



“Python Programming”

Assignment-3

Topic – Object-Oriented Design and Robust Programming in a Library Management System

Submitted by – Meghna Kumar

Roll no - 2501730214

Course – B. Tech CSE (AI & ML)

Section – A

Faculty name- Mr Sameer Farooq

Introduction

This project is a simple command-line Library Inventory Manager made using Python. It helps manage books by storing their details, tracking issued/available status, and saving all data in a JSON file. The program uses object-oriented programming, file handling, and basic error management to create a small but functional library system.

Objectives

- To design classes using OOP concepts.
- To store and load data using JSON.
- To build a menu-driven CLI for library operations.
- To apply exception handling and logging.
- To create a simple, real-world library management solution.

Program Description

The program contains two classes:

Book – stores title, author, ISBN, and status (issued/available).

LibraryInventory – manages a list of books, adds books, searches, issues, returns, and saves data to books.json.

A menu interface allows the user to add, issue, return, view, and search books. JSON is used for permanent storage, and logging/exception handling makes the program reliable. All features are implemented inside one single Python file.

Program Code

```
import json
import logging
from pathlib import Path

logging.basicConfig(
    filename="library.log",
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(message)s"
)

class Book:
    def __init__(self, title, author, isbn, status="available"):
        self.title = title
        self.author = author
        self.isbn = isbn
        self.status = status

    def issue(self):
        if self.status == "available":
            self.status = "issued"
            return True
        return False

    def return_book(self):
        if self.status == "issued":
            self.status = "available"
            return True
        return False

    def to_dict(self):
        return {
            "title": self.title,
            "author": self.author,
            "isbn": self.isbn,
            "status": self.status
        }

    def __str__(self):
        return f"{self.title} - {self.author} | ISBN: {self.isbn} | {self.status}"

class LibraryInventory:
    def __init__(self, file_path="books.json"):
        self.file_path = Path(file_path)
        self.books = []
        self.load_data()

    def add_book(self, book):
        self.books.append(book)
        self.save_data()
        print("Book added!")
        logging.info(f"Book added: {book.title}")

    def search_by_title(self, title):
        return [b for b in self.books if title.lower() in b.title.lower()]

    def search_by_isbn(self, isbn):
        for b in self.books:
            if b.isbn == isbn:
                return b
        return None

    def display_all(self):
        if not self.books:
            print("No books in database.")
            return
        for b in self.books:
            print(b)

    def save_data(self):
        try:
            data = [b.to_dict() for b in self.books]
            with open(self.file_path, "w") as f:
                json.dump(data, f, indent=4)
        except Exception as e:
            logging.error(f"Error saving file: {e}")

    def load_data(self):
        try:
            if not self.file_path.exists():
                self.file_path.write_text("[]")
```

```

        with open(self.file_path, "r") as f:
            data = json.load(f)
            self.books = [Book(*item) for item in data]

    except Exception as e:
        logging.error(f"Error loading file: {e}")
        self.books = []

def menu():
    print("\n==== Library Inventory Manager =====")
    print("1. Add Book")
    print("2. Issue Book")
    print("3. Return Book")
    print("4. View All Books")
    print("5. Search Book by Title")
    print("6. Exit")

def main():
    inventory = LibraryInventory()

    while True:
        menu()
        choice = input("Enter choice: ")

        try:
            if choice == "1":
                title = input("Title: ")
                author = input("Author: ")
                isbn = input("ISBN: ")
                inventory.add_book(Book(title, author, isbn))

            elif choice == "2":
                isbn = input("Enter ISBN to issue: ")
                book = inventory.search_by_isbn(isbn)

                if book and book.issue():
                    print("Book issued.")
                    inventory.save_data()
                else:
                    print("Cannot issue book.")

            elif choice == "3":
                isbn = input("Enter ISBN to return: ")
                book = inventory.search_by_isbn(isbn)

                if book and book.return_book():
                    print("Book returned.")
                    inventory.save_data()
                else:
                    print("Cannot return book.")

            elif choice == "4":
                inventory.display_all()

            elif choice == "5":
                title = input("Enter title to search: ")
                results = inventory.search_by_title(title)
                if results:
                    for b in results:
                        print(b)
                else:
                    print("No matching books found.")

            elif choice == "6":
                print("Exiting program...")
                break

            else:
                print("Invalid choice.")

        except Exception as e:
            print("Error:", e)

if __name__ == "__main__":
    main()

```

Sample Output

```
===== Library Inventory Manager =====
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book by Title
6. Exit
Enter choice: 1
Title: Harry Potter
Author: JK Rowling
ISBN: 67583
Book added!

===== Library Inventory Manager =====
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book by Title
6. Exit
Enter choice: 4
Harry Potter - JK Rowling | ISBN: 67583 | available

===== Library Inventory Manager =====
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book by Title
6. Exit
Enter choice: 2
Enter ISBN to issue: 67583
Book issued.

===== Library Inventory Manager =====
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book by Title
6. Exit
Enter choice: 2
Enter ISBN to issue: 67583
Book issued.

===== Library Inventory Manager =====
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book by Title
6. Exit
Enter choice: 3
Enter ISBN to return: 67583
Book returned.

===== Library Inventory Manager =====
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book by Title
6. Exit
Enter choice: 5
Enter title to search: Harry Potter
Harry Potter - JK Rowling | ISBN: 67583 | available

===== Library Inventory Manager =====
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book by Title
6. Exit
Enter choice: 6
Exiting program...
|
```

Conclusion

This project successfully demonstrates OOP, JSON file handling, CLI design, and error management in Python. It provides a simple and efficient way to manage a small library system and fulfills all assignment requirements.