

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**PROG211 – OBJECT-ORIENTED PROGRAMMING 1**

Title : Assignment 1

Issue Date : Week 2

Due Date : Week 4

Lecturer/Examiner : Mr Amandus Caulker

Name of Student/s : Precious Magdalene Siafa

Student ID No. : 905005168

Class : BIT 1101

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: 21/10/25

LECTURER’S COMMMENTS/GRADE:

for office use only upon receive

Remark

DATE :  
 TIME :

RECEIVER’S NAME :

**Mini Library Management System**

**Introduction**

This document explains why dictionaries, lists, and tuples were chosen to build the Mini Library Management System.

**1. Books – Dictionary**

Decision: Use a dictionary with ISBN as the key.  
Why:

* Very fast to find, update, or delete books (O(1) lookup time).
* ISBN is unique, so no duplicates.
* Reflects how books are identified in the real world.
* Allows storing multiple details (title, author, genre, copies) under one key.

Structure:

books = {

"ISBN": {

"title": str,

"author": str,

"genre": str,

"total\_copies": int,

"available\_copies": int

}

}

**2. Members – List of Dictionaries**

Decision: Store each member as a dictionary inside a list.  
Why:

* Easy to loop through and display members.
* Flexible structure that can grow as more members join.
* Works well for a small system where speed isn’t a big issue.

Structure:

members = [

{

"member\_id": str,

"name": str,

"email": str,

"borrowed\_books": [isbn1, isbn2, ...]

}

]

**3. Valid Genres – Tuple**

Decision: Use a tuple for fixed genre options.  
Why:

* Tuples can’t be changed by mistake (immutable).
* Makes it clear these values are constants.
* More memory-efficient and safe than a list.

Structure:

VALID\_GENRES = ("Fiction", "Non-Fiction", "Sci-Fi")

**4. Borrowed Books – List in Member Record**

Decision: Store borrowed ISBNs in a list for each member.  
Why:

* Easy to add and remove items when borrowing or returning.
* Keeps only the book reference (ISBN), not the full book details.
* Allows each member to have a different number of borrowed books.

**5. Function Design**

* All functions return (bool, str) → success status + message.
* Each function does one clear job (e.g., add, update, delete).
* Validation happens before changing data to keep the system clean.

**Conclusion**

Using:

* Dictionaries for books gives speed and structure.
* Lists for members offers flexibility and simplicity.
* Tuples for genres protects data from unwanted changes.

This design is easy to understand, efficient for a small system, and ready to scale in the future.