Data Structures and Algorithms — Lab 4

# Objective

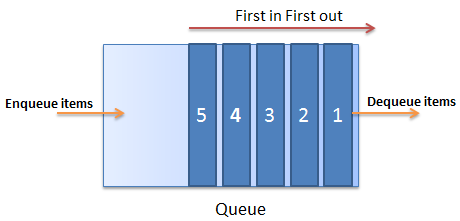
* Implementation of Queue

## Introduction

Queue is a data structure that works on First in First out (FIFO) approach. In general, items are added to the end of Array (In the way that we join at the end of a queue for a bus) and items are removed from the front of an Array. (The people at the front of the queue for the bus can get on the bus first.)

Queue class must have 5 operations. These operations are:

1. **Queue(int)** // create & then initialize the size of array
2. **isEmpty()** // no element in queue
3. **isFull()** // size of queue = number of elements in queue
4. **Enqueue(Gen)** //add an element in queue (always at end of queue)
5. **Dequeue()** //remove an element from start of queue
6. **~Queue() //** Deletes dynamically allocated memory



# Lab Task 1

The operations that we need to implement a queue using templates are:

1. **Queue(int size)**
2. **void enqueue(Gen value)**
3. **Gen dequeue()** // returns elements from the first of array
4. **void displayFirst()** // displays the value placed at the start (front) of queue
5. **bool isFull()**
6. **bool isEmpty()**
7. **~Queue()**

You should implement both:

1. **LINEAR** QUEUE – having dequeue O(n)
2. **CIRCULAR** QUEUE – having dequeue O(1)

# Lab Task 2

Write a function (that is not a part of class Queue) that reverses the contents of a queue. Contents of the original queue must be reversed. You are not allowed to use any data structure **other than stack and queue** (arrays in main should NOT be used).

# Lab Task 3

You have to implement a First in First out - FIFO (Queue Functionality) using **Stacks** only.