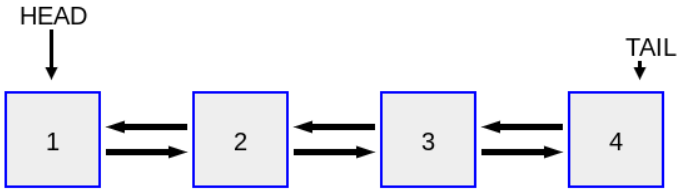
CS136L: Computer Science II Lab – Fall 2018

Assignment	Points	Announced	Due
#8	40	Nov-21	Dec-07

Doubly Linked List

1 Overview

In this lab we will implement a linked list capable of traversing both forward and backwards between nodes.



2 Learning Outcomes

By the end of this project students should be able to:

- write custom collections;
- use generic classes and interfaces;
- work effectively with a partner using pair-programming;
- write an effective report that describes the students' problem-solving process.

3 Pre-Lab Instructions

Do this part before you come to lab:

- Review chapters 15, 16 and 18 on the Java Collections Framework, basic data structures and generic classes.

4 Submission Instructions

1. You should use NetBeans for this lab assignment.
2. In addition to the lab report, submit a ZIP archive that is exported from NetBeans by using the NetBeans menu (File → Export Project → To ZIP).
3. Add Javadoc comments for all classes, fields and methods that you implement

5 Lab Instructions

Do this part in lab:

1. The first step is to create the `Node<T>` class that represents a single node in a linked list. The `Node<T>` class from the singly linked list implementation has two fields: data and next. The data field stores the node payload and the next field points to the following node. A doubly linked list node needs to point to the preceding node as well. Note that in this lab, the `Node<T>` class is generic and an inner class inside the `DoublyLinkedList<T>` class.

2. Now, we will implement the `DoublyLinkedList<T>` class. Study the below required methods and then discuss with your partner and the lab TA what fields you need to add to the class. This class should provide these methods:

   a. `String toString()` ➔ Returns the data values in the linked list separated with commas and sounded with two square brackets. For example [3,1,20].
   b. `void addFirst(T value)` ➔ Adds a value to the linked list so that is stored in the first node.
   c. `void addLast(T value)` ➔ Adds a value to the linked list so that is stored in the last node.
   d. `T getFirst()` ➔ Retrieves the data of the first node. If there is no first node (i.e. the linked list is empty), it should throw an `ElementNotFoundException` exception.
   e. `T getLast()` ➔ Retrieves the data of the last node. If there is no last node (i.e. the linked list is empty), it should throw an `ElementNotFoundException` exception.
   f. `void removeFirst()` ➔ Removes the first node. If there is no first node (i.e. the linked list is empty), it should throw an `ElementNotFoundException` exception.
   g. `void removeLast()` ➔ Removes the last node. If there is no last node (i.e. the linked list is empty), it should throw an `ElementNotFoundException` exception.
   h. `int size()` ➔ Returns the number of values in the linked list.
   i. `T get(int index)` ➔ Returns the value at `index`. If that index doesn't exist, it should throw an `IndexOutOfBoundsException` exception.
   j. `int indexOf (T value)` ➔ Returns the index of the node storing `value`. If value is not in the linked list, it should return -1.
   k. `ListIterator<T> listIterator()` ➔ Returns an instance of a `DoublyListIterator` class.

3. The third step is implementing the `DoublyListIterator` class which implements the provided `ListIterator<T>` interface.

4. Finally, test your code using the provided `DoublyLinkedListApp` program.

6 Lab Report

Each pair of students will write a single lab report together and each student will turn in that same lab report on BBLearn. Submissions from each student on a pair should be identical.

Your lab report should begin with a preamble that contains:

- The lab assignment number and name
- Your name(s)
- The date
- The lab section number

It should then be followed by four numbered sections:

1. Problem Statement

In this section you should describe the problem in *your* own words. The problem statement should answer questions like:

- What are the important features of the problem?
- What are the problem requirements?

This section should also include a reasonably complete list of requirements in the assignment. Following your description of the problem, include a bulleted list of specific features to implement. If there are any specific functions, classes or numeric requirements given to you, they should be represented in this bulleted list.

2. Planning

In the second section you should describe what planning you did in order to solve the problem. You should include planning artifacts like sketches, diagrams, or pseudocode you may have used. You should also describe your planning process. List the specific data structures or techniques you plan on using, and why.

3. Implementation and Testing

In the third section you should describe how you implemented your plan. As directed by the lab instructor you should (as appropriate) include:

- a copy of your source code
- a screen shot of your running application / solution
- results from testing

4. Reflection

In the last section you should reflect on the project. Consider different things you could have done to make your solution better. This might include code organization improvements, design improvements, etc.

You should also ask yourself what were the key insights or features of your solution? Were there alternative approaches or techniques you could have employed? How would these alternatives have impacted a different solution?

5. Partner Rating

Every assignment you are required to rate your partner with a score -1, 0 or +1. This should be submitted in the comment section of the BBLearn submission, and not in the report document. If you don't want to give your partner a negative rating making sure not to use a dash before listing the number! You do not have to tell your partner the rating you assign them. A rating of 1 indicates that your partner was particularly helpful or contributed exceptional effort. A rating of 0 indicates that your partner met the class expectations of them. Rating your partner at -1 means that they refused to contribute to the project, failed to put in a reasonable effort or actively blocked you from participating. If a student receives three ratings of -1 they must attend a mandatory meeting with the instructor to discuss the situation, and receiving additional -1 ratings beyond that, the student risks losing a letter grade, or even failing the course.

6. Contribution

Every assignment you are required to describe your contribution to coding of the solution and writing of the report. You must include the percentage of your contribution. This should be submitted in the comment section of the BBLearn submission, and not in the report document.

7 Grading Rubric

This lab assignment will be graded according to this rubric:

Criteria / Component	Points
Report	10 pts
Correct implementation of `String toString()`	1

Correct implementation of `void addFirst(T value)`	1
Correct implementation of `void addLast(T value)`	1
Correct implementation of `T getFirst()`	1
Correct implementation of `T getLast()`	1
Correct implementation of `void removeFirst()`	2
Correct implementation of `void removeLast()`	2
Correct implementation of `int size()`	1
Correct implementation of `T get(int index)`	2
Correct implementation of `int indexOf (T value)`	2
Correct implementation of `ListIterator<T> listIterator()`	2
Correct implementation of `ListIterator void add(E e)`	2
Correct implementation of `ListIterator boolean hasNext()`	2
Correct implementation of `ListIterator boolean hasPrevious()`	2
Correct implementation of `ListIterator T next()`	2
Correct implementation of `ListIterator T previous()`	2
Correct implementation of `ListIterator void remove()`	2
Correct implementation of `ListIterator void set(E e)`	2

Lab assignment penalties:

Item	Points
Compilation errors	-40
Missing program	-40
Missing report	-10
Missing names	-4
Missing partner rating	-2
Missing screenshots	-2
Missing contribution description and/or percentage	-5
Too late to pair (if attended)	-4
Absent	-10
Insufficient / No commenting (if required)	-5
Improper format (e.g. NetBeans ZIP archive instead of a Java file)	-5

Note:

If your partner is not responding to your emails and/or not collaborating, don't hesitate to reach out to the lab TA aide and/or the primary instructor.