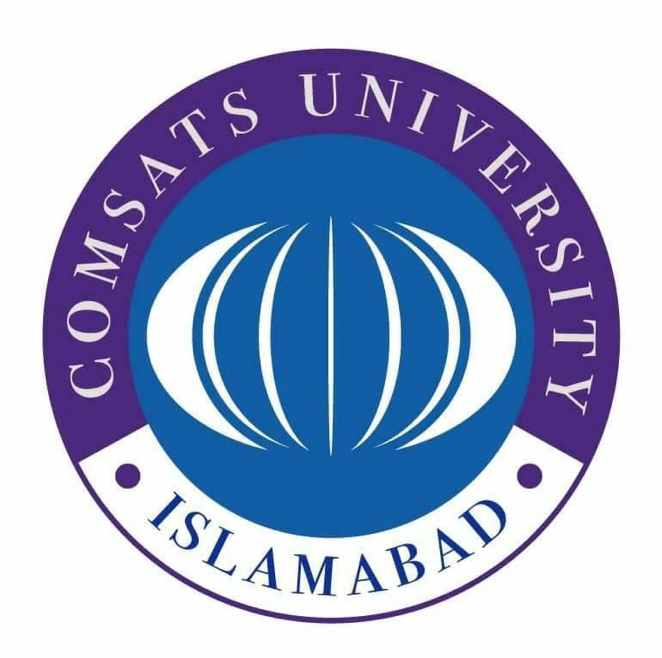
**Assignment 1**

**“Lab 5 — Cinema Management System”**

****

**Submitted to:** Sir Shahid Bhatti

**Submitted by:** Muhammad Ibrahim

**Roll No.** SP25-BCS-101

**Section:** B

**Subject:** Object-Oriented Programming

**Date Of Submission:** October 10, 2025

**DEPARTMENT OF COMPUTER SCIENCE**

**COMSATS UNIVERSITY ISLAMABAD,**

**LAHORE CAMPUS**

# Implementation:

## Seat Class

* **Attributes and Constructor:** Defines seat ID, type, price, and availability; constructor initializes these with availability set to true.
* **Booking and Cancellation Methods:** bookSeat() and cancelBooking() manage the seat’s availability status.
* **Accessors and Mutators:** Provides getters and setters for seat properties.
* **Utility Methods:** toString() returns formatted seat information; equals() compares seats based on their ID.

## Screen Class

* **Attributes and Constructors:** Manages a 2D array of Seat objects, screen name, seat count, and row lengths; supports default and custom initialization with predefined seat pricing.
* **Booking and Cancellation Methods:** Allows booking or canceling seats by coordinates or seat ID, with bounds checking for validity.
* **Accessors and Utilities:** Provides methods to get seat counts, available seats, seats by type, and first available seat; also includes methods to display seats in detailed or compact layouts.
* **Modifiers and Helpers:** Supports modifying row seat types and prices, internal helper methods manage pricing, seat types, row lengths, and coordinate validation.

## Cinema Class

* **Attributes and Constructor:** Maintains an array of Screen objects, cinema name, and number of screens; constructor initializes default screens.
* **Accessors:** Provides methods to get cinema name, screen count, list of screens, and retrieve a screen by name.
* **Forward Operations:** Enables booking and canceling seats on a specific screen by forwarding requests to the corresponding Screen object.
* **Aggregated Information & Display:** Computes total seats, available seats, and seats by type across all screens; includes methods to display cinema layout in both compact and detailed formats

## CityCinema Class

* **Attributes and Constructor:** Manages a collection of Cinema objects for a city, storing the city name and the number of cinemas; constructor initializes the city.
* **Cinema Management:** Supports adding, removing, and preloading cinemas; provides accessors to retrieve cinemas by name and get cinema count.
* **Forwarded Booking and Cancellation:** Delegates booking and canceling requests from city level to the appropriate cinema, screen, and seat, with console feedback on success or failure.
* **Aggregated Information & Utilities:** Computes total seats, available seats, and seats by type across all cinemas; includes a VIP seat finder and report generation for city-wide cinema layouts.

