

---

**: Linked Lists - Books Database**

---

This assignment was designed to exercise the use of functions and dynamically allocated **structs** with pointer member variables in a **doubly linked list** with a header (**rc**, **\*head**, and **\*tail**) scheme as discussed in class (see slides' layout).

Write a program that implements a database of books. The database should be kept in a text file, **booksdb.txt**, which is used for both input and output. The program should detect the existence of the file in the current directory, and if it exists, populate the linked list with the contents of the file before anything else. Create a menu for user interface with options for adding/deleting a record to/from the database, displaying one or all records on the screen, and saving the list to the text file **booksdb.txt**.

Example menu screen:

Select from the following options:

1. Add a book
  2. Delete a book
  3. Display a book
  4. Display all books
  5. Save and exit
- Enter your choice:

Use the following format for the input/output file:

**isbn**  
**author**  
**title**

For example:

**978-1-337-11756-2**  
**Davender S. Malik**  
**C++Programming: Program Design Including Data Structures, 8th ed.**

Where the text file:

- has three fields per record (one per line)
- has an empty line between records
- resides in the same folder with the other relevant project files
- as mentioned above, it is called **booksdb.txt**

Furthermore,

- use functions for handling each menu task
- put the functions in separate files
  - `addBook()` should go in `addbook.cpp`
  - `delBook()` in `delbook.cpp`
  - `showBook()` in `showbook.cpp`
  - `showBooks()` in `showbooks.cpp`
  - `popList()` in `poplist.cpp`
  - `menu()` in `menu.cpp`
- use `valgrind` to make sure there are no memory leaks
- include the `valgrind` run in your `hw06.scr` file

Organize it as usual; include a header file (`hw06.h`) and `Makefile` with your project.

Use the CLI `script` command to capture an interaction with the program:

1. Start the program normally (i.e., `./hw06`)
2. Add at least five books
3. Display all books
4. Delete the last book
5. Close the program
6. Start again, now within `valgrind` (i.e., `valgrind ./hw06`)
7. Add another book
8. Delete the second book
9. Display all books
10. Save and Exit

Create a `tar` package file `hw06.tar` that includes all relevant files; `hw06.h`, `hw08.cpp`, `addbook.cpp`, `delbook.cpp`, `showbook.cpp`, `showbooks.cpp`, `popList.cpp`, `menu.cpp`, `Makefile`, `booksdb.txt`, and `hw06.scr`.

Submit the `tar` package file `hw06.tar` to canvas by the due date on top of this page.