

**LANGARA COLLEGE**  
*DEPARTMENT OF COMPUTING SCIENCE AND INFORMATION SYSTEMS*  
**CPSC 1160 - ALGORITHMS AND DATA STRUCTURES I**  
**Assignment 09 – Lab 09**  
**November 23, 2017**

### Instructions

- This assignment is worth 10 points, and is due on November 30 at 04:00 PM.
- All the program files (.cpp and .h files) and the document file containing the complexity analysis are required to be put in a folder named **Lab09**; the whole folder then should be submitted as a single zipped file on D2L.

### Part1: Linked Queue ADT [3 points]

Implement the Queue ADT as a linked structure according to the following class diagram:

LinkedQueue<T>
-front: Node<T> *
-rear: Node<T> *
+LinkedQueue()
+insert(newElement: T): void
+remove() : T
+isEmpty() : bool const

### Notes:

- Design and implement the Node<T> class appropriately.
- Your code is required to throw exceptions when applicable.

### Part2: Linked Priority Queue ADT [3 points]

Comment your implementation of the part 1 with changes required to convert this ADT into a Priority Queue ADT. Assume the priority of an element is identified by an integer ranged from 1 to 5 where smaller integers show higher priorities.

### Part3: Complexity Analysis [2 points]

Analyze the time complexity of the insert and remove functions in both Linked Queue and Linked Priority Queue ADTs.

### Part4: Radix Sort [2 points]

Utilize the Linked Queue ADT implemented in Part 1 to write a function that uses the Radix Sort algorithm to sort a list of integers in the range [0, n) in the ascending (non-descending) order. Here is the prototype of this function:

```
void radixSort(int inputList[], int size, int n = 1000);
```

Write a program that reads a list of 3-digit non-negative integers from keyboard, calls the `radixSort` function to sort the list, and prints the sorted list on the screen.