



School of Engineering & Technology	
Department: SOET	Session: 2025-26
Program: BCA (Cyber Security)	Semester: I
Course Code: ETCCCPP103	Number of students:
Course Name: Problem Solving with Python	Faculty: Neha Kaushik

## Assignment: (Unit-2)

### Submission Instructions:

- Submission Deadline:** Assignments must be submitted within one week of the assignment's release date.
- Submission Platform:** All assignments are to be submitted via the Learning Management System (LMS) or Moodle (<https://lms.krmangalam.edu.in/>).
- GitHub Link:** You must provide a link to your GitHub repository with your submission (Optional).
- 
- Individual Submission:** Assignments are to be completed and submitted by each individual student.
- Formatting:** All assignments must adhere to the specific format shared in class.

### Mini Project Assignment: Library Book Manager

**Course:** Programming for Problem Solving using Python

**Assignment Title:** Library Inventory & Borrowing System

**Assignment Type:** Individual

**Estimated Duration:** 8–10 hours

**Weightage:** 15% of course grade

---

### Real-World Problem Context

University libraries handle hundreds of books and must manage availability, borrowed books, and student records. Traditional registers and manual tracking lead to mistakes, delays, and loss of records.

This project focuses on developing a **Python-based Library Book Manager CLI** that maintains book data, student borrowing records, and searching functionality using core Python fundamentals.



## Learning Objectives

After completing this assignment, students will:

- Manage and store book data using Python **lists, dictionaries, tuples, and sets**
  - Use **conditionals, loops, functions, recursion, and lambda expressions**
  - Apply **searching and filtering with list comprehensions**
  - Work with **data entry, updating records, and reporting**
  - Format CLI output using strings, tabs, and tables
  - Practice **modular program design & menu-driven control flow**

## Assignment Tasks

## Task 1: Project Setup & Menu Screen

- Create project folder: library\_manager/
  - File: library.py
  - Add header (name, date, project title)
  - Print welcome screen and menu:
    - Add Book
    - View Books
    - Search Book
    - Borrow Book
    - Return Book
    - Exit

## **Deliverable:** Project structure + basic CLI menu

## Task 2: Book Data Entry

- Input book details manually:
    - Book ID, Title, Author, Copies
  - Store in dictionary of lists, e.g.:

books = {

"B101": {"title": "Python Basics", "author": "Guido", "copies": 5},



"B102": {"title": "DSA", "author": "Cormen", "copies": 3}

}

- Allow user to enter at least **5 books**

**Deliverable:** Book input & storage logic

### Task 3: Display & Search Books

- Display books in tabular format
- Implement searching by:
  - Book ID
  - Title (substring match)
- Use **functions** for searching
- Print "**Book Found / Not Found**" message

**Deliverable:** Working search + formatted book list

### Task 4: Borrowing System

- Ask for:
  - Student name
  - Book ID
- Check if book exists and **copies > 0**
- Reduce available copies
- Store borrowing info in dictionary:

borrowed = {"Amit": "B101", "Neha": "B102"}

- Use if-elif-else logic
- Display confirmation or error message

**Deliverable:** Borrow feature + updated stock

### Task 5: Return Book + List Comprehension

- Student enters name + Book ID
- Increase copies
- Remove borrowing entry



- Use **list comprehension** to display all borrowed books

Example comprehension:

```
borrowed_list = [f'{student} -> {book}' for student,book in borrowed.items()]
```

**Deliverable:** Return system + borrowed list

### Task 6: User Loop & Exit

- Program runs in a **while loop with menu**
- Option to perform multiple operations until user selects exit
- Use:
  - f-strings
  - \n, \t
  - proper formatting

**Bonus:** Save / load records from a text/CSV file

**Deliverable:** Complete working program

---

### Submission Checklist

- library.py script with all tasks completed
- Minimum:
  - 5 stored books
  - 2 borrow actions
  - 1 return action
- Clean formatting & comments
- Menu-driven program screenshots

**Bonus:**

- GitHub repo + README
  - CSV storage implemented
-



### Evaluation Rubric

Criterion	Weight	Excellent (5)	Good (4)	Fair (3)	Poor (2)
Git setup	5%	Fully structured repo	Basic repo	Unorganized	None
Problem framing	10%	Real-world clarity	OK	Weak	Missing
Data handling	15%	Full CRUD functions	Partial	Weak	None
Control logic	20%	Clean loops/conditions	Some errors	Basic	Wrong logic
Functions	15%	Modular, documented	OK	Few functions	No functions
Lists & Dicts	15%	Proper and advanced	OK	Limited use	Incorrect
Output formatting	10%	Table, clear UI	OK	Messy	Poor
Code quality	10%	Commented & clean	OK	Untidy	Copied

---

### Academic Integrity Policy

- Work **must be original**
- Plagiarism = **Zero marks**
- Cite references if used