

**LANGARA COLLEGE**  
*DEPARTMENT OF COMPUTING SCIENCE AND INFORMATION SYSTEMS*  
**CPSC 1160 - ALGORITHMS AND DATA STRUCTURES I**  
**Assignment 10 – Lab 10**  
**November 30, 2017**

**Instructions**

- This assignment is worth 10 points, and is due on November 30 at 06:30 PM.
- This is an in-class activity; you need to show your code to the TA or the instructor to get your mark.

**Part 1: Singly Linked List [7 points]**

Implement the Singly Linked List ADT according to the following class diagram:

LinkedList<T>
-head: Node<T> * -tail: Node<T> *
+LinkedList() +insertAt(index: int, newElement: T): void +removeFrom(index: int) : T +removeAll(value: T) : void +isEmpty() : bool const +print() : void const +~LinkedList()

**Notes:**

- Design and implement the Node<T> class appropriately.
- Your code is required to throw exceptions when applicable.
- The removeAll() and the print() functions are required to be implemented **recursively**.
- Use the following program to test your class:

```
#include <iostream>
#include "LinkedList.h"
using namespace std;
int main(){
    LinkedList<int>  myList;
    for(int i = 0; i < 10; i++){
        myList.insertAt(i, i);
        myList.insertAt(i, 10 - i);
    }
    cout << "The elements of the list are: \n";
    myList.print();
    myList.removeAll(5);
    myList.removeAll(9);
    cout << "After removing all the occurrences of 5 and 9, the
        elements of the list are: \n";
```

```
myList.print();
for(int i = 0; !myList.isEmpty(); i++)
    cout << "Element at " << i << " is removed: " <<
        myList.removeFrom(i);
myList.print();
system("PAUSE");
return 0;
}
```

## **Part 2: Complexity Analysis [3 points]**

Analyze the time complexity of the following functions: `removeFrom()`, `removeAll()`, and `isEmpty()`. Express the result of your analysis by a complexity function. If applicable, provide the worst-case and best-case analysis.