Library Management System using OOP Concepts

OOP with Examples and Explanation

Presented by: S.PAVAN KALYAN

Project Objective

- - Understand and apply OOP principles:
- • Classes & Objects
- • Encapsulation
- • Abstraction
- • Inheritance (optional)
- • Polymorphism (optional)
- - Create a console-based Library Management System

Project Overview

- Features:
- - Add, view, issue, and return books
- - Manage members
- - Prevent over-issuing of books

- Tools:
- - Python / Java
- - Console input/output

Class Design

- 1. Book Class
- - book_id, title, author, total_copies, issued_copies
- 2. Member Class
- - member_id, name, borrowed_books
- 3. Library Class
- - add_book(), issue_book(), return_book()
- - add_member(), remove_member(), view_books()

OOP Concepts Used

- Classes & Objects - Book, Member, Library
- Encapsulation - Private attributes (_attr)
- Abstraction - Method-based internal logic
- Inheritance (Optional) - Subclasses like Student/Teacher
- Polymorphism (Optional) - Method overriding in subclasses

Example: Book Class (Python)

- class Book:
- def __init__(self, book_id, title, author, total_copies):
- self.__book_id = book_id
- self.__title = title
- self.__author = author
- self.__total_copies = total_copies
- self.__issued_copies = 0

- def display_info(self):
- print(f'{self.__title} | Available:

Example: Member Class

```
class Member:
    def __init__(self, member_id, name):
        self.__member_id = member_id
        self.__name = name
        self.__borrowed_books = []

    def borrow_book(self, book_id):
        self.__borrowed_books.append(book_id)
        print(f'{self.__name} borrowed Book ID: {book_id}')
```

Example: Library Class

- class Library:
- def __init__(self):
- self.books = {}
- self.members = {}

- def add_book(self, book):
- self.books[book._Book__book_id] = book
- def add_member(self, member):
- self.members[member._Member__member_id] = member

Sample Console Output

- Welcome to the Library System
- 1. Add Book
- 2. Add Member
- 3. Issue Book
- 4. Return Book
- 5. View Books
- Choose an option: 1
- Enter title: Python 101
- Book added successfully!

Bonus Features (Optional)

- - File Handling: Save data to CSV/text
- - Search Books by Title/Author
- - Exception Handling for invalid inputs
- - Inheritance: Student, Teacher as subclasses of Member

Submission Format

- - File Type: .py or .java
- - Include code comments
- - Optional: Output samples or screenshots
- - Deadline: [Insert Date]

Thank You