

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

page1image9370432

**PROG211 – OBJECT-ORIENTED PROGRAMMING 1**

page1image9370432

Title : Mini Library Management System.

Issue Date : Week 2

Due Date : Week 4

Lecturer/Examiner : Mr. Amandus coker

Name of Students : Makiba Hassan

Student ID No. : 905005090

Class : BIT 1102F

Year/Semester : 2/1

Academic Honesty Policy Statement

page1image9370432

I, hereby attest those contents of this attachment are my own work. Referenced works, articles, art, programs, papers or parts thereof are acknowledged at the end of this paper. This includes data excerpted from CD-ROMs, the Internet, other private networks, and other people’s disk of the computer system.

Student’s Signature: Date:

page1image9370432

LECTURER’S COMMMENTS/GRADE:

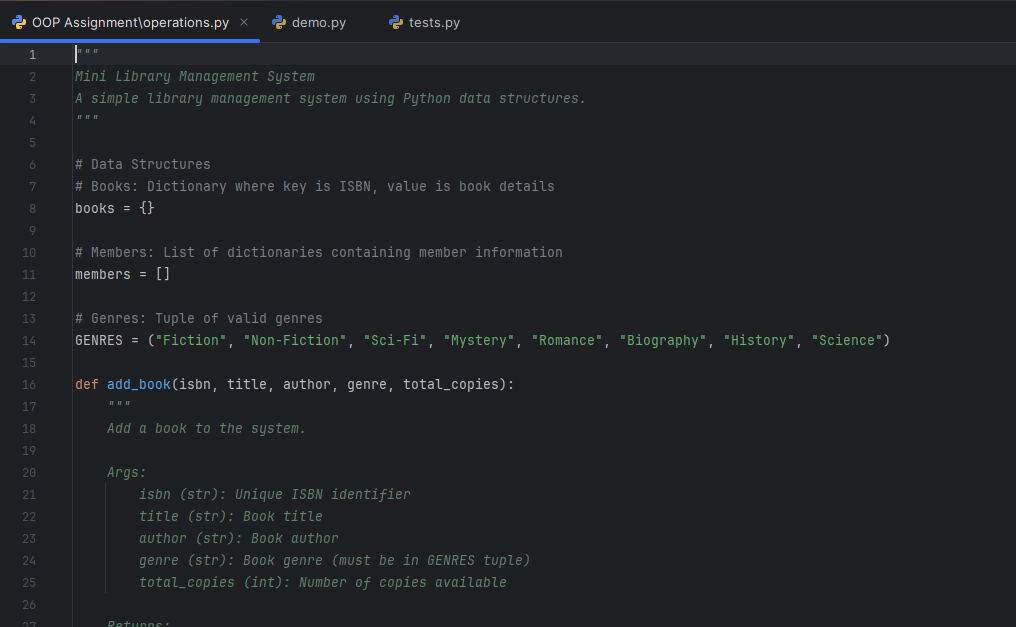
for office use only upon receive

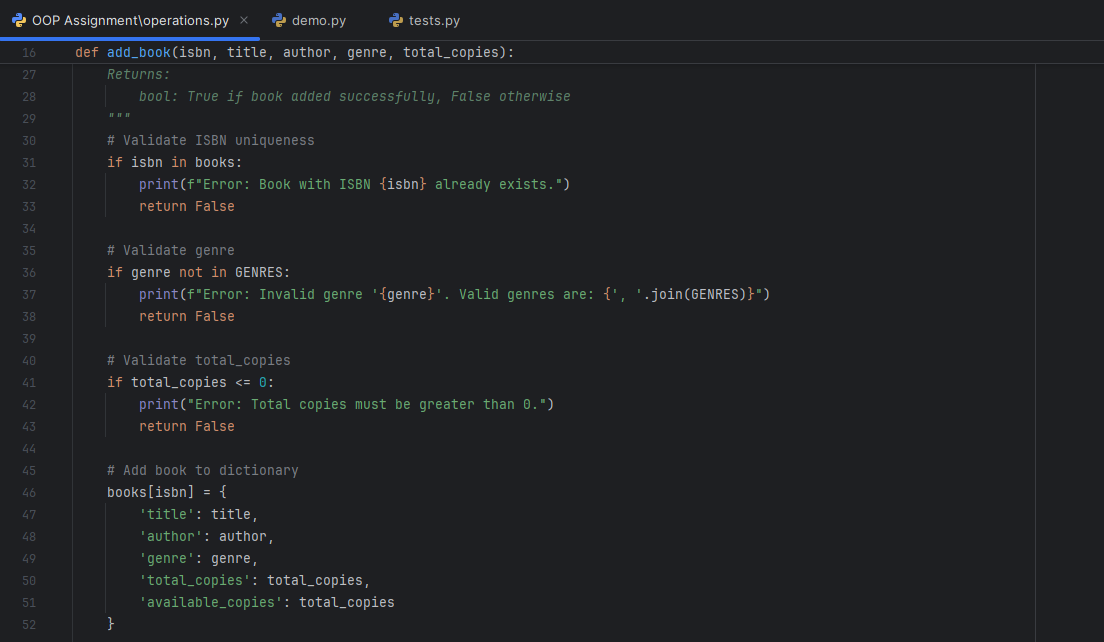
Remark

DATE:  
 TIME:

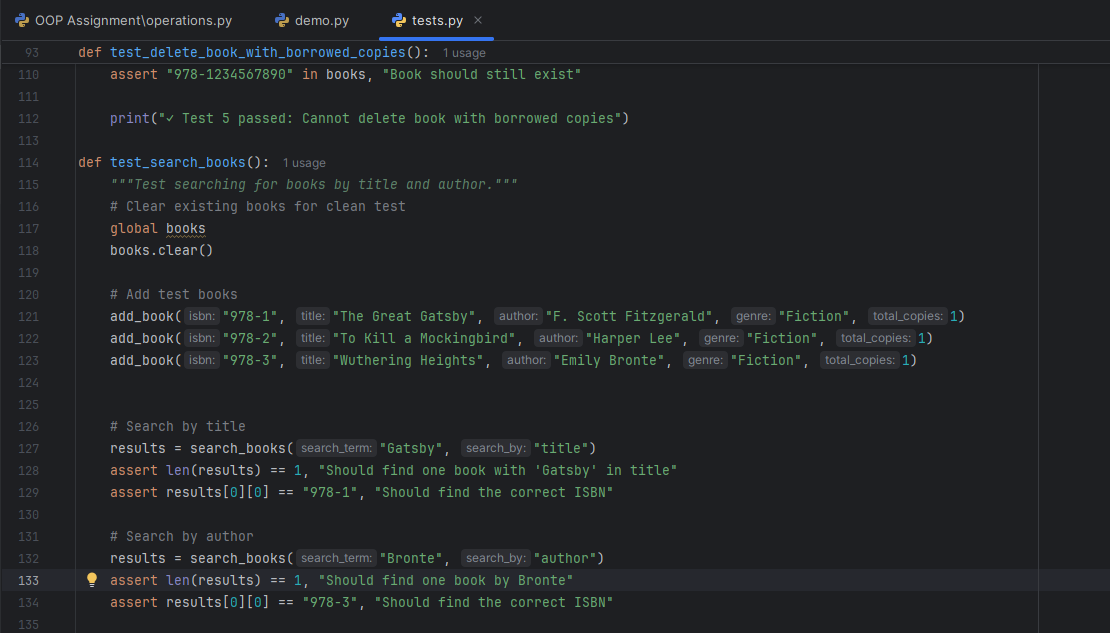
RECEIVER’S NAME:

