COP 4530
Project 1
Spring 2024

Instructions

For this programming project, you will implement a d*oubly linked list* to insert the integer data in increasing order and convert a list to a string.

You are not allowed to use the C++ Standard Library Lists or other sources to implement the linked list

Code testing- Detailed instructions will be made available in a separate module in Canvas.

Submission

Submit a .zip file containing all your files with the format "`your login id_your group member login id_ assignment 1.zip`". Your submission should contain the following files-

1. `IntDList.hpp`
2. A .cpp file that defines all the declared functions in the `IntDList.hpp` file.
3. Any additional files necessary to successfully compile your code.

Abstract Class and Files

`void addToHead(int);(helper function)`
Add a node with the input value as the head node

`void insertInOrder (int);(required function)`
Add nodes in an increasing order. So if we are adding nodes in the following manner-

```
myList.insertInOrder(9);
myList.insertInOrder(8);
myList.insertInOrder(0);
myList.insertInOrder(3);
```

And print the link list, it should return `0389`

`void addToTail(int);(helper function)`
Add a node with the input value as the tail(last) node

`int deleteFromHead();(required function)`
delete the head and return its value

`int deleteFromTail();(required function)`
Delete the tail and return its value

`void deleteNode(int);(required function)`
Delete node which contains the input integer value

`string addToString() const;(required function)`
This method returns the string of the ordered integers. Use `sstream` and `iomanip` (with argument to `setw` as `0`). See `example.cpp` for details

Example interaction

Below are some examples of how your code should run(as shown in the test files)

```
IntDLList myList; // is empty
myList.addToHead(8); // list: 8
myList.insertInOrder(3); // List: 3 8
myList.addToTail(9); // List: 3 8 9
myList.deleteNode(8); // List: 3 9
myList.addToString(); //returns a string “3 9”
myList.deleteFromTail();//List: 3 (returns 9)
myList.insertInOrder(5); //List: 3 5
myList.deleteFromHead();//List 5 (returns 3)
```

Grading-

- Correct implementation of the required functions- **(4*20)=80**
- Correct Destructor implementation- **10**
- Documentation - **10** (1. Include comments detailing the logic behind the function implementations

2. Add your name and your other group member's name at the top of your file as a comment)

Assumptions-

1. `addtoHead()` is always called before `insertInOrder()`
2. A value to be deleted is always present in the linked list.