

CS 1213

Programming with Python Assignment 6: Working with File(s)

Create a Python program in a single file named `your4x4_6.py` (where *your4x4* is replaced with your OU 4x4 username, `_6` is the literal underscore and number six character (representing this is Assignment 6), and `.py` is the literal suffix to the file) which meets the following requirements:

Requirements:

Prompt the user to enter a file name / path to read.	2 points
Read each line of the user-specified file and interpret each as follows:	10 points
<ul style="list-style-type: none">• Ignore blank lines• If the line begins with the text "name" (ignore quotes