Data Structures (2028C) -- Spring 2019 – Homework 1

Topics covered: Working with Files and Structures

Homework due: ***Thursday, Jan 31 at 6:00PM***

Objective:

The objective of this homework is to create structures, read from, create and write to files. This assignment will incorporate arrays

Scenario:

You are a programmer for Google Books who has been tasked to take the plain text from a series of formatted files and create a card catalog file and a statistics file. This needs to be written using C++.

Requirements:

1. Ask the user for the name of the file to be processed.
2. Attempt to open the file of the given name. If unsuccessful, it should output an error message and prompt the user to enter another file name.
3. With the file open, read into a Structure:
    a. Title
    b. Author full name (the name is stored in the file as First and Last name with a space between)
    c. Word Count (total number of words in the book contents. A word is one or more letters (not counting punctuation) separated by a space so ice cream is considered 2 words)
    d. Letter frequency (this is the number of times each letter has been encountered in the book contents)
    e. Line Count (count of new line characters in the contents section)
4. Save this information in the file CardCatalog.txt
    a. If the file doesn't exist, create it.
    b. If the file does exist, append to it.
    c. Leave a blank line between each card catalog entry.
    d. The file should be human readable such as (is should be similar but not necessarily the same as the following[i]):

        Title: Moby Dick
        Full Author: Herman Melville
        Author First Name: Herman
        Author Last Name: Melville
        Word count: 375
        Line count: 17
5. Ask the user if they want to see the letter frequency. If they agree, it should look like[ii]:

    Moby Dick letter frequency:
    a: 0.0762%
    b: 0.0 %
    c: 0.0253%
    d: 0.0405%

6. Ask the user if they wish to process another book. If they do, repeat requirement 2. If they don't, program should quit. This should not add to previous results.

Submission:
Submit all source code files and any required data files in a zip file. Include a write up as a PDF including:

- The name of all group members (minimum 2 members, maximum 4 members).
- Instructions for compiling and running the program including any files or folders that must exist.
- What each group member contributed. If the contributions are not equitable, what portion of the grade each group member should receive.

Submission should be submitted via BlackBoard.

Grading:
1. 5% - Requirement 1 works correctly
2. 10% - Requirement 2 works correctly
3. 25% - Requirement 3 works correctly
4. 20% - Requirement 4 works correctly
5. 20% - Requirement 5 works correctly
6. 10% - Requirement 6 works correctly
7. 10% - Code is well formatted, well commented and follows a

If program fails to compile, the grade will be limited to a max grade of 50%.

---

[i] The numbers and values are in this example should not be considered correct
[ii] This should include all 26 letters and should not include any other characters. This should not differentiate between upper and lower case letters.