

Programming in C

Final Project



March 24, 2024

Student: Peter Vaughan

NSCC eCampus

**Instructions**

Develop a Book Management System in C that allows users to add, delete, search for, and list books.

**Project Structure**

main.c: Main program entry point, user interface, and menu system.

book.h: Definition of the Book structure and function prototypes.

book.c: Implementation of functions to manipulate book data.

The videos brough back memories about adding headers and other documents. I remembered some do’s and do not’s but I explored more. We should do more practical hands-on experience with this in class. However, I like c programming despite it being confusing at times but it might be too much for those who are not interested.

**1. Define the Book Structure (book.h)**

Define a structure that will represent each book. This should include fields for the book's title, author, ISBN, and publication year.

// book.h

#ifndef BOOK\_H

#define BOOK\_H

typedef struct {

char title[256];

char author[100];

char isbn[13]; // ISBN-13 format

int year;

} Book;

void addBook();

void deleteBook();

void searchBook();

void listBooks();

void saveBooks();

void loadBooks();

#endif

**2. Implement Core Functions (book.c)**

You will implement the core functionalities of your system in this file. This includes operations to add, delete, search for, and list books, as well as save and load data from a file.

Dynamic Array for Storing Books: Consider using a dynamically allocated array to store your books. You will need to keep track of the array size and capacity to manage memory efficiently.

File I/O: Implement saveBooks and loadBooks to write/read the book data to/from a file named books.dat. Use fwrite and fread for this purpose.

Adding a Book: Prompt the user for book details, create a new Book struct, and add it to your array. Ensure you handle memory allocation properly to avoid overflows.

Deleting a Book: Allow users to delete a book based on its ISBN. You will need to find the book in your array and then shift the remaining elements accordingly.

Searching for a Book: Implement search functionality that allows users to find books by title, author, or ISBN. Iterate over your array and print details of matching books.

Listing All Books: Iterate over your array of books and print the details of each book.

I probably approached this in a much more complicated way then needed. A tutor and Chris helped me build a stronger understanding of c. I wanted to explore this beyond the project expectations. Despite C being more difficult than some other languages, it is my favourite.

**3. Main Program Flow (main.c)**

Implement a simple text-based menu system that calls the appropriate functions based on the user's choice. Use a loop to continually display the menu until the user chooses to exit.

#include <stdio.h>

#include "book.h"

int main() {

int choice;

loadBooks();

while (1) {

printf("\n--- Book Management System ---\n");

printf("1. Add Book\n2. Delete Book\n3. Search Books\n4. List All Books\n5. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

getchar(); // Consume newline character

switch (choice) {

case 1: addBook(); break;

case 2: deleteBook(); break;

case 3: searchBook(); break;

case 4: listBooks(); break;

case 5: saveBooks(); return 0;

default: printf("Invalid choice. Please try again.\n");

}

}

}

**4. Compilation and Testing**

Compilation: Include instructions on compiling the project, such as

gcc -o book\_management\_system main.c book.c -Wall.

Testing: Test each functionality thoroughly. Pay special attention to memory management and file I/O operations.

I hope next year we can explore this deeper.

**5. Documentation**

Include comments in your code explaining the logic behind significant sections.

Done

Provide a README file with an overview of your project, how to compile and run it, and a brief description of each feature.

Done

**Evaluation Criteria**

Your project will be evaluated based on functionality, code quality, and adherence to the project requirements. Ensure that your code is well-organized, commented, and follows good programming practices.