Final Project: Community Digital Library System

Project Duration: 3.5 hours

The Scenario: You're a volunteer developer for a small community library. The head librarian, Sarah, currently uses a system of physical index cards to track every book, DVD, and magazine. It's slow and prone to errors.

Sarah has asked you to build a prototype for a simple, text-based digital library management system. The program will run in the console, and since it's a first draft, all data will be stored in memory (it will be a blank slate every time the program starts).

Your goal is to translate Sarah's needs into a functional Python application.

Core Concepts to be Tested: Your solution will be evaluated on your ability to use:

- **Data Structures:** Choosing the right way to store and organize library items.
- **Control Flow:** Creating logical paths and repeating actions based on user needs.
- **Functions:** Breaking down complex problems into small, reusable pieces of logic.
- **Object-Oriented Programming:** Modeling real-world items (books, DVDs) and concepts (the library itself) as objects that interact with each other.

Project Deliverables: A Librarian's Wishlist

Sarah has provided a list of features she needs, starting from the most fundamental.

Phase 1: The Blueprint for a Library Item

Before the library can have a catalog, you need a standard way to represent any single item that can be borrowed.

Sarah's Needs:

- "I need a blueprint for any item in our collection. Every item must have a **title**, an **author** (or director, for movies), and a unique **item ID** so I can find it easily."
- "Crucially, I need to know its status: is it **on the shelf**, or has it been **checked out**?"
- "I also need a way to get a quick, single-line summary of an item's details."

Your Task:

1. **Design First:** On paper or in a text file, sketch a simple class diagram for this "library item blueprint." What information will it contain, and what actions can it perform?
2. **Implement the Blueprint:** Write the Python code that represents this core library item.

Phase 2: Building the Library Catalog

Now that you can represent a single item, you need a system to manage the entire collection. This will be the heart of the library.

Sarah's Needs:

- "I need a central place—a digital catalog—to hold all of my library items."
- "I must be able to **add a new item** to the catalog. The system should be smart and not let me add an item if another one with the same ID already exists."
- "When a patron wants to borrow something, I need to **check the item out**. This should update its status to show it's no longer on the shelf. The system shouldn't let me check out an item that's already checked out."
- "Similarly, when an item is returned, I need to **check it back in**, updating its status so it's available again."

Your Task:

- Design and build the "catalog" system. It should be able to hold all your library item objects and perform the core functions of adding, checking out, and checking in items as described by Sarah.

Phase 3: Handling Different Media Types

The library's collection is diverse. Different types of media have unique details that need to be tracked.

Sarah's Needs:

- "We have **Books**, and for those, it's very important to know the **number of pages**."
- "We also have **DVDs**, and for those, I need to track the **run time** in minutes."
- "When I look up the details for a Book or a DVD, the summary **must include this extra information** so I can answer patron questions quickly."

Your Task:

- Expand your system to handle these specialized media types. Your existing catalog should be able to manage Books and DVDs alongside generic items without requiring major rewrites. (Hint: Think about the 'is-a' relationship between a Book and a Library Item).

Phase 4: The Librarian's Terminal

The backend logic is complete. It's time to build the user-friendly interface that Sarah will interact with every day.

Sarah's Needs:

- "I need a simple menu that I can use from my computer's terminal. It should present me with a clear list of options."
- "The program needs to keep running so I can perform multiple tasks in a row. There must be a clear way to **exit** the program when I'm finished for the day."
- "From this menu, I should be able to do everything we've planned:
    - Add a new item (and specify if it's a Book, a DVD, etc.).
    - Check an item out using its ID.
    - Check an item in using its ID.
    - See a complete list of **all items** in the library and their current status."

Your Task:

- Build the interactive command-line interface that brings all the catalog's functionality together for the librarian.

Future Feature Requests (Stretch Goals)

If you finish the core requirements, Sarah has some more ambitious ideas for version 2.0.

1. **Overdue Items Report:** Sarah doesn't have a way to track due dates yet, but she'd like a report that simply lists all the items that are currently checked out. Can you add a feature to the menu that generates this list?
2. **Patron Management:** Create a new blueprint for a `Patron` (a library member). A patron should have a name and a list of item ids they currently have checked out. Modify your check in and check out logic to assign an item to a specific patron. This tests the relationship between different objects.
3. **Keyword Search:** Right now, Sarah can only find items by their exact ID. Add a feature that allows her to search for items by a keyword in the title.. The program should display all items whose titles contain the searched keyword.

Submission Guidelines

- Upload a folder with all your project files to github. Provide a link to this github.
- The code should be clean, readable, and include comments explaining your design choices.
- Your program must be able to run from top to bottom without any errors.