



Final Project

Library Management System

It is required to write a program that will create and maintain a Library management system to keep track of books inside the library.

Each Book has ISBN, Title, Author name, quantity, price, publication date (another struct of month, and year).

Your program should allow the user to execute the following commands:

1. **LOAD (READ FROM FILE):**

- This command loads the book's data from a file called "**books.txt**". The file is a text file that is comma delimited in which each entry is found on a separate line.
- Each entry must contain the ISBN, Title, Author name, quantity, price, publication date. An example of one line in the file is as follows:

```
9780136019701,An Introduction to Organic Chemistry,Timberlake Karen,10,3.54,12-2008
```

```
9781506304212,Mathematics for Social Scientists,Kropko Jonathan,7,4.73,12-2015
```

- SEARCH:** The system should process a request by the user to look up information about a specific Book. The user must supply the Book ISBN, and the system should provide all data for the book of that ISBN or a message indicating that the specified ISBN is not found.

Data should be printed in the following format.

```
T    !9781506304212
      ! An Introduction to Organic Chemistry
B      ! Timberlake Karen
  b      ! 10
Q      !3.54$
Q      !December 2008
```

3. **ADD:** The system should prompt the user field by field for the data of a single book and add it to the system.
4. **DELETE:** The user should be prompted for the ISBN and the book associated with that ISBN should be deleted from the system.
5. **PRINT:** Print the data of all books.
Note: Data should be printed the same way as in Query(Search).

(BONUS): Print data in sorted order. The sort is done based on title.
6. **SAVE:** Save the books data by writing them out to the same file in a format similar to that explained in the Load command.
7. **QUIT:** Exit (without saving the data in the external file). You should display a warning to the user that all of his/her changes will be discarded.
8. **MODIFY(BONUS):** The user should be prompted for ISBN. Then the user should be prompted field by field to modify the information for that Book.

Menu

To allow user to choose from various functions you should provide a menu so user can select the desired option. Load should be called by default.

User can select one of the following

- ADD
- DELETE
- SEARCH
- PRINT
- SAVE
- QUIT
- MODIFY(BONUS)

Validation

- Ensure that the ISBN is unique when adding or modifying a book. If a duplicate ISBN is detected, display the message: "Error: Duplicate ISBN."
- Validate that the ISBN is exactly 13 digits long

Error Handling:

Provide clear error messages for common issues, such as:

- "Error: Book not found" when attempting to delete or modify a non-existent book.
- "Error: Invalid command" for unrecognized menu options.

General instructions

Implement each task as a separate function for clarity and modularity.

Use global variables to simplify data management.

You should do a menu to enable easy access for all of these functions.

Late submissions will not be accepted.

You should work in groups of 3 members.

Deadline Saturday 28th December 2024 at 7:00 AM

Cheating Policy:

- All actual programming should be an independent effort. If any kind of cheating is discovered, penalties will apply to the participating students by zero in year work marks, so delivering nothing is so much better than delivering a copy.

Deliverables:

- Report that will contain description for your work, sample runs for the program, user manual (how to use the system), and main algorithms used (search algorithm, sort algorithm).
- Source code.

Good luck ☺