**Title**: Library Management System - Object-Oriented Concepts  
**Subtitle**: Introduction to OOP with Python

**Slide 2: Overview of the Library System**

* **A simple system for managing books and magazines.**
* **Uses core Object-Oriented Programming (OOP) concepts:**
  + **Inheritance**: Reusability of common functionality.
  + **Polymorphism**: Different behaviors for different types of items.
  + **Encapsulation**: Hiding specific attributes like ISBN and issue number.
  + **Abstraction**: Defining common interface for all library items.
* **Lambda functions** and **map()** used for transforming and filtering data:
  + Filtering items based on the publication year.
  + Transforming all titles to uppercase.

**Slide 3: OOP Concepts Used**

* **Inheritance**: Book and Magazine inherit from LibraryItem, which provides common properties like title, author, and year.
* **Polymorphism**: display() method is implemented differently for each class (Book and Magazine).
* **Encapsulation**: ISBN and issue number are encapsulated within their respective classes (Book and Magazine).
* **Abstraction**: LibraryItem is an abstract class that defines the display() method interface, but the actual implementation is provided by Book and Magazine.

**Slide 4: Abstract Base Class**

* **LibraryItem**:
  + This is an **abstract base class** that cannot be instantiated.
  + It defines **common properties** for all library items (title, author, and year).
  + It uses **@abstractmethod** to force child classes to implement the display() method.

**Slide 5: Book Class**

* The Book class inherits from **LibraryItem** and adds an additional attribute: **ISBN**.
* The display() method is overridden to include **ISBN**.
* This class shows how we can extend functionality and add more specific attributes like **ISBN**.

**Slide 6: Magazine Class**

* The Magazine class also inherits from **LibraryItem** and adds a unique attribute: **issue number**.
* The display() method is overridden to include **issue number**.
* This demonstrates how inheritance helps manage different types of library items.

**Slide 7: Using Lambda and Map**

* **Lambda Function**: A small anonymous function that is used to apply transformations to data.
  + Used to **filter** items published after 1950.
  + Used to **map** all titles to uppercase.
* **filter()**: Filters out items based on a condition (year > 1950).
* **map()**: Transforms each element of a collection (converts titles to uppercase).

**Slide 8: Final Code and Output**

* **Code**:
  + After processing the items (filtering and mapping), display them using the display() method.
  + **Output**:
    - **Book Title**: The Great Gatsby, Author: F. Scott Fitzgerald, Year: 1925, ISBN: 12345
    - **Magazine Title**: Time, Author: Time Inc., Year: 2023, Issue Number: 45
    - **Book Title**: 1984, Author: George Orwell, Year: 1949, ISBN: 67890

**Slide 9: Conclusion**

* **Core OOP Concepts**: Inheritance, Polymorphism, Encapsulation, and Abstraction were applied to design the system.
* **Efficient Data Processing**: Lambda functions, filter(), and map() helped in filtering and transforming library items.
* **Simple yet Effective Design**: The design is effective for managing books and magazines and can be extended with more features.