

# SCD-Lab

lab#3 - Home

Name: Anas Altaf

Roll.no: 22f-3639

Answers:

Task-3:

```
package T_3;
import java.util.*;
class UserManager {
    private HashSet<String> usernames = new HashSet<>();
    void addUser(String username) {
        if (username == null || username.isEmpty()) {
            System.out.println("Invalid username");
            return;
        }
        if (usernames.contains(username)) {
            System.out.println("Username already taken");
        } else {
            usernames.add(username);
            System.out.println("Username added: " + username);
        }
    }
    void removeUser(String username) {
        if (username == null || username.isEmpty()) {
            System.out.println("Invalid username");
            return;
        }
        if (usernames.remove(username)) {
            System.out.println("Username removed: " + username);
        } else {
            System.out.println("Username not found");
        }
    }
    void verifyUsername(String username) {
        if (username == null || username.isEmpty()) {
            System.out.println("Invalid username");
            return;
        }
    }
}
```

```

    }
    if (usernames.contains(username)) {
        System.out.println("Username is taken");
    } else {
        System.out.println("Username is available");
    }
}
}

public class Task_3 {
    public static void main(String[] args) {
        UserManager um = new UserManager();
//Use cases
        um.addUser("Bazurg");
        um.addUser("New User");
        um.addUser("");
        um.addUser(null);
//verification
        um.verifyUsername("New User");
        um.verifyUsername("");
        um.verifyUsername(null);
//Removing
        um.removeUser("Bazurg");
        um.removeUser("");
        um.removeUser(null);
    }
}

```

output:

```

Username added: Bazurg
Username added: New User
Invalid username
Invalid username
Username is taken
Invalid username
Invalid username
Username removed: Bazurg
Invalid username
Invalid username

```

## Task-4:

```

package LabTask;

```

```
import java.util.LinkedList;
import java.util.PriorityQueue;

public class EmergencyRoomQueue {
    private PriorityQueue<Patient> priorityQueue;
    private LinkedList<Patient> regularQueue;

    public EmergencyRoomQueue() {
        priorityQueue = new PriorityQueue<>((p1, p2) ->
Integer.compare(p1.priority, p2.priority));
        regularQueue = new LinkedList<>();
    }

    public static void main(String[] args) {
        EmergencyRoomQueue erQueue = new EmergencyRoomQueue();
        erQueue.addPatient("my User", true);
        erQueue.addPatient("Saim", false);
        erQueue.viewNextPatient();
        erQueue.processNextPatient();
        erQueue.viewNextPatient();
        erQueue.processNextPatient();
        erQueue.processNextPatient();
    }

    public void addPatient(String name, boolean isEmergency) {
        Patient patient = new Patient(name, isEmergency ? 0 : 1);
        if (isEmergency) {
            priorityQueue.add(patient);
        } else {
            regularQueue.add(patient);
        }
    }

    public void processNextPatient() {
        if (!priorityQueue.isEmpty()) {
            Patient patient = priorityQueue.poll();
            System.out.println("Processing priority patient: " + patient.name);
        } else if (!regularQueue.isEmpty()) {
            Patient patient = regularQueue.poll();
            System.out.println("Processing regular patient: " + patient.name);
        } else {
            System.out.println("No patients in the queue");
        }
    }

    public void viewNextPatient() {
        if (!priorityQueue.isEmpty()) {
            Patient patient = priorityQueue.peek();
            System.out.println("Next priority patient: " + patient.name);
        }
    }
}
```

```

        } else if (!regularQueue.isEmpty()) {
            Patient patient = regularQueue.peek();
            System.out.println("Next regular patient: " + patient.name);
        } else {
            System.out.println("No patients in the queue");
        }
    }

    private static class Patient {
        String name;
        int priority;

        Patient(String name, int priority) {
            this.name = name;
            this.priority = priority;
        }
    }
}

```

## Output:

Next priority patient: my User  
 Processing priority patient: my User  
 Next regular patient: Saim  
 Processing regular patient: Saim  
 No patients in the queue

Process finished with exit code 0