Name: Zach Caldwell
Class: CS 112-01
Date: April 11, 2017

| Cues: | Notes: |
|---|---|
| | 1. Working with CSV files using the `csv` module: |
| csv.reader() | a. `csv.reader` objects can be obtained using `csv.reader()`, passing in an open handle to a CSV file. Example:<br>```>>> import csv```<br>```>>> exampleFile = open('example.csv')```<br>```>>> csvReader = csv.reader(exampleFile)```<br>```>>> type(csvReader)```<br>```<class '_csv.reader'>``` |
| list(csv.reader) | b. `csv.reader` objects can be converted directly to lists of value lists, one sub-list per line, using `list()`, which makes the data easier to work with. Example (re-using the variables above):<br>```>>> csvData = list(csvReader)```<br>```>>> print(csvData)```<br>```[['abc', '123'], ['def', '456'], ['ghi', '789']]``` |
| line_num<br><br>Direct iteration | c. `csv.reader` objects can be also be directly iterated over, and in that case they store the current line number in the `line_num` attribute. Example:<br>```>>> for row in csvReader:```<br>```>>>     print('Row', str(csvReader.line_num),```<br>```\        str(row))```<br>```Row 1 ['abc', '123']```<br>```Row 2 ['def', '456']```<br>```Row 3 ['ghi', '789']``` |
| Single-use gotcha | d. `csv.reader` objects can be listed / iterated over only once, and must be re-created in order to reread the CSV file. |
| csv.writer<br><br>writer.writerow() | e. `csv.writer` objects enable writing CSV data, and must be called with an open writable file handle. One can then write data a row at a time by repeatedly calling the `writer.writerow()` method, passing in a list of row values each time. Example:<br>```>>> outFile = open('out.csv', 'w')```<br>```>>> outWriter = csv.writer(outFile)```<br>```>>> outWriter.writerow(['abc', 123, 'spam'])```<br>```>>> secondRowData = [456, 'def']```<br>```>>> outWriter.writerow(secondRowData)```<br>```>>> outFile.close()``` |
| Delimiter keyword | f. `csv.writer()`'s `delimiter` keyword argument allows one to specify a field delimiter besides the default comma, e.g. one could use the tab ('\t') character instead (technically making it a TSV file). |

| Cues: | Notes: |
|---|---|
| lineterminator keyword | g. `csv.writer()`'s `lineterminator` keyword argument allows one to specify a different line ending that the default, for example '\n\n' which would double-space each line. |

**Summary/Reflection:**