**07/07/2025:**

**Q1. Solved: Movie Ticket Booking (with TreeMap)**

**Problem Statement:**

1. **Design a ticket booking system for a multiplex with multiple screens.**
2. **Use TreeMap<Integer, String> to store seat numbers (key) and booking status (value: "Booked" /"Available").**
3. **Preload the system with 5 seats as available.**
4. **Allow the user to book a seat by entering seat number (mark as "Booked").**
5. **Print all seats in ascending order of seat numbers.**
6. **If seat is already booked, throw SeatAlreadyBookedException.**

* **Logical steps:  
  Initialize a TreeMap with 5 seats → "Available"**
* **Take seat number from user (say, seat 3)**
* **if status is "Available", change to "Booked"**
* **else throw SeatAlreadyBookedException**
* **finally print the updated seat map in ascending order**

**Linked List:**

LinkedList is a data structure where elements called as node and they are stored in a linear sequence, but instead of being store in contiguous memory like an array each node points to the next.

Each node typically has to parts, first, data: It stores the value

and second, pointer:stores the reference;address of the next node

In a singly linkedlist, nodes only points to the next node. In a doubly linkedlist, each node has a pointer

to the next and the previous node.

Q3. Solved: Gym Membership Tracker (LinkedHashMap)

Problem Statement

In MonsterGym, monsters register for workouts.

You need to:

store monsterName paidFees in a LinkedHashMap<String, Integer>

insert 4 monsters

print their registration in insertion order

calculate total collection

if any monster pays more than 20,000, mark them as "VIP"

Logical steps:

1. put 4 monsters into LinkedHashMap
2. Loop for printing
3. Add logic for “VIP”
4. Sum total

Logical steps(own):

1. Declare LinkedHashMap
2. Add values
3. Iterate through the LinkedHashMap and first assign “VIP” based on given condition.
4. Calculate the total.
5. Print the total.

Q4. Unsolved: Wildlife Adoption Tracker (TreeMap)

Problem Statement:

**In a wildlife reserve, animals are adopted by donors.**

**You need to build a system to:**

**Store animalID→ adopterName using a TreeMap<Integer, String>**

**Insert at least 5 sample entries**

**Display all animals sorted by their ID**

**If adopter name is "null" or blank, throw InvalidAdopterException**

**Logic:**

1. **Store animalId -> adopterName using a TreeMap<Integer,String>**
2. **Insert atleast 5 sample entries**
3. **Dsiplay all animals sorted by their ID**
4. **If adopter name is “null” or blank, throw InvalidAdopterException**
5. **Print total number of adoptions.**

**Q 4. Delivery Drone Status Monitor (HashMap)**

**Problem Statement:**

**• A startup uses drones to deliver packages in a city.**

**• Each drone has an ID and current battery percentage.**

**• Store droneID batteryPercentage in a HashMap<Integer, Double>**

**• Insert at least 5 sample drones**

**• If battery 15%, mark as "Needs Recharge"**

**• If battery> 100%, throw invalid BatteryException**

**QUEUE:**

Queue is a linear data structure that follows the FIFO principle.

Basic queue ops:

1. Enqueue: Add an element to the rear of the queue.
2. Dequeue: Remove an element from the front of the queue.
3. front/peek: To see what element is at the front without removing it.
4. isEmpty(): Check if the queue is empty.

**QUEUE in JAVA:**

Java’s queue interface provides, the standard queue ops, with common implementation like

1. Linked List.
2. Array Dequeue
3. Priority queue with ordering based on priority.

Queue<Integer> queue = new LinkedList<>();

Queue.add(10);

Queue.add(20);

Queue.add(30);

Sout(queue.poll()) ;

Sout(queue.peek());

Topics to revise:

Queue

File Handling

Byte Stream

Buffer Stream

Character Stream

Problem:

1. First Unique character in a String. You are given a string of lowercase alphabets character arriving one by one. Implement a class first unit, that supports:

a) Add(char ladder) 🡪 It adds a character to the string.

b) Show(first unique) 🡪 It will return the first character that has appeared exactly once in the string. So far I no such character exists return #. You must keep the order of the character using a LinkedHashMap to track their frequency.