Exercise: Library Management System

**Objective:**

Build a console-based Library Management System using C to practice inheritance, polymorphism, the use of abstract and virtual methods with overrides, and interface implementation. This system should allow users to interact with books and media items within a library's catalog.

**Requirements:**

* Implement a console-based user interface for interaction.
* Use inheritance to create a hierarchy of library items.
* Demonstrate polymorphism with method overrides.
* Utilize both abstract and virtual methods in your design.
* Implement an interface for checkable items.

System Components:

1. Library Item Hierarchy:

* Base Class: `LibraryItem`
* Properties: `ID`, `Title`, `YearPublished`
* Methods: `DisplayInformation()` (Abstract)
* Derived Classes: `Book`, `Magazine`, `DVD`
* `Book`: Additional properties like `Author`, `Genre`. Override `DisplayInformation()`.
* `Magazine`: Additional properties like `IssueNumber`, `Category`. Override `DisplayInformation()`.
* `DVD`: Additional properties like `Director`, `Duration`. Override `DisplayInformation()`.

1. Interface: `ICheckable`

* Methods: `CheckOut()`, `CheckIn()`

3. Implementation of Interface:

* Only `Book` and `DVD` items can be checked out. Implement the `ICheckable` interface for these classes.

4. Console UI:

* Implement a simple console application to interact with the library system.
* Allow users to add items to the library, display all items, check out, and check in items.

Exercise Steps:

1. Define the Base Class and Derived Classes:

* Create an abstract base class `LibraryItem` and derived classes `Book`, `Magazine`, and `DVD`. Implement the necessary properties and methods.

2. Implement the Interface:

* Define the `ICheckable` interface with `CheckOut()` and `CheckIn()` methods.
* Implement this interface in `Book` and `DVD` classes.

3. Polymorphism:

* Use virtual and override keywords to demonstrate polymorphism with the `DisplayInformation()` method.

4. Console UI:

* Implement a console application that allows users to add library items, display them, and perform check-out and check-in operations.

Task:

* Implement the Library Management System as described.
* Ensure that you utilize inheritance, polymorphism, abstract and virtual methods, and interface implementation throughout your design.
* Test the system by adding various types of items and performing check-out and check-in operations.