BlockBloom (Recruitment Tasks)

Task - 1:

List down situations (atleast 2) you have personally encountered where centralization has been a problem. Explain how decentralization will solve the problem in that situation.

Create a **PDF** for this task.

Task - 2:

This task is related to OOP.
You are tasked with designing a **Moving Booking System** using Object-Oriented Programming principles in C++/Java on a **command-line interface**(that is, just interact using text printed using cout/System.out.print()). The system will allow customers to book or cancel tickets for movies. Apart from customers, there is a manager of this system who can add and remove movies from the system. You can implement it as creatively as you can, but take care of all edge cases, for example, the system should not allow a customer to book tickets if all seats have already been filled or if he does not have sufficient amount. Ensure that the manager does not add two movies with overlapping time. Ensure one customer doesn't book/cancel tickets of another (so you need a password).

Necessary functions: The customers must be able to see the list of all movies, book and cancel ticket. The manager must be able to add, remove movies (after movie has ended) and see the list of customers who have booked ticket for a movie.

Making life easier: You can assume an upper bound on the number of movies that the manager will add. Movies are identified by their ID. You can also assume a fixed number of customers N who use this system, they are identified with their ID. You can initialize the system with one Manager and N number of customers along with their attributes. Initialize account balance for manager as 0 and the account balance for customers as some X. These assumptions will help you to easily maintain fixed sized arrays to manage things. Array index can be used as ID 🙂

Here are some tips for implementation. You need not stick to this, but it's just a hint for you if you feel stuck:

Create a Movie class with attributes like ID, title, duration, start time, capacity (max number of viewers), number of tickets booked, ticket price etc and methods like bookTicket() and cancelTicket().

Create a Manager class with password and account balance as attributes.

Create a Customer class with ID, password, account balance as attributes.

Try to make use of OOPs principles. Keep attributes as public, private or protected as necessary. You might want to use static variables.

Make a **.txt file**. In the beginning, list down assumptions you have made and a brief guide on how to interact with your system. Copy your code to the same .txt file below the assumptions and instructions.

Submission:

Upload these 2 files to Google Drive and share the folder link in the below google form. Make sure to give view access to your submission!!!!

Google form: Form

“Remember that we are not expecting you to be an expert/make a flawless submission. We want to see your enthusiasm and also make you familiar with the foundational concepts needed for the project.

However, if we see an AI-generated submission/plagiarism, we will right away discard you as a mentee.”

Deadline: 15 December 2024 EOD(strict)

Best of Luck!