Project [10]: File Input/output

Due Tuesday, 12/11/2018, 11:59 pm

Project Goals

The goal of this project is to:

1. Get students familiar with functions to read from and write to files, in text and binary modes

Important Notes:

1. **Formatting:** Make sure that you follow the precise recommendations for the output content and formatting: for example, do not change the text in the first problem from "`Please enter item data (part number, quantity, price): `" to "`Enter the item: `". Your assignment will be auto-graded and any changes in formatting will result in a loss in the grade.
2. **Comments:** Header comments are required on all files and recommended for the rest of the program. Points will be deducted if no header comments are included.
3. **Restriction:** The use of `goto` statements anywhere within this program is prohibited. Points will be deducted if `goto` is used.

Problem 1

Write a program to repeatedly ask the user to enter information regarding inventory for a business (item part number, quantity, price) and then saves the information to a file called `inventory.txt`. The program stops the loop when the user enters 0 for the part number.

The program should function as follows (items underlined are to be entered by the user):

```
This program stores a business inventory.
Please enter item data (part number, quantity, price): 3, 1, 2.4
Please enter item data (part number, quantity, price): 1, 4, 3.0
Please enter item data (part number, quantity, price): 0
Thank you. Inventory stored in file inventory.txt.
```

Note:

- Your program must write to the file in binary mode, using the `fwrite` function

Save your program as `save_inventory.c`

Challenge for problem 1 (10 extra credit points):

Modify your program to check if the user enters an identical part number again. If this occurs, the program should print a message and ask for the input again.

The program should function as follows (items underlined are to be entered by the user):

```
This program stores a business inventory.
Please enter item data (part number, quantity, price): 3, 1, 2.4
Please enter item data (part number, quantity, price): 1, 4, 3.0
Please enter item data (part number, quantity, price): 3, 4, 3.0
This item has been entered before.
Please enter item data (part number, quantity, price): 2, 4, 1.3
Please enter item data (part number, quantity, price): 0
Thank you. Inventory stored in file inventory.txt.
```

Save your challenge separately as `save_inventory_c.c`

Problem 2

Write a program to read information from the `inventory.txt` file and display it to the screen, formatted as follows: Part#, Quantity, and Item Price in the table header should be separated by tabs. The part number field should take 5 spaces (values right justified), the quantity field should take 8 spaces (values right justified), and the price field should take 9 spaces with 2 numbers after the decimal (values right justified, with the $ sign in front of the price).

The program should function as follows:

```
Below are the items in your inventory.
Part#   Quantity        Item Price
    3          1        $     2.40
    1          4        $     3.00
    2          4        $     1.30
```

Note:

- Your program must read from the file in binary mode, using the `fread` function

Save your program as `disp_inventory.c`

Grading Rubric

Grading will be done for each problem as follows:

Correctly-named file	5%
Header comment	2%
Program compiles	5%
Correctly-reading data from terminal	18%
Correct result printed	20%

Submission details

To submit your project, you will have to use the submission script. You do this by either:

1. Working on an ECC machine
2. Working on the provided VMware
3. Secure Copying your files (See Mac Support for information)

To Submit your project:

- Have a directory called "project10"
- Save your *.c files in that directory
- To submit: **(don't type the '>' symbols)**

  > cd project10
  > submit

The submission script copies all files in the current directory to our directory. You may submit as many times as you like before the deadline, we only keep the last submission.

Academic Honesty

Academic dishonesty is against university as well as the system community standards. Academic dishonesty includes, but is not limited to, the following:

Plagiarism: defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion.

Cheating: defined as (1) obtaining or providing unauthorized information during an examination through verbal, visual or unauthorized use of books, notes, text and other materials; (2) obtaining or providing information concerning all or part of an examination prior to that examination; (3) taking an examination for another student, or arranging for another person to take an exam in one's place; (4) altering or changing test answers after submittal for grading, grades after grades have been awarded, or other academic records once these are official.

Cheating, plagiarism or otherwise obtaining grades under false pretenses constitute academic

dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include cancelling a student's enrolment without a grade, giving an F for the course, or for the assignment. For more details, see the University of Nevada, Reno General Catalog.