

King Fahd University of Petroleum & Minerals
College of Computer Science and Engineering
Information and Computer Science Department
ICS 202 – Data Structures

[Linked Lists-Part I](#)

Objectives

The objective of this lab is to design, implement and use Linked Lists

Outcomes

After completing this Lab, students are expected to:

- Understand the structure of a singly linked list class
- Add methods and enhance the functionality of a linked list class
- Use linked lists class in an application

Notes

For the purpose of this lab, you may download the attached programs.

Lab Exercises

1. Download the files related to Linked Lists associated with this lab. Compile and run the programs.
2. Warm-up Exercise: Write a method **public int length()** in the class **SLL<T>.java** that returns the length of the linked list. (Additional Exercise: Can you make a recursive length method?)

3. Add the following methods to the class **SLL<T>.java**:

(a) **public void insertAfter(T newElem, T existingElem)** that inserts an element **newElem** after the element **existingElem**. If no **existingElem** exists, then the method **throws an exception**. If more than one instance of **existingElem** exists, then the method inserts after the first instance of **existingElem**.

For example, suppose your linked list (of integers) is: [3 5 4 2 9],

Then a call to **insertAfter(Integer.valueOf(5), Integer.valueOf(9))** would result in the following linked list: [3 5 4 2 9 5]

A call to **insertAfter(Integer.valueOf(7), Integer.valueOf(5))** would result in [3 5 7 4 2 9 5]

A call to **insertAfter(Integer.valueOf(8), Integer.valueOf(10))** would throw an exception since Element 10 does not exist in the linked list.

- (b) Add the following methods with the same methodology as **insertAfter**:
- **public void deleteAfter(T elem)**

Make sure you test for cases where the list has only one element.

- (c) Write a **test class** to test these methods.