CS1040 – Program Construction

Lab Exercise 8: Implementing a University Study Room Reservation System

Exercise Description:

Imagine that University of Moratuwa has built a new Educational Resources Center (ERC). ERC contains different resources that facilitate student learning process, such as study rooms, discussion areas, computer labs, printing facilities etc. In this exercise, you will design and implement a University Study Room Reservation System to manage reservations for study rooms in ERC, UOM. This exercise will test your understanding of input/output operations, exceptions, and concurrency concepts.

Tasks:

1. Design a `StudyRoom` class with the following attributes:
    - Room number (integer)
    - Capacity (integer)
    - Availability status (boolean)
2. Implement appropriate getters and setters for the attributes.
3. Create a class called `StudyRoomReservationSystem` that will manage the study room reservations.
4. In the `StudyRoomReservationSystem` class, create an array or collection of `StudyRoom` objects to represent the available study rooms in ERC.
5. Implement the following methods in the `StudyRoomReservationSystem` class:
    - `reserveStudyRoom(int roomNumber)`: Reserves the study room with the specified room number if it is available.
    - `releaseStudyRoom(int roomNumber)`: Releases the study room with the specified room number.
    - `displayStudyRoomStatus()`: Displays the status (availability) of all study rooms.
6. Create a custom exception class called `StudyRoomUnavailableException`. This exception should be thrown when a reservation is attempted for a study room that is already occupied.
7. Make sure your program prevents conflicts when multiple students try to reserve or release study rooms simultaneously.
8. In the main method of the `StudyRoomReservationSystem` class, perform the following actions:

- Create multiple study room objects with different room numbers and capacities.
- Display the initial status of all study rooms.
- Simulate concurrent study room reservation and release operations.
- Handle the `StudyRoomUnavailableException` appropriately when a reservation is attempted for an occupied study room.
- Display the final status of all study rooms after the operations.

Note: Be mindful of potential deadlock scenarios and ensure to avoid such conflicts. Your answers to this exercise will be automatically evaluated using CodeRunner. Hence, make sure you use exactly the mentioned names for classes, methods, and exceptions.

Feel free to extend the Study Room Reservation System with additional features and functionalities if you wish.

Good luck, and happy coding!