Assignment: CSV File Manipulation in Python

After completing this assignment, a student can:

- Import a .CSV file, replace bad data with good data and produce a new .CSV file
- Print text to illustrate the result
- Run code to verify that it **works as intended** (WAI)

Lab Assignment Prompt: Analyze Customer Satisfaction Ratings

Every time you visit your favorite restaurant, The Happy, Eccentric Eatery (THEE), the dining room does not seem to be busy. Perhaps THEE is losing business, and you'd like to understand why.

Like many businesses, the restaurant collects data it receives from online reputation management sites. THEE is having trouble with one file because some of the data in one of column is bad.

You offer to help clean the data for them, in exchange for a free drink the next time you visit. The business agrees, and they send you their ratings data file: **unit_7_restaurant_rating.csv**

Your task is to import the .csv file into your Python environment, replace the bad data with good data, and output a new .csv file with the corrected information.

NOTE: Run **ALL** of the code you created and make sure it **works as intended** (WAI).

Write a function that meets the following criteria:

1. Inputs the data from the file **unit_7_restaurant_rating.csv**
    a. Store the input file into a dictionary to make manipulation easier once loaded into Python.
2. Scans all the data in the column named "Worth_the_price" (this is the column with bad data).
    a. This column is a 'yes' or 'no' column.
    b. Some records contain invalid data. These records list the value '45' instead of 'no'.
    c. Replace all the '45' values with 'no'.
3. Exports the corrected dataset into a new .csv file.
    a. Include all the original data **EXCEPT** the bad records.
    b. Include the corrected records from requirement B.

Assignment continues on the following page.

4. Includes parameters to handle the file names, bad data, and replacement value.

   Your function should be able to take **ANY** input file, name of a bad column, bad data to look for, and replacement value, and export a new .csv file.

   However, you may wish to default these parameters to meaningful values for this assignment.

   At a minimum, the parameters **MUST** accept the following data:

Input file name	Name of the input file
Output file name	Name of the output file
Bad data column name	Name of the column that has the bad data In this case: "Worth_the_price"
Bad data value	The specific data that your code is looking to replace In this case: The default, '45'
Replacement value	The specific data with which your code will replace the bad_data_value In this case: The default, 'no'

NOTE: This task must be performed using the **csv** Python library.

REMINDER: Run all of the code you created and make sure it **works as intended** (WAI) and **Review Assignment Rubric** **(on pages 3-5).**

Assignment Upload (Submission) to Blackboard:

1. From the command line at the root directory of your project, execute the following command:

   ```
   git bundle create firstname-lastname-CSV-date.bundle --all
   ```

   Note: firstname and lastname are your first and last names, and date is in the format of mmddyyyy.

2. Access your Blackboard course (https://courses.cscc.edu/)
3. Open the course and navigate to the assignment.
4. Click **UPLOAD** in the title of the assignment
5. Click Browse Local Files and attach your file
6. Upload (**SUBMIT**) the file to the corresponding assignment link in Blackboard.

Lab 7 Scoring Guide				
How am I doing?	**Needs Improvement 0-19 pts**	**Emerging 20-31 pts**	**Proficient (80%+) 32-40 pts**	**Score /40**
Code Functionality **Key Requirements:** • The function includes 5 parameters. Default values for any of them is optional. • Opens the required .csv file. • Stores the result into a dictionary. • Finds the records with "45" values in the "Worth_the_price" column. • Replaces the records with "45" values in the "Worth_the_price" column with "no". • COUNTS AS TWO REQUIREMENTS: Saves the updated dataset into a NEW .csv file. • Utilizes arguments fed into the function for all of the above.	• Fewer than Five requirements delivered as expected. • 30 Points deducted if lab produces a syntax error that prevents the lab from running.	• Five, or Six requirements delivered as expected. • 15 points deducted if lab produces a runtime error but still runs under certain circumstances.	• Seven or Eight requirements delivered as expected.	

Lab 7 Scoring Guide

How am I doing?	Needs Improvement 0-19 pts	Emerging 20-31 pts	Proficient (80%+) 32-40 pts	Score /40
Code Structure	• Script is not simple and is overly complex, AND • Several or many unnecessary redundant operations, AND/OR • Functions do not use the return keyword, AND/OR • The functions does not use arguments as inputs.	• Script is not simple or is overly complex, OR • Some unnecessary redundant operations, OR • Functions do not use the return keyword, AND/OR • The function uses between one and three arguments as input.	• Script is simple and not overly complex • Few or no unnecessary redundant operations • All functions use the **return** keyword. • The function uses four or five arguments as input.	

How am I doing?	Needs Improvement 0-9 pts	Emerging 10-15 pts	Proficient (80%+) 16-20 pts	Score /20
Code Quality & Readability	• Code is difficult for humans to read or understand AND • PEP-8 validation produces multiple hard warnings or errors AND deviations from PEP-8 do not improve code clarity AND/OR • Some unnecessary code artifacts are left in the assignment, AND	• Some code is difficult for humans to read and understand, OR • PEP-8 validation produces one or two hard warnings or errors AND/OR deviations from PEP-8 do not improve code clarity. AND/OR • Variable naming styles are	• All code is easy for humans to read and understand. • PEP-8 validation produces no hard warnings or errors and Deviations from PEP-8 improve code clarity. • No unnecessary code artifacts are left in the assignment. • All comments are appropriate and	

• Some comments are excessive or unnecessary, AND/OR • Variable naming styles are inconsistent, AND/OR • Docstrings are missing.	inconsistent, AND/OR • Docstrings are missing.	provided valuable clarity. • Variable naming styles are consistent. • Docstrings are present.