# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BEEE 13 (Grp1+2)**

# 

# Lab 7: Implmentation of Queues in different problems

**Date: 18th March, 2024**

**Time: 10 am - 1 pm**

# Lab Instructor: Anum Asif

# 

# Lab 7: Implementation of Queues in different problems

**Introduction**

This lab is based on queues and its implementation statically and dynamically.

**Objectives**

Objective of this lab is to get familiar with the queues and implement it in a programming language

**Tools/Software Requirement**

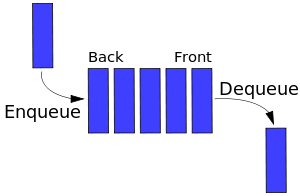
Visual Studio 2012 or gcc or g++

**Helping Material**

Lecture slides, text book

**Description**

In [computer science](http://en.wikipedia.org/wiki/Computer_science), a queue is a particular kind of [abstract data type](http://en.wikipedia.org/wiki/Abstract_data_type) or [collection](http://en.wikipedia.org/wiki/Collection_(computing)) in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position and removal of entities from the front terminal position. This makes the queue a [First-In-First-Out (FIFO) data structure](http://en.wikipedia.org/wiki/FIFO_(computing)).



The following sets of operation are generally supported by queue:

1. void Enqueue(element) – add an element at the rear end of the queue

2. element Dequeue() – removes and display the element from the front end of the queue

3. bool isEmpty() – checks if the queue is empty or not

4. bool isFull() – checks if the queue is full or not

5. void Clear() – release the memory allocated by queue

6. void FirstElement() – display the contents of first element of queue at front location

**Lab Tasks**

You have to implement a waiting room management system in an emergency ward of a hospital. Your program will assign an Id number to a patient in a first come first serve basis. The lower the id, the sooner the service will be provided to the patient.

Your program will contain the following methods:

**RegisterPatient():** This method assigns an Id (which is auto-generated) to a patient and register him/her to the system.

**ServePatient():** This method calls a patient to provide hospital service to him/her.

**CancelAll():** This method cancels all appointments of the patients so that the doctor can go to lunch.

**CanDoctorGoHome():** This method returns true if no one is waiting, otherwise, returns false.

**ShowAllPatient():** This method shows all ids of the waiting patients in SORTED order. (Hint: use the sorting methods learnt in class using the appropriate data-structure for each task) [Sorted according to their names]

Solve the above problem using an *array based queue*.

**Important Note:** Practice your knowledge of OOP with C++ when creating a solution.

**Solution:**

|  |
| --- |
| Solution |
| Task 1 Code:  Task 1 Output Screenshot: |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS.