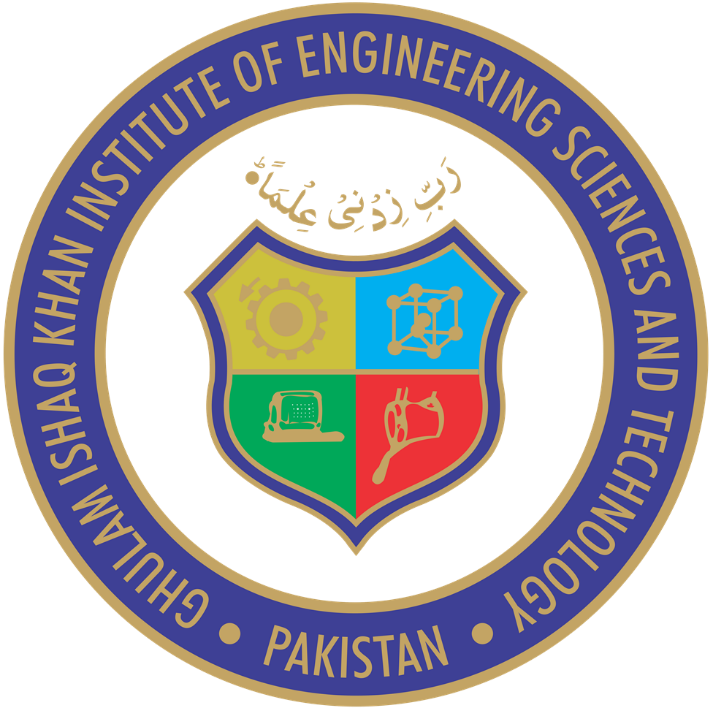
****

**CS-221 Data Structures & Algorithms Lab**

**Semester Project Report**

**MEMBERS:**

* **2021323** (M. Abdullah Rizwan)
* **2021191** (Hammad Qaiser)
* **2021016** (Abdul Moiz Javaid)

**Table of Contents:**

**1. Project Description----------------------------------3**

**2. Data Structures Used------------------------------- 3**

**3. Screenshots--------------------------------------------4**

**Project Description:**

The code provided is a C++ program that implements a hospital management system using a priority queue to manage the patient queue while storing the patient’s data in a structure.

The program begins by defining a Patient struct to represent a patient and their medical information. The Patient struct includes fields for the patient's name, age, gender, and condition (severity of their medical condition). The struct overloads the < operator to allow patients to be compared based on their condition, with patients who have more severe conditions being given higher priority.

Next, the program defines a Hospital class to implement the hospital management system. The Hospital class has a private member variable patient\_queue\_ which is a priority queue of Patient objects. The class provides public member functions to add new patients to the queue, remove patients from the queue, and check if the queue is empty.

In the main function, the program creates an instance of the Hospital class and adds some patients to the queue using the AddPatient function. It then processes the patient queue by removing patients from the queue one at a time using the GetNextPatient function and printing their names. The program continues to process the queue until it is empty, as determined by the IsEmpty function.

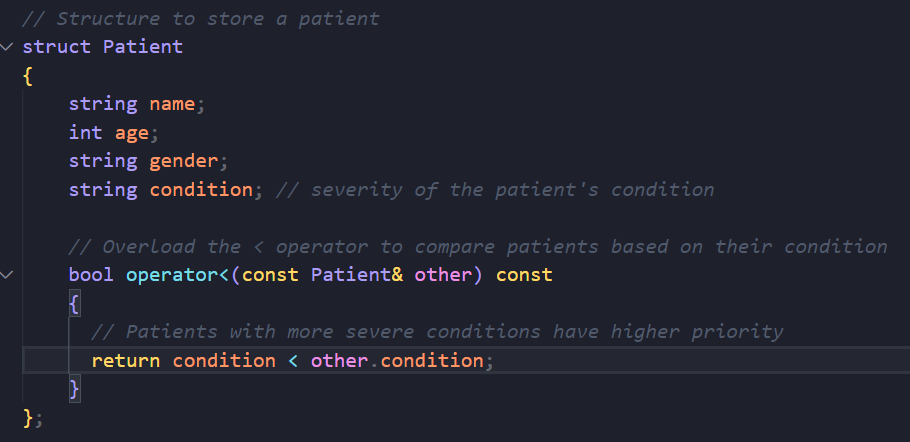
Overall, this C++ program demonstrates how to use a priority queue to implement a hospital management system that manages the order in which patients are seen by doctors based on the severity of their condition.

**Data Structures used:**

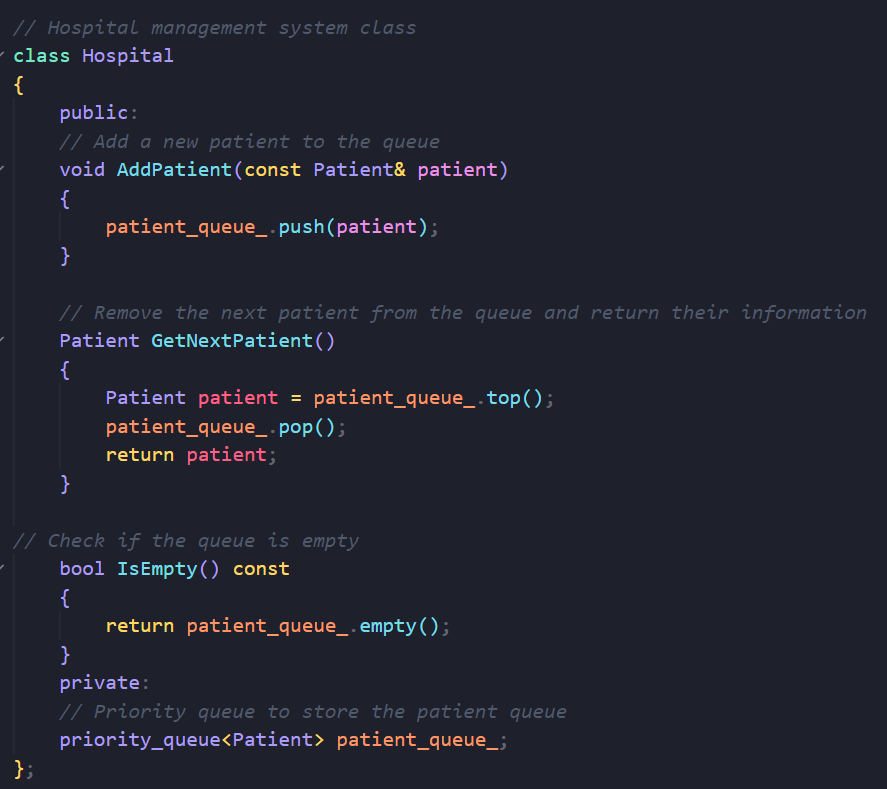
* Structure (used to store patients’ data)
* Queue (used to prioritize patients according to their conditions)

**Screenshots:**

* The Patient struct that stores patient’s details:



* The Hospital class:



* The main function:

