Assignment 03 – Using the Google APIs to access YouTube Data

For this assignment, you will access data from YouTube, which is owned by Google. You will use the credentials from the developer account you created in Assignment API3.

**ASSIGNMENT:**

- Write a program to retrieve some data from the YouTube API, and do some processing on that data, as outlined in the program specifications.

  ***Your program must follow the specifications and the programming guidelines provided below.***

- Write a one-page report describing your program. The report must include the following ***labeled*** sections:
  - PURPOSE;
  - INPUT
  - OUTPUT
  - WHAT THE PROGRAM DOES
  - ADDITIONAL INFORMATION (Anything you want to share -- students often talk about special issues that came up and how they addressed them, for instance. You may not have anything you want to include, and that is fine.)

  The report should be submitted in .pdf or Word format, in a document ***labeled with your last name***, eg: Dugas_HW3_Report.pdf.

- Zip your report, code, .csv file, and a **screen shot of any output** into a zip (compressed) file that is labeled (both inner and zip folders) with your last name, eg: Dugas_HW3.zip (I.E. it should unzip to a folder that also has your last name) Submit in Canvas. *Screen shots may be included in your report, but also must be included separately in the zip file.*

**PROGRAM SPECIFICATIONS:**

NOTE: All programming for this assignment -- data search/retrieval and processing/analysis -- must be accomplished through a single program. Do not write two separate programs.

All programming must be in python 3 unless otherwise arranged with the instructor.

If you use external python packages, please note that in your program comments in the run instructions.

Your program should be named: **lastname.py**

You may find these sites helpful:
https://github.com/googleapis/google-api-python-client/blob/master/docs/start.md

https://developers.google.com/youtube/v3/docs/videos

There are two types of output for your program: a .csv file and output printed to the terminal.

1. The .csv file should contain the specified raw data fields that were retrieved, one row for each record retrieved. The fields should be these data fields: Video Id, Views, Likes*, Dislikes*, Duration, Title. The first row of the .csv file should be a header file that contains field names.
   * if likeCount or dislikeCount are missing, substitute 0.

2. The analysis results, in user-friendly report form as specified below, should be printed to the console.

The program must run in a terminal window. Prompt the user for a search term and a maximum number of results***. Do not use an argument list.***

Use the search term and max to do a YouTube search. Write the results, as specified above, to a .csv file that will remain in existence when the program is completed.

Print the **search term**, **search max**, and results of your analysis to the console.

Perform the analysis described below on the results of your search, and print the results to the console. Be sure to include a header to describe each set of results.

Analysis –

1. List, ***in table form***, the Video Id, Views, Likes*, Dislikes*, Duration, Title. The first row of the table should be a header file that contains field names. Number each video line of the table.
   * if likeCount or dislikeCount are missing, substitute 0.

2. List the rank (1 to 5), the like percentage, view count, like count*, and title for the top 5 videos with the highest like percentage ( like count / view count ), sorted by highest percentage first. The info should be printed in table form, with a header line. If there are not 5 videos with a like percentage, leave extra lines blank.
   * if likeCount is missing, substitute 0
   Note: You'll need to check for divide by 0 error, as there may be cases where there are videos with zero views but non-zero likes/dislikes (such as not-yet-published streaming videos). They should not be included in this list.

**PROGRAM GUIDELINES:**

You may incorporate materials provided by the instructor, although you must add your own original content as well. Your grade will be based on your original content, so be sure and add significantly to what is provided.

Programs will be screened for plagiarism. If you “borrow” code, be sure to document the details of the source; otherwise it will be considered plagiarism and result in a zero grade for the assignment. Borrowed code will not count toward your grade, only original code will be considered.

Programs should employ good programming practices. An example is the use of descriptive variable and function names.

Annotation and Comments: ***IMPORTANT***

- Program header must include **your** name and assignment information (use comments).

- Comments must also be used at the beginning of the program to give an overall description of the purpose of the program.
- Comments must also include detailed running instructions to run in a terminal window.
- Comments should also be used throughout the code to explain what it is doing.  It should be possible to re-create your program based on the comments alone.  Poorly commented programs will receive poor grades.

***Note for students traveling outside of the US:***

Social media sites are sometimes blocked outside of the US.  To get around this, you can use a VPN, either a commercial one or Stevens's VPN.  For Stevens:  IT Service Desk / Search Knowledge Base / Network / VPN