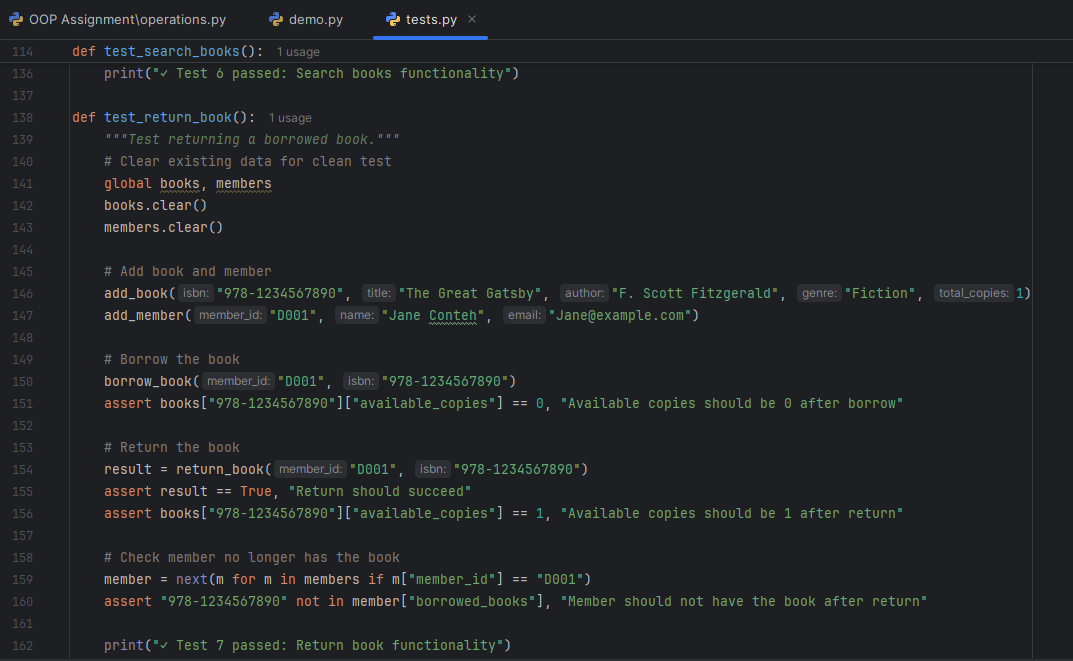
**Operations.py**

****

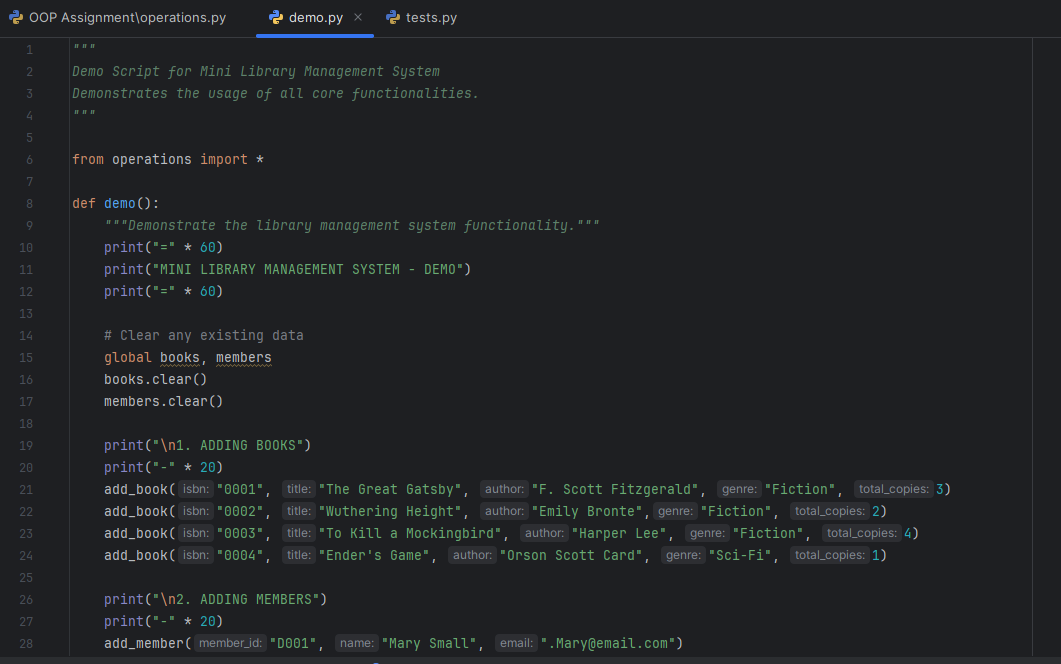
****

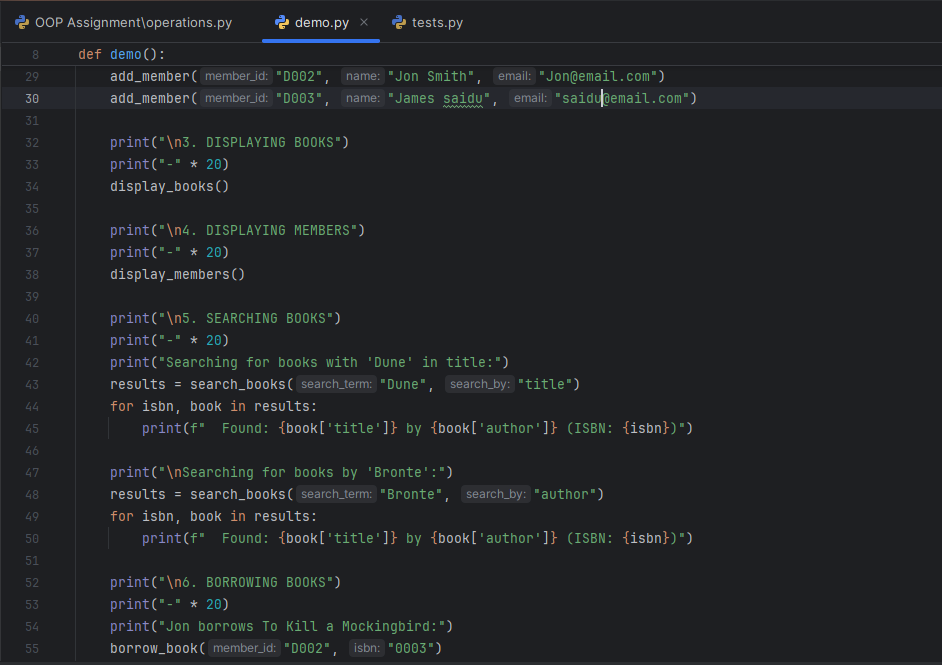
**Test.py**

****

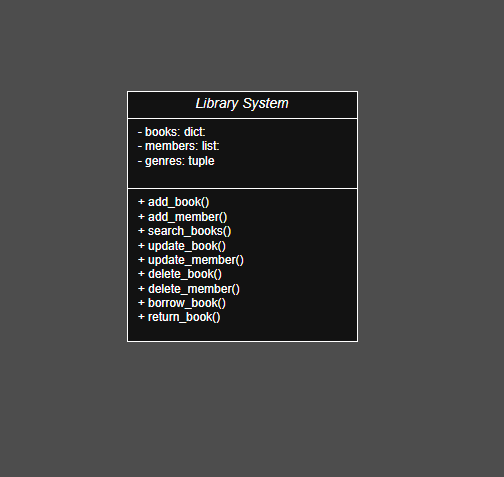
****

**Demo.py**

****

****

**UML Diagram showing structures and functions**



**Design Rationale**

1. **The use of dictionary**

Dictionaries were used for books because they store data as key-value pairs, which makes searching and updating very fast. Each book’s ISBN is unique, so it was used as the key, while the value stores the book details such as title, author, genre, and total copies.

**Why we use dictionary**

* Quick access to book details using ISBN.
* Easy to update and delete specific books.
* Prevents duplicate book entries.

**Example:**

books = {

"001": {"title": "Python Basics", "author": "John Smith", "genre": "Non-Fiction", "total\_copies": 3}

}

1. **The use of list**

A list was used for members because it allows storing multiple items in order, and each member can be represented as a dictionary. Each member dictionary contains attributes such as member ID, name, email, and borrowed books.

**Why we use list**

* Simple and flexible structure for adding or removing members.
* Easy to loop through and search by member ID.
* Works well for systems with a moderate number of users.

**Example:**

members = [

{"member\_id": 1, "name": "Makiba Hassan", "email": "makiba@example.com", "borrowed\_books": []}

]

1. **The use of tuple**

A tuple was used for genres because it contains a fixed set of values that should not be changed (e.g., Fiction, Non-Fiction, Sci-fi). Tuples are immutable, which means their value cannot be modified which is perfect for storing constant data.

**Why we use tuple**

* Ensures that valid genres are consistent throughout the system.
* Prevents accidental editing or addition of invalid genres.
* Saves memory since tuples are lightweight.

Example:

genres = ("Fiction", "Non-Fiction", "Sci-Fi", "Romance", "Mystery")