

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**PROG211 – OBJECT-ORIENTED PROGRAMMING 1**

Title : Individual Assignment

Issue Date : Week 2

Due Date : Week 4

Lecturer/Examiner : Mr. Amandus Coker

Name of Student/s : Abu Bakarr Mansaray

Student ID No. : 905004218

Class : BBIT1101F

Semester/Year : 3/2

Academic Honesty Policy Statement

I/We, 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: 25th October 2025

for office use only upon receive

LECTURER’S COMMMENTS/GRADE:

Remark

DATE :  
 TIME :

RECEIVER’S NAME :

# Rationale for Library Management System

## 1. Data Structure Choices

### Dictionary for Books

The system uses a dictionary (dict) to store all book records, with each book’s ISBN number serving as the unique key. This structure is chosen because dictionaries provide O(1) average-time complexity for search, insertion, and deletion. It allows for fast lookup and ensures unique identification of books through their ISBNs. Updates can also be done efficiently through key-based access.

### List for Members

Members are stored in a list of dictionaries, where each dictionary holds details about a member and their borrowed books. Lists are simple to traverse and allow dynamic growth, making them suitable for small datasets. Although lookups are O(n), this is acceptable given the manageable size of member data.

### Tuple for Genres

Genres are stored as a tuple to maintain immutability and consistency. Tuples are ideal for fixed sets of categories (e.g., Fiction, Non-Fiction, Programming) and provide memory efficiency while ensuring the genres cannot be modified accidentally at runtime.

## 2. Functions and Real-World Modeling

Each function in the system represents a real-world library operation. For example:

* • add\_book(): Cataloging a new book into the library.
* • add\_member(): Registering a new library member.
* • borrow\_book(): Checking out a book, reducing available copies, and linking it to the borrower.
* • return\_book(): Returning a borrowed book and increasing available copies.
* • search\_books(): Searching for books by title, author, or genre.
* • update\_book() / update\_member(): Editing existing records.
* • delete\_book() / delete\_member(): Removing outdated or invalid records safely.

## 3. Data Integrity and Validation

Data integrity is ensured through multiple validation mechanisms, including:

* • Unique ISBNs: Prevents duplicate book entries.
* • Unique Member Emails: Avoids duplicate member registration.
* • Genre Validation: Ensures books are added only with valid genres from the predefined tuple.
* • Borrowing Limits: Prevents borrowing unavailable books or multiple copies.
* • Safe Deletion Rules: Disallows removing books that are currently borrowed or members with outstanding loans.
* • Error Handling: Update and delete operations provide feedback on success or failure.

## 4. Summary

The choice of dictionary, list, and tuple provides a balance of simplicity, efficiency, and reliability. Each function mirrors real-world library actions, ensuring intuitive operations, while built-in validation mechanisms preserve data integrity throughout the system’s use.

