Exam Week 1: PY B4

OVERVIEW:

In this assignment, you are required to develop a Command Line Interface (CLI) project using Python for a **Book Store Management System**. The project should apply the Python concepts you've learned throughout the course. You are not required to use any Graphical User Interface (GUI), frameworks, or external libraries - everything should run directly in the terminal.

TASKS:

Add Books:

Allow users to add books to the store with at least the following details:

- Title
- Author
- ISBN or Book ID.
- Genre
- Price
- Quantity in stock

You can include additional fields if necessary, such as **Publisher, Year of Publication**, etc.

Prevent Duplicate Books:

Ensure that the same ISBN (or a unique identifier like a Book ID) cannot be assigned to multiple books. This will prevent adding the same book more than once.

View Books:

Display all saved books in a well-organized format, listing the details of each book (title, author, genre, price, stock, etc.).

The list should be easy to read and presented in a neat, user-friendly way.

Save to File:

Store all book information in a file of your choice (e.g., **.txt**, **.csv**, **.json**).

Books should be automatically saved to the file upon addition or modification.

Load from File:

Ensure all previously saved book data is loaded when the program starts.

If no file exists, prompt the user to create a new one or start fresh.

Remove Books:

Provide an option to delete books from the file by their ISBN or Book ID.

Error Handling:

Display meaningful error messages for invalid inputs, such as:

- The book title must be a string.
- The price must be a positive number.
- The quantity must be a non-negative integer.

Provide clear guidance to users on resolving input issues (e.g., invalid ISBN, negative quantity, etc.).

NOTES:

File Structure:

Organize your project into multiple Python files, each dedicated to specific features or functionalities (e.g., add_book.py, view_books.py, delete_book.py, book_data.py, etc.).

Application Design:

Create a simple architecture or design plan for the application. You may use any tool to draw the plan or sketch it by hand and include the image in the project folder. Visual representations of your plan are encouraged.

Modular Code:

Write reusable and maintainable code by encapsulating features within functions. Ensure each task (adding books, searching, removing, etc.) is handled in a separate function or module.

No External Libraries:

Use only Python standard libraries as taught in the course. Avoid third-party packages or frameworks.

Menu System:

Design an interactive menu with an Exit option for easy navigation. All features (adding, viewing, removing, etc.) should be accessible from this menu.

Example of Expected Project Flow:

Start Program: Load existing book data from the file (if available).

Main Menu:

- Add Book
- View Books
- Search Book
- Remove Book
- Exit

User Actions:

The user can select any of the menu options to perform actions like adding new books, viewing the list of books, searching for books by specific details, or removing books.

Each action will trigger a corresponding function or method **(e.g., add_book(), view_books(), search_books(), remove_book())**.

Exit Program: Save all data to the file and exit.

Submission:

Submit your project via GitHub Repository Link. Make sure your repository contains all the necessary files, including Python scripts, a sample data file (e.g., books.csv or books.txt), and your application design plan. **OR** as ZIP file.

Good Luck!