

Bahria University, Islamabad

Department of Software Engineering

# Data Structures & Algorithms Lab

(Fall-2024)

Teacher: Engr. Raheela

Huzaifa Khawar & Aryan

01-131232-034 & 01-131232-014

**LAB PROJECT**

**Comments:**

**Signature**

**Project Title:**

**Hospital Management System Using Data Structures in C++**

* **Objective:**

To develop a **Hospital Management System (HMS)** in C++ that uses efficient data structures to manage patient records, hospital staff, and appointment schedules. The system will allow for the addition, deletion, and retrieval of data, providing critical features such as priority-based patient handling, scheduling, and patient record tracking.

* **Problem Statement:**

Hospitals face challenges in managing patients, especially during emergencies when quick prioritization is needed. Traditional manual systems are inefficient for handling urgent cases, scheduling appointments, and managing patient data. A computerized system can efficiently manage these tasks by leveraging appropriate data structures such as **priority queues**, **linked lists**, and **hash tables**.

* **Scope and Features**

1. **Patient Management:**
   * Add patient records (e.g., name, age, illness, severity).
   * Maintain a priority-based queue for emergencies (e.g., higher severity patients treated first).
2. **Appointment Scheduling:**
   * Schedule appointments with available doctors.
   * Handle cancellations or rescheduling.
3. **Staff Management:**
   * Add, remove, or search for staff (e.g., doctors, nurses).
   * Maintain specialization information for doctors.
4. **Patient Record Retrieval:**
   * Search for a patient’s medical history.
   * Update records after treatment.
5. **Billing System:**
   * Generate bills based on treatments, room charges, and medicines.
6. **Room Allocation:**
   * Allocate rooms to patients dynamically.
   * Maintain availability status.

* **Functionalities in Detail**

**1. Add Patient Record**

* Input patient details (name, age, symptoms, severity level).
* Add to a priority queue where patients with higher severity are served first.

**2. Appointment Scheduling**

* Match a patient with an available doctor based on specialization.
* Store appointments in a queue or linked list for retrieval.

**3. Find Patient Record**

* Search by name or ID using hash tables for quick access.
* Display complete patient history (appointments, treatments, bills).

**4. Room Allocation**

* Use a dynamic array or hash table to track room availability.
* Automatically assign rooms based on availability.

**5. Billing**

* Compute the bill for each patient based on services used.
* Store and display billing details.

**6. Display System Data**

* Display current patients, staff, and room statuses.
* Provide a summary of appointments.
* **System Design**

**Data Structures Used:**

1. **Priority Queue (Heap):**
   * For emergency patient handling based on severity.
2. **Hash Tables:**
   * For fast lookup of patient and staff records.
3. **Linked Lists:**
   * To manage appointments in sequential order.
4. **Dynamic Arrays:**
   * For room allocation tracking.

### **Conclusion**

This **Hospital Management System** is a functional implementation of real-world hospital operations. It uses **priority queues** for emergency handling, **hash tables** for fast lookups, and **dynamic arrays** for resource tracking. The system can be further enhanced with features like patient discharge, billing integration, and database support for persistence. This project showcases the practical use of C++ data structures in healthcare management.