## Cinema Demo (CityCinema Demo)

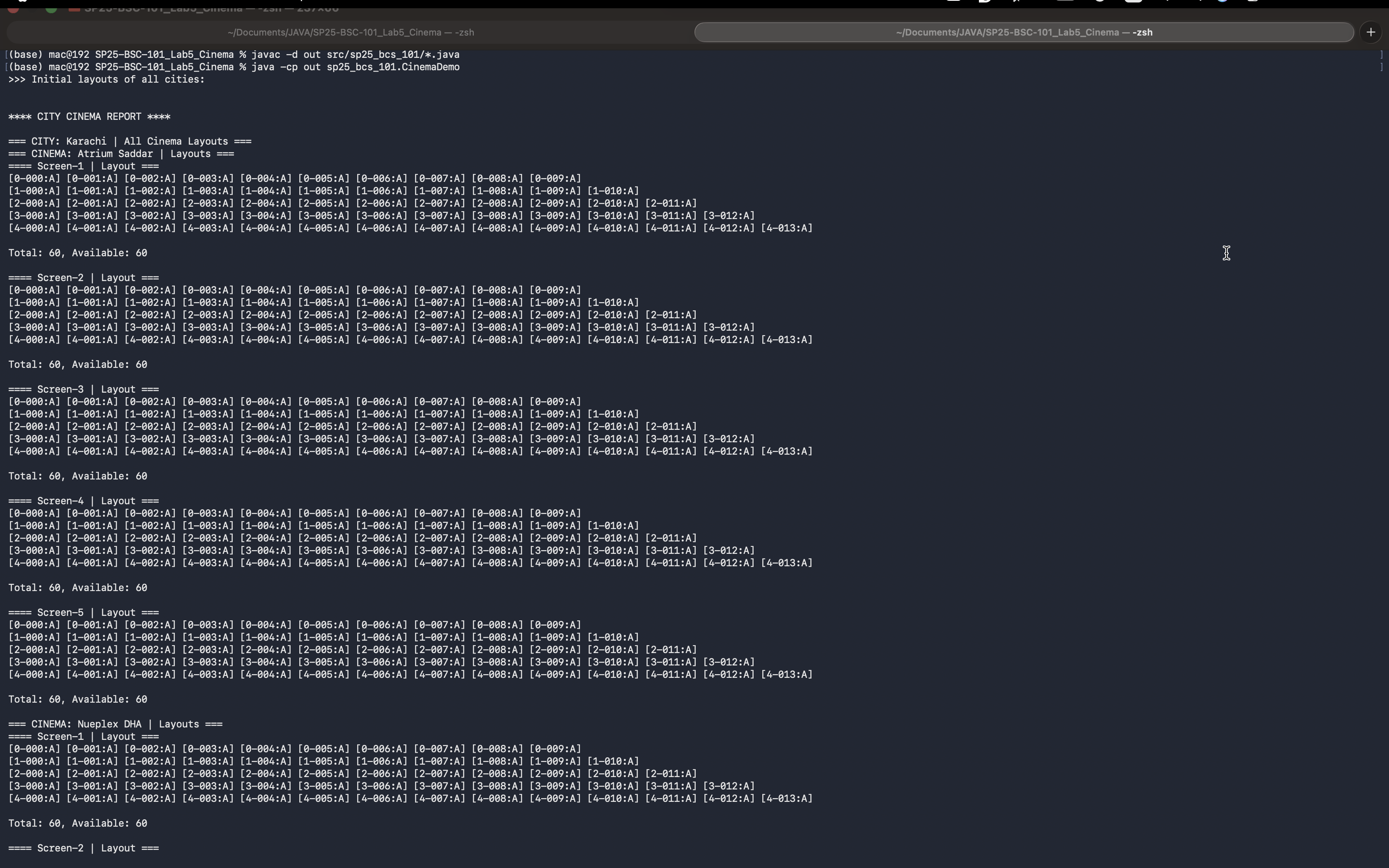
* **Purpose:** Demonstrates the functionality of the CityCinema system across multiple cities, cinemas, screens, and seats.
* **Initialization:** Creates CityCinema objects for Karachi, Lahore, and Islamabad and preloads them with realistic cinemas.
* **Operations Performed:** Simulates workflow, including booking a seat, attempting duplicate booking, cancelling a booking, and searching for VIP seats.
* **Output & Reporting:** Displays city-wide cinema layouts before and after operations, prints VIP seat availability, and provides a final summary of total and available seats.

