Programming Assignment #7

Due Nov 19 by 3pm **Points** 100 **Submitting** a file upload
Available Nov 12 at 8pm - Dec 5 at 12pm 23 days

This assignment was locked Dec 5 at 12pm.

Reading, Analyzing and Summarizing Real-World Log Files

Here's a link to a real-world Apache HTTPD log file: **access-log.txt**

1.) Start by creating a new Visual Studio solution for this assignment. Just like before, it should be a Console app. After you have created the solution directory, download that real-world log file and move it into the same directory that Visual Studio created the new "program.cs" file in.

2.) Now, create a program that reads the data from that file and displays it with pre-pended line numbers. (This is an example if incremental improvement - first just make sure you can read the file.)

NOTE: Use "..//..//..//access-log.txt" as the filename for the log file inside of your code.

3.) Once that is working, change the program so that it parses each line and extracts the IP address, HTTP status code (usually 200), and the requested URL path (ex. "/styles/com.stockcharts.workbench.sc/scui.js") from each line of the log file. Note: URL paths do NOT include question marks or any of the stuff after the question mark. For now, have your program print out just those three fields to the console.

4.) Once that is working, change the program so that it stores summary information about those fields in three different dictionaries - one for the IP addresses, one for the status codes, and one for the URL paths. Keep a running total for each unique entry in each of those dictionaries. At the end of the program, print out the contents of each dictionary in a seperate section. For the "IP Address" and "URL Paths" dictionary, only print out paths that have 10 or more requests.

5.) Once that is working, change the end of the program so that you display each of the dictionary's contents in sorted order. For the status counters, sort by the status itself (lowest to highest). For the IP Address counters, sort by frequency of IP address (highest to lowest). For the URL Paths, sort by frequency of URL path (highest to lowest).

EXTRA CREDIT: 5 points

6.) Write out that summary information in a CSV file called "access-summary.csv". Make sure your file can be read in by a Spreadsheet program like Excel or OpenOffice Calc.

SAMPLE OUTPUT FOR PROGRAMMING ASSIGNMENT 7

IMPORTANT NOTES:

- Just like before, code readability counts! Minimize the need for comments by maximizing the readability of your code and the identifiers in your code.

- Don't use stuff we haven't covered in class - SPECIFICALLY LINQ! You cannot use any part of LINQ for this assignment.
- Everything you need to create this solution has been discussed in the "1200" modules.
- Upload your entire Solution directory including the access-log.txt file and the files that your program generated when you ran it. (Those files should already be in your Visual Studio directory.)
- To be clear - only turn in code that you yourself wrote. It is a violation of the Student Code of Conduct to turn in work written by someone else and claim it as your own. Please do not share your code with other students directly. If someone asks you for help, help them understand general concepts (maybe by reviewing these modules together) but don't just give them your code.

My Grading Guideline:

30% Does the program compile and run?
40% Does the program generate correct results?
10% Does the program use OO design principles correctly?
10% Does the program handle potential error conditions properly?
10% Does the program follow the C# style guide rules