXX and my gitName is YYYYY, my git collaboration url is: ZZZZZZZZZZZZZZZ