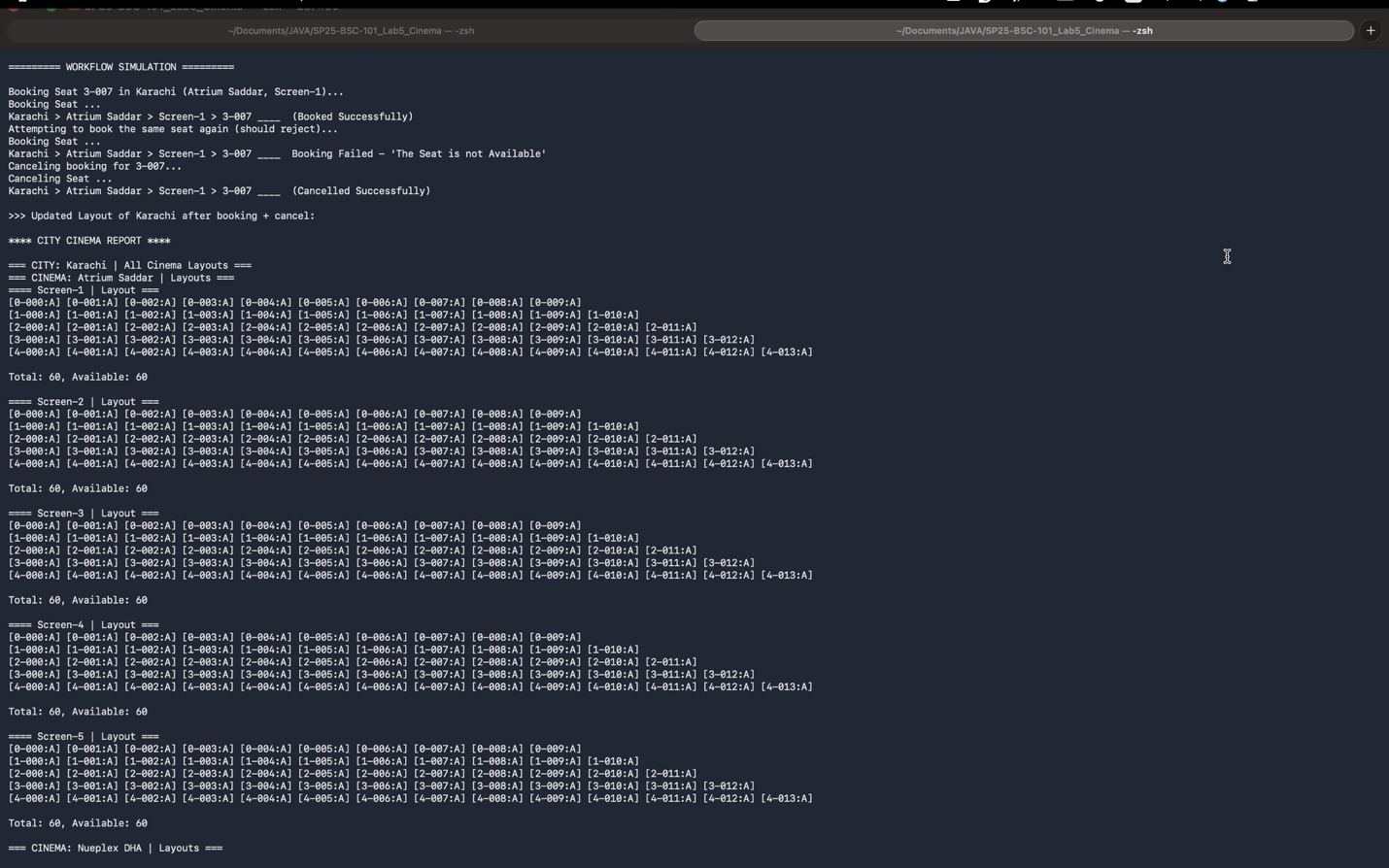
## Seat Demo

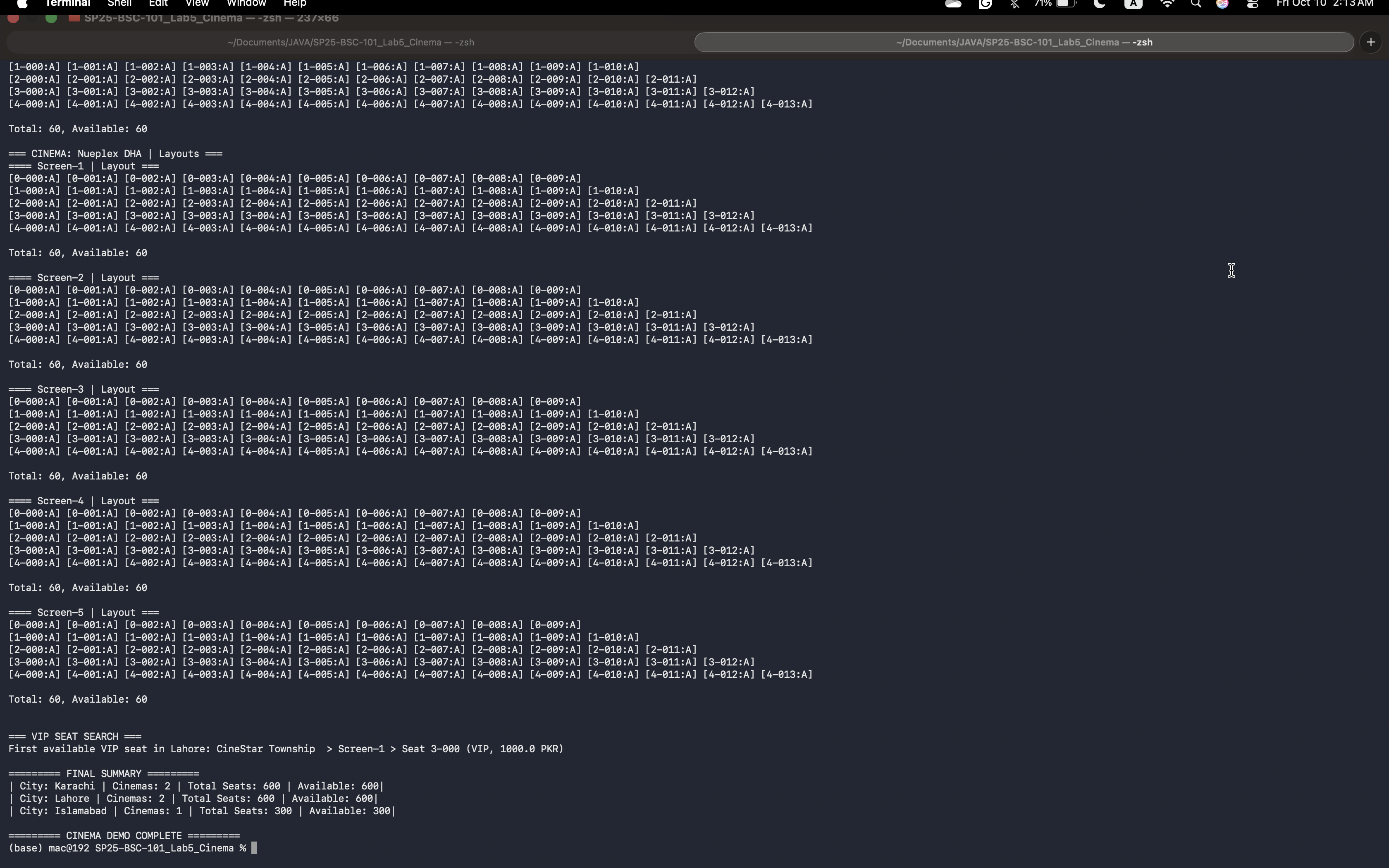
* **Purpose:** Demonstrates the functionality of the Seat class, including creation, booking, cancellation, and price modification.
* **Operations Performed:** Tests booking and rebooking logic, cancels a booked seat, and updates seat prices, displaying results after each operation.
* **Output:** Uses toString() to show detailed seat information and provides console messages to indicate the success or failure of operations.

# **Sample run output**

## Cinema Demo (City Cinema Demonstration)







## Seat Demo

