Object Oriented Programming set-2
End Semester Exam

Instructions

1. This is a **closed book online proctored** exam.
    a. You should not refer to books, notes or online resources.
    b. You should not discuss questions or answers with anyone (including outsiders)
    c. You should have your camera and microphone **ON** at at all times and no headphones
2. Write the solutions clearly and legibly in A4 sheets, using pen (NOT pencil) and at the end of the exam you should submit the scanned copy of your solutions as explained by the faculty
3. Follow all other instructions given by the faculty during the exam

Descriptive Questions (10 Marks each)

Note: For programs add comments for any assumptions made

1. Input: list.csv with the following columns:
    - Title, ISBN, Count, status

    Write the following class: LibraryApp with the following Methods:
    - createTable() - Read the csv file and use the values to populate a table 'Books' in the database 'Library'
    - addBook() - Insert new book in the table
    - showBooks() - Display all available books
    - issueBook() - For a given book, if it is available decrease its count by 1. Mark status as 'not available' if count reaches 0 Use prepared statements to implement this.
    - returnBook() - For a given book,increase its count by 1. Mark status as 'available' if previous count was 0

    Note:
    - Use appropriate arguments for the methods and use additional methods as needed.
    - Main method is optional

2. Create two classes MovieRecord and MovieManager.
    a. The **MovieRecord** class should be designed to capture Movie's details such as title, imdb-id, genre (Horror or Comedy or etc.) and profit. (Note: you might have to revisit the implementation to support some operations listed below. So, leave enough space)
    b. The **MovieManager** class will have data structures to maintain the MovieRecords. The MovieManager should support the following operations. You have to use appropriate collections to support the operations (Hint: Remember that the number of objects won't get duplicated even if you have multiple collections. The collections only maintain references to the objects)
        i. addMovieRecord - adds the movie record to the primary collection and also the additional collections that you may think are necessary.
        ii. retrieveRecordByID - takes imdb ID as input and returns the correct record by ID.
        iii. retrieveHighestProfit - returns the Movie with the highest profit. It should return this in constant time. (Hint: Think what you can do, such that the MovieRecords can be *compared* easily based on the profit)
        iv. retrieveAllByGenre - takes Genre as input and returns all MovieRecords belonging to that Genre
    c. Write the **justification for your choices** above (in plain english)
        i. Explain clearly why you chose that specific type of collection for implementing that specific operation