

COMP20010



# Data Structures and Algorithms I

## 03 - Tutorial: Linked Lists

*Dr. Aonghus Lawlor*  
*[aonghus.lawlor@ucd.ie](mailto:aonghus.lawlor@ucd.ie)*



# Linked Lists

- In this tutorial we will see how to implement a linked list in Java.
- We will apply our linked list to the Scoreboard problem (we previously used an ArrayList)
- Make the implementation generic
- Add iteration to the implementation

# Linked List

- Get some starter code at the Github link



# SinglyLinkedList

```
public class SinglyLinkedListString {  
  
    private static class Node {  
        private String data;  
        private Node next;  
  
        public Node(String e, Node n) {  
            this.data = e;  
            this.next = n;  
        }  
        public String getData() {  
            // TODO  
        }  
        public Node getNode() {  
            // TODO  
        }  
    }  
  
    private Node head = null;  
    private int size = 0;  
    public SinglyLinkedListString {  
  
    }  
  
    public int size() {  
        // TODO  
    }  
    public boolean isEmpty() {  
        // TODO  
    }  
  
    public Node first() {  
        // TODO  
    }  
}
```

```
    public Node last() {  
        // TODO  
    }  
  
    public void addFirst(String data) {  
        // TODO  
    }  
  
    public void addLast(String data) {  
        // TODO  
    }  
  
    public String removeFirst() {  
        // TODO  
    }  
  
    public String removeLast() {  
        // TODO  
    }  
  
    public String toString() {  
  
    }  
  
    public static void main(String [] args) {  
        SinglyLinkedListString ll = new SinglyLinkedListString();  
        ll.addLast("Java");  
        ll.addLast("C++");  
        ll.addLast("Python");  
        ll.addLast("Scala");  
        System.out.println(ll);  
    }  
}
```

# Generic Version

```
/*  
This is the generic version of the SinglyLinkedList
```

You need to change the signature from:

```
public class SinglyLinkedList {
```

to:

```
public class SinglyLinkedList<E> {
```

and update all the other cases where 'String' should be replaced with 'E'.

```
*/  
  
public class SinglyLinkedList<E> {
```



# Iterators

```
public Iterator<E> iterator() {  
    return new ListIterator<E>();    // create a new instance of the inner class  
}  
  
private class ListIterator<E> implements Iterator<E> {  
  
    private Node iterator;  
  
    ListIterator() {  
        iterator = head;  
    }  
  
    @Override  
    public boolean hasNext() {  
        return (iterator.getNext() != null);  
    }  
  
    @Override  
    public E next() {  
        E data = iterator.getData();  
        iterator = iterator.getNext();  
        return data;  
    }  
  
    @Override  
    public void remove() {  
        // NOT IMPLEMENTED  
    }  
}
```

```
Iterator it = ll.iterator();  
for (String data : it) {  
    System.out.println(data);  
}
```


# Scoreboard

replace your ArrayList implementation of Scoreboard.java with a SinglyLinkedList and check they give the same score board results...

```
public static void mainScores() {
    File file = new File("scores.txt");
    try {
        Scanner scanner = new Scanner(file);
        ScoreBoard scoreboard = new ScoreBoard(10);
        while (scanner.hasNext()) {
            String line = scanner.nextLine();
            Scanner lineReader = new Scanner(line).useDelimiter(",\\s?+"); // comma followed
            by any number of spaces
            String name = lineReader.next();
            int score = lineReader.nextInt();
            //System.out.println("read entry: " + name + " " + score);

            GameEntry entry = new GameEntry(name, score);
            scoreboard.add(entry);
        }
        scanner.close();

        System.out.println(scoreboard.toString());
    } catch (FileNotFoundException e) {
    }
}
```

HIGH SCORES		
RANK	SCORE	NAME
1ST	10000	BOB
2ND	10000	JWC
3RD	10000	SKT
4TH	10000	TBS
5TH	10000	MNM
6TH	10000	WKJ
7TH	10000	SVO
8TH	10000	WHO
9TH	10000	TRN
10TH	10000	JWC
		
CREDIT		0