|  |
| --- |
| Serial No: |
| **Sessional Exam-I** |
| **Total Time: 1.5 Hour** |
| **Total Marks: 20** |
|  |

|  |
| --- |
| **CL1004: Object Oriented**  **Programming- Lab** |
| Tuesday, 11th March, 2025 |
| **Course Instructor** |
| Hifza Umer |

|  |
| --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Student Name Roll No. Course Section Student Signature |

**Note:** Plagiarism (from some else or the internet) will lead to zero marks in the exam.

**Question 01: (20 M arks)**

**Hotel Reservation System**

A **Hotel Reservation System** that facilitates smooth hotel room management, customer reservations, and payment handling. The system enables hotels to **initialize and manage rooms**, handle customer bookings and cancellations, process payments and refunds, and analyze reservation trends for improved efficiency. It includes key functionalities like **room availability tracking, reservation handling, and revenue calculation** to enhance the booking experience for both hotel staff and guests.

**1. Structure Design:**

**(A) Hotel Structure**

* **Represents a hotel with essential details.**

**Attributes:**

* hotelID (int)
* hotelName (string)
* location (string)
* totalRooms (int)
* availableRooms (int)
* roomAvailability[totalRooms] (array of booleans)

**Methods:**

* initializeHotel(): Initializes hotel details (e.g., "Grand Palace, Location: New York, 200 rooms").
* displayHotelInfo(): Displays hotel details (e.g., "Grand Palace, Available Rooms: 150").
* checkRoomAvailability(): Returns the number of available rooms (e.g., "Rooms left: 25").
* updateRoomAvailability(): Updates room status when reserved/canceled (e.g., "Before: 120 - After: 119").

**(B) Customer Structure**

* **Stores customer details for reservations.**

**Attributes:**

* customerID (int)
* name (string)
* contact (string)

**Methods:**

* addCustomer(): Takes input and stores customer details (e.g., "Ali Rizwan, ID: CUST1021").
* displayCustomerInfo(): Displays customer details.

**(C) Reservation Structure (Nested in Hotel and Customer)**

* **Manages booking and room assignments.**

**Attributes:**

* reservationID (int)
* customer (Customer)
* hotel (Hotel)
* roomNumber (int)
* paymentStatus (bool)

**Methods:**

* createReservation(): Assigns a room and reserves it (e.g., "Room 305 booked, Reservation ID: R12345").
* cancelReservation(): Cancels the reservation and updates availability (e.g., "Room 305 available again").
* sortReservations(): Sorts reservations by hotel name, customer name, or date (e.g., "Sorted by Guest Name").

**(D) Payment Structure (Nested in Reservation)**

* **Handles transactions and verification.**

**Attributes:**

* paymentID (int)
* reservationID (int)
* amount (float)
* status (bool)

**Methods:**

* processPayment(): Handles payment verification and transaction (e.g., "$450 charged, Payment Successful").
* refundPayment(): Issues a refund if cancellation occurs (e.g., "$450 refunded").

