Name: Zach Caldwell
Class: CS 112-01
Date: March 14, 2017

Cues:	**Notes:**
	1. Reading and writing files: a. File paths:
File path parts	i. Paths all have two parts: a *directory path* and a *filename*. Paths only containing a partial directory specification (e.g. '.\Project.docx'') are called *relative* paths, while full paths (e.g. 'C:\Users\Guest\Documents\Project.docx') are called *absolute* or *fully-qualified* paths.
Absolute vs relative paths	
'.' and '..' paths	ii. Standard relative-path implementations can use two special folders, known as the *dot* ('.') and *dot-dot* ('..') folders. These refer to the current folder and its parent folder, respectively.
Platform path differences	iii. Paths use drive letters and escaped backslashes on Windows (e.g. 'C:\\Users\\Zach'), while using root folders and forward slashes on UN*X / POSIX-style systems (e.g. '/home/'). 1. This can be done automatically using the **`os.path.join()`** function. Example: **`>>> import os`** **`>>>os.path.join('C:','Users','Zach')`** **`'C:\\Users\\Zach'`**
os.path.join()	
Current working directory	iv. The current working directory can be set or retrieved using **`os.chdir()`** or **`os.getcwd()`**, respectively. Example: **`>>> import os`** **`>>> os.getcwd()`** **`'/home/Zach'`** **`>>> os.chdir('/usr/local/bin')`** **`>>> os.getcwd()`** **`'/usr/local/bin'`**
Creating new folders	v. New folders can be created recursively with the **`os.makedirs()`** function. Example: >>> import os os.makedirs('C:\\Projects\\NewProject\\SecretStuff\\Code')

Name: Zach Caldwell
Class: CS 112-01
Date: March 14, 2017

Cues:	**Notes:**
Checking existence	vi. Paths and files can be checked for existence and differentiated using **`os.path.exists()`**, **`os.path.isfile()`**, and **`os.path.isdir()`**.
Using files	b. Working with files: i. Files can be opened, closed, read, and written to using **`open()`** (which expects a filename and open mode and returns a File object), **`.read()`** (which reads the file in to a string), **`.write()`** (which overwrites or appends a given string to the file, depending on the open mode), and **`.close()`**, respectively. Example: **`>>> spamFile = open('spam.txt', 'w')`** **`>>> spamFile.write('Hello World!\n')`** **`>>> spamFile.close()`** **`>>> spamFile = open('spam.txt', 'r')`** **`>>> fileContents = spamFile.read()`**

Summary/Reflection: