

**Computer Science Department**

**COMP133 ( Spring 2015)**

**Project Phase Three *Due Date: Mon (11/5/2015)(Sections 2+3 in lecture and Section 1 beginning of lab)***

In this phase, you need to add the book names (strings) to the functions you created in phase two as follows:

***void displayMainMenu();*** // displays the main menu shown above

This function will remain similar to that in phase two.

***void addBook( char names[][NAMESIZE], int bins[], double prices[], int \*size);***

This function will receive the arrays containing the book names, bin numbers and the prices as parameters. It will also receive a pointer to an integer which references the current size of the list (number of books in the list).

The function will check to see if the list is not full. If list is not full ( size < MAXSIZE) then it will search for the appropriate position of a given bin number and if the bin number is already in the list it will display an error message. If not, the function will shift all the bins starting from the position of the new bin to the right of the array and then insert the new bin into that position. Same will be done to add the name and price of the book to the names and prices arrays respectively.

***void removeBook(char names[][NAMESIZE], int bins[], double prices[], int \*size);***

This function will receive the arrays containing the book names, bin numbers and the prices as parameters. It will also receive a pointer to an integer which references the current size of the list (number of books in the list).

The function will check if the list is not empty. If it is not empty (size > 0) then it will search for the bin number to be removed and if not found will display an error message. If the bin number exists, the function will remove it and shift all the elements that follow it to the left of the array. Same will be done to remove the name and the price of the book from the names and prices arrays respectively.

***void searchForBook(char names[][NAMESIZE], int bins[], double prices[], int size);***

This function will receive the arrays containing the book names, bin numbers and the prices as parameters. It will also receive an integer which has the value of the current size of the list (number of books in the list).

The function will check if the list is not empty. If it is not empty (size > 0) then it will ask the user to enter a bin number and will search for that bin number. If the bin number is not found it will display an error message.

If the bin number is found then it will be displayed along with the book name and price in a suitable format on the screen.

***void uploadDataFile (char names[][NAMESIZE], int bins[], int prices[], int \*size );***

This function will receive the arrays containing the book names, bin numbers and the prices as parameters. It will also receive a pointer to an integer which references the current size of the list (number of books in the list).

The function will open a file called ***books.txt*** for reading and will read all the book names, bin numbers and prices and store them in the arrays.

***void updateDataFile(char names[][NAMESIZE], int bins[], double prices[], int size);***

This function will receive the arrays containing the book names, bin numbers and the prices as parameters. It will also receive an integer which has the value of the current size of the list (number of books in the list).

The function will open the file called ***books.txt*** for writing and will write all the book names, bin numbers and prices in the arrays to that file.

***void printBooks (char names[][NAMESIZE], int bins[], double prices[], int size);***

This function will receive the arrays containing the book names, bin numbers and the prices as parameters. It will also receive an integer which has the value of the current size of the list (number of books in the list).

This function will print the information (names, bins and prices) currently stored in the arrays in a suitable format.

***Note: You need to define a constant called MAXSIZE ( max number of books stored) equal to 100 as well as a constant NAMESIZE (max size of a book name) equal to 50.***

***A book name may contain only letters, numbers, and underscore and must start with a letter (e.g. Introdcution\_To\_Computers, Computer\_Programming\_Fun, Geography\_1 and so forth). You may assume that book names will NOT have any space characters in them.***

**Items that should be turned in by each group:**

1. ***A hard copy of program code.***
2. ***hard copy of a complete run similar to the output shown below.***

***SAMPLE RUN:***

Make sure your program works **very similar** to the following sample run:

***Assuming that at the beginning of the run file books.txt has the following information stored:***

***Introduction\_To\_Geography*** 1234 72.45

***Computer\_Fun*** 2345 25.20

******

******

******

***Late project phases (after lecture or lab on the due date) will not be accepted for any reason***.