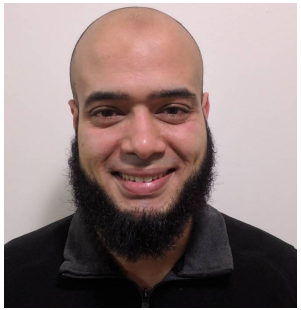
Python Programming

Library System Project

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Background

- Many libraries have a library system
  - Helps to add books and searching for them
  - Maintains information about the borrowed books
- We will create a simple version of this classical system
- The **main user** of the system is an admin
  - Who might add a book, user or perform some relevant operation
  - You don’t need to provide login/logout functionalities in this console system
- The system starts with a menu
  - It shows all possible choices
  - The admin selects a choice.
    - Some operation is performed
  - Then the main menu is listed again

The menu

- Take a minute to read these choices

```
Program Options:
1) Add book
2) Print library books
3) Print books by prefix
4) Add user
5) Borrow book
6) Return book
7) Print users borrowed book
8) Print users
Enter your choice (from 1 to 8):
```

Books operations: Adding a book

- Every system needs data. The core data here is the book and users
- The admin needs to be able to add books
- Each book has the following information
  - id, name and quantity
  - Example: 101, Cpp How To Program, 7
    - We have 7 copies for book Cpp How To Program
    - The book ID is 101

Books operations: Searching for a book

- Searching your database of books is a typical operation
- We will search the system using the book name.
- Instead of the complete book name, we will allow a **prefix**
  - Prefix: The first letters of a word
- Assume we have 3 books in the system, their names:
  - CppHowToProgram, CppForDummies, CppForAdvancedLevels, CoreJava
- Query
  - Cpp ⇒ CppHowToProgram, CppForDummies, CppForAdvancedLevels
  - CppFo ⇒ CppForDummies, CppForAdvancedLevels
  - Core ⇒ CoreJava
  - Java ⇒ Nothing

Book Operations: Listing books

- Another typical operations is to just list all books in the system

Book Operations: Listing users borrowed a book

- Given that several users may borrow a book, the admins may want to know who borrowed what.
  - Remember we have several copies per book.
- Input: Book Name
  - E.g. Math1
- Output: list of the user names who borrowed the book
  - E.g. Mostafa, John, Mark, Ali

User Operations: Add a user

- Each user has only an Id and name
- We only request 2 operations
  - Borrowing a book
  - Returning a book

User Operation: Borrow a book

- Borrowing books is a repetitive scenario in libraries
- Each book already has a specific number of copies (the quantity)
- To borrow a book, this quantity must be > 0
  - Otherwise, this book can’t be borrowed
- After borrowing, the quantity must be decreased
- The admin enters the user name and the book name
  - If there are enough quantity of the book, the system does the following:
    - Mark that this user borrowed a copy
    - Decrease the quantity with 1
  - If there is no available copies, the system notifies the admin

User Operation: Return a book

- Same logic, but this time the system does the reverse:
  - Mark that the user returned a copy
  - Increment the current quantity

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."