

Quiz 1

COMP6362004 – Data Structures Computer Science

Valid on Even Semester Year 2022/2023

Revision 01

Learning Outcomes

- **LO1** – Explain the concept of data structures and their usage in computer science.
 - **LO2** – Illustrate any learned data structure and its usage in applications.
 - **LO3** – Apply data structures using C.
-

Topic

Data Structure

Sub Topics

- Insert
 - Delete
 - Search
 - Serve
 - Display
-

Case – Hospital Priority Queue Management System

A hospital wants to develop a **priority-based patient queue management system** to improve efficiency in handling emergency cases. The hospital categorizes patients into **four priority levels**:

1. **Critical** (ICU Priority – Highest Priority)
2. **Emergency** (Emergency Room Priority)
3. **Serious** (Urgent Care Priority)

4. **Mild** (General Checkup Priority – Lowest Priority)

Patients with **higher priority** must be served **first**. If two patients have the same priority, they are served based on the **First In First Out (FIFO)** rule. The hospital needs a system that supports **queue management, patient history tracking, and doctor-patient assignments**.

System Requirements

The system consists of **7 main menus**:

1. **Add Patient to Queue**

- This menu allows the user to add a patient to the queue. The user must provide the patient's **ID, Name, Priority Level, and Assigned Doctor**.
- The system ensures the **Patient ID is unique** before inserting the patient into the queue.
- Patients are stored in a **Linked List** based on priority order.
- The patient's details are also stored in a **Hash Table** for fast lookups and a **Binary Search Tree (BST)** for historical tracking.
- Additionally, the system maintains a **Tree Structure** that associates patients with their respective doctors.

2. **Serve One Patient**

- This menu serves the **highest-priority patient** in the queue.
- The system will display a message confirming that the patient is being treated by their assigned doctor.
- If there are no patients, the system will notify the user that the queue is empty.

3. **Serve All Patients**

- This menu processes all patients in the queue based on their priority.
- The system serves each patient one by one and removes them from the queue.

4. **Search Patient History**

- This menu allows the user to search for a patient's historical data using their **Patient ID**.
- The system retrieves the patient's details from the **Hash Table** and displays the information.
- If the patient is not found, the system will display a message indicating that there is no record.

5. Display Patients by Priority

- This menu displays the **current waiting patients**, categorized by priority level.
- The system groups patients under **Critical, Emergency, Serious, and Mild**, making it easy to see the current queue status.

6. Delete All Data

- This menu clears all patient records from the **queue, BST, hash table, and doctor-patient tree**.
- The system will confirm the deletion and notify the user that all data has been erased.

7. Exit

- This menu terminates the program after clearing all data, ensuring a fresh start for the next session.

Validation Rules

- **Patient ID** must be unique.
- **Priority Level** must be: "Critical", "Emergency", "Serious", or "Mild".
- **Doctor Name** must be between **3 and 25 characters**.

Expected Output

Adding Patients Example:

Input:

P001 Alice Critical Dr.Smith

P002 Bob Emergency Dr.Adams

P003 Charlie Mild Dr.James

Output:

Patient Alice (Critical) added successfully.

Patient Bob (Emergency) added successfully.

Patient Charlie (Mild) added successfully.

Displaying Patients by Priority:

User selects option 5

=== Patients Grouped by Priority ===

Critical:

- Alice (ID: P001)

Emergency:

- Bob (ID: P002)

Serious:

(No patients)

Mild:

- Charlie (ID: P003)

Serving One Patient Example:

User selects option 2:

ATTENTION! Patient Alice (Critical) is now being treated by Dr. Smith

Searching for a Patient:

User Input:

Enter Patient ID to search: P001

System Output (If Found):

Patient found: Alice (ID: P001)

Priority Level: Critical

Assigned Doctor: Dr. Smith

System Output (If Not Found):

No record found for Patient ID: P999

Deleting All Data:

User selects option 6

All patient records have been deleted. System reset.

Exiting the Program:

User selects option 7

Exiting program...

Objective

Implement a **modular C program** following this **problem specification** to efficiently manage hospital patients using **Linked Lists, BSTs, Hash Tables, and Trees**.