

21 October, 2025

Abriel Elijah Conteh - 905005069

Design Rationale: Mini Library Management System:

Overview

This project was designed using core Python data structures (dictionaries, lists, and tuples) to manage a small library system without using classes or object-oriented programming.

Data Structure Choices

1. Dictionaries for Books:

- Each book is stored with its ISBN as the key for fast lookup.
- Book attributes (title, author, genre, total copies) are stored as key-value pairs.
- This structure supports efficient updates and deletions by ISBN.

2. Lists for Members:

- Members are stored in a list because the dataset is small and easy to iterate.
- Each member is represented as a dictionary, allowing flexible updates to their information and borrowed books.

3. Tuples for Genres:

- The tuple `GENRES` defines a fixed list of valid genres.
- Tuples are immutable, ensuring genre categories remain constant.

Functional Design

- CRUD Functions: Each operation (Create, Read, Update, Delete) directly manipulates global data structures.
- Borrow/Return: These mimic real-world borrowing:
 - Decrease available copies when borrowed.

- Increase when returned.
- Limit each member to three active loans.

Data Integrity

- Unique ISBNs prevent duplicate books.
- Members cannot be deleted if they still have borrowed books.
- Borrowing checks ensure available copies and respect the 3-book limit.

Reasoning

This functional approach demonstrates understanding of how lists, dictionaries, and tuples interact while maintaining data flow between functions, reflecting fundamental programming logic.

Conclusion

The system is designed for simplicity, readability, and clear function-driven data control a foundational step before advancing to object-oriented programming.