**(E) HotelManager Structure**

* **Manages multiple hotels and reservations.**

**Attributes:**

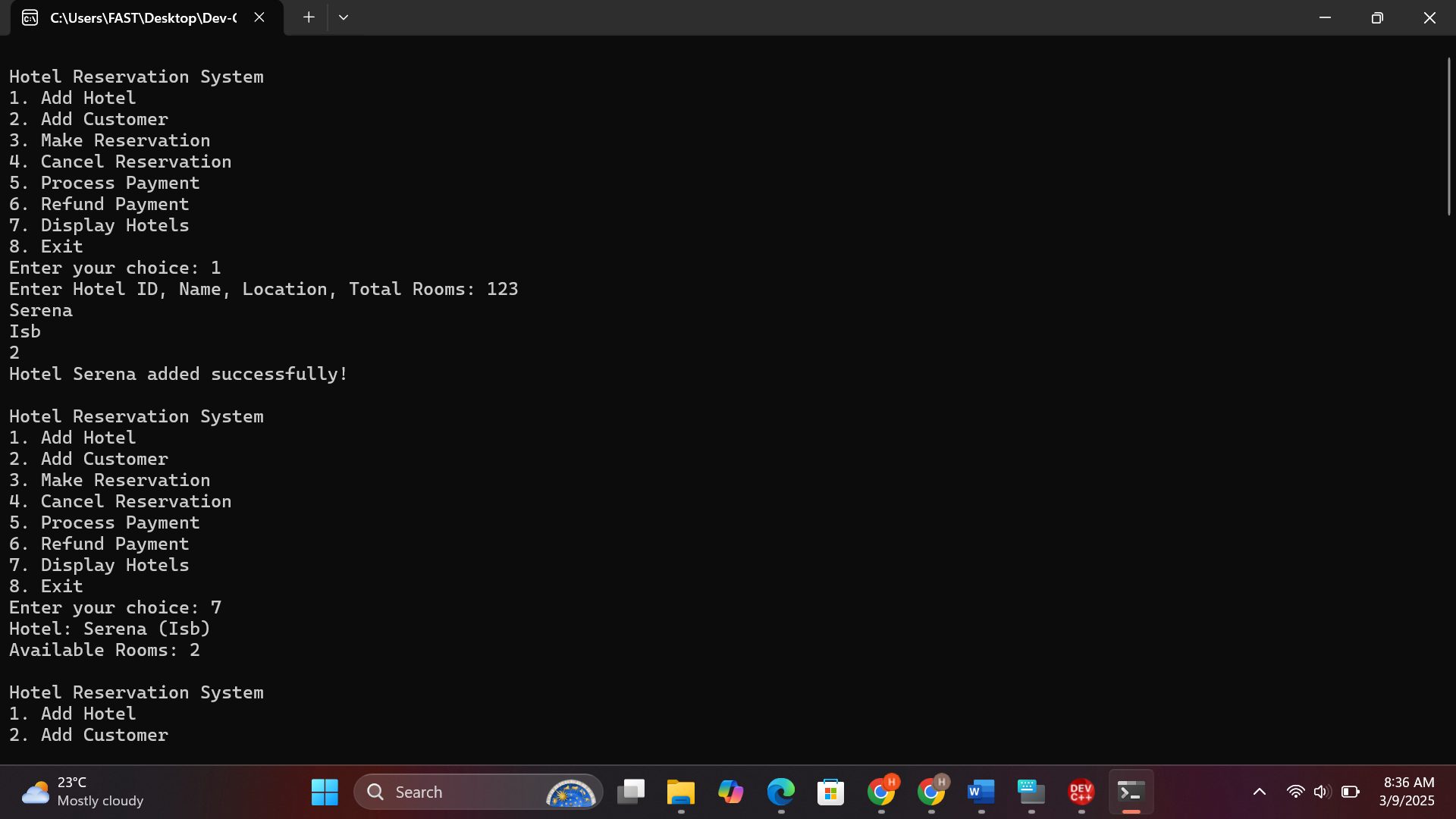
* hotels (array of Hotel)
* reservations (array of Reservation)

**Methods:**

* addHotel(): Adds new hotel data (e.g., "Hotel Sea View added: Miami, 150 rooms").
* searchHotels(): Searches hotels based on location, room availability, or rating (e.g., "Found 3 hotels in Miami").
* displayAllHotels(): Displays available hotels.
* calculateTotalBookings(): Computes total reservations made (e.g., "Total rooms booked: 85").
* calculateRevenue(): Sums up payments received (e.g., "Total revenue: 85 × $450 = $38,250").
* sortHotelsByAvailability(): Sorts hotels based on free rooms.

**(F) Global Functions**

1. calculateTotalRevenue(): Computes the total earnings from bookings (e.g., "Revenue = 200 × $400 = $80,000").
2. calculateCancellationRate(): Calculates cancellation percentage (e.g., "(12/100) × 100 = 12%").
3. findMostBookedHotel(): Determines which hotel has the highest reservations (e.g., "Hotel Grand Palace: 95% rooms booked").
4. predictRoomAvailability(): Uses probability to estimate future occupancy (e.g., "85% rooms likely to be booked next Saturday").
5. applyDiscount(): Adjusts price dynamically based on demand (e.g., "Low demand → Price reduced by 10%").

When a hotel booking is added, an object of HotelBooking is created and stored in the dynamically allocated array of bookings. When a guest checks in, the program searches for the booking by its ID and updates its status accordingly. The displayBookings function iterates through all stored bookings and prints their details, including the booking ID, guest name, room number, check-in and check-out dates, room type, price per night, total cost, and booking status (Confirmed, Checked-In, or Canceled). In the main function, the BookingManager class is used to create and manage hotel reservations. Initially, two bookings, "Booking H101" (Emily Brown, Room 202, Deluxe, $150 per night) and "Booking H102" (James Carter, Room 305, Suite, $300 per night), are added to the system. The program first displays the bookings, showing that both are confirmed. Then, the user marks "Booking H101" as checked-in, and the updated list is displayed, reflecting the change. Next, another booking, "Booking H103" (Sophia Wilson, Room 410, Standard, $100 per night), is added to the system. The bookings are then sorted by total cost in ascending order, and the sorted list is displayed again. The expected output clearly shows the booking ID, guest details, total price, room number, and the updated status for each booking A computer screen shot of a black screen

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.