SENG2200/6220 –Programming Languages & Paradigms
Computer Lab for Week 7, Semester 1, 2020

Objectives

This lab aims to understand the iterators in Java and do exercises to implement iterator for a generic container class.

Questions

Modify the following code to support an iterator:

```
public class SimpleLinkedList<E> {
  private Node<E> sentinel;
  private int size;

  public SimpleLinkedList<E>() {
    sentinel = new Node<E>();
    size = 0;
  }

  public int getSize() {
    return size;
  }

  public void append(E o) {
    Node n = sentinel;
    while (n.getNext()!=null) {
      n = n.getNext();
    }
    n.setNext(new Node<E>(o));
    size++;
  }

  public void prepend(E o) {
    Node n = new Node<E>(o);
    n.setNext(sentinel.getNext());
    sentinel.setNext(n);
    size++;
  }

  public E removeHead()
         throws IndexOutOfBoundsException {
    if (size<=0){
      throw new IndexOutOfBoundsException(
        "Cannot remove from empty list");
    }
    Node<E> n = sentinel.getNext();
    sentinel.setNext(n.getNext());
    size--;
    return n.getData();
  }

  public E removeTail()
         throws IndexOutOfBoundsException {
```

```
    if (size<=1) {
      return removeHead();
    }
    Node n = sentinel;
    while (n.getNext().getNext()!=null) {
      n = n.getNext();
    }
    Node<E> o = n.getNext();
    n.setNext(null);
    size--;
    return o.getData();
  }

  private class Node<E> {
    private E data;
    private Node next;
    private Node() {
      this(null);
    }
    private Node(E data) {
      this(data,null);
    }
    private Node(E data, Node next) {
      setData(data);
      setNext(next);
    }
    public Node getNext() {
      return next;
    }
    public void setNext(Node nextNode) {
      next = nextNode;
    }

    public E getData() {
      return data;
    }
    public void setData(E dataVal) {
      data = dataVal;
    }
  }
}
```