

Programming Assignment #7

Manage a Reading List

CS 1428.004, Fall 2011

Instructor: Jill Seaman

Due: in class Friday, 11/18/2011 (upload electronic copy by 9:30am)

Problem:

You have a friend who is a book worm and wants a program to manage the list of books he wants to read. He wants to be able to add books to the list, remove books from the list, and display the list.

A friend of his started writing the program for him, but now she is too busy to finish it. Her program uses an array of string to represent the books in the list, and she has a menu system implemented to let the user add and remove books, and display the list. She even started writing the function that performs the operation to remove a book, but it is incomplete.

You will complete the program. The unfinished program calls 4 functions that each operate over the array of books. You can see the function calls in the program, but there are no prototypes or function definitions for these functions. You will need to provide them.

Do not change anything in the original program--you must define your functions to match the function calls.

Here is a description of each of the functions that you need to write:

addBook: accepts the list of books, the size (the number of array elements in use), and a string containing the title of the book to add. If the array is not yet full, add the title to the next open slot in the array (and increment the size). Return true if you were able to add the title, and false if you could not because the array was full.

showList: accepts the list of books and the size. Displays the titles currently in the array underneath a column header.

The next two functions are called from the removeBook function:

getBookPosition: accepts the list of books, the size, and a string containing the title of the book to remove. Searches the list for the title. If found, returns the position of the book in the list. If not found, it returns -1. (If the title is in the list more than once,

this function should return any one of the positions of the book in the list.)

removeBookAtPosition: accepts the list of books, the size, and the position in the list of the book to remove. Removes the element by shifting all the elements lower in the list up one position (and decrements the size). You can set the last element in the list to "" to erase it, but it's not necessary.

NOTES:

This program is an example of using a partially filled array. During the program, the number of strings (titles) in the array will vary. This is why we need to maintain a size variable that stores the number of entries in the array currently in use. If a given function increments or decrements the size, be sure to make it a reference parameter for that function.

You will be using the string data type for the elements of the list. You can use assignment (=), comparison (==), and output (<<) operators on strings, as well as pass them as arguments to functions.

The incomplete program and a the output from a test run of the finished program are available on the class website.

Style:

See the Style Guidelines document on the course website. **Especially pay attention to the comments required for functions.** The grader will deduct points if your program violates the style guidelines.

Logistics:

Your program must be free of compiler errors.

Do you not use any features of C++ that we have not yet covered in class (use features from Chapters 1-7 (and **strings**) only).

Name your file **assign7_xxxxxxxxx.cpp** where xxxxxxxxxx is your 9 character TX state ID number, the one that is on your ID card. It should look something like this: A04123456. If yours is just six digits, then add "A00" to the front.

There are **two** steps to the turn-in process:

1. Submit an **electronic copy** using the following upload link:

<http://www.cs.txstate.edu/~js236/homework>

(There is a link directly to this page on the course website).

Click on CS1428.004, and log in with your Net ID and follow the directions to upload your file.

2. Submit a **printout** of the file on the day it is due. Please print your name on the front page, and staple if there is more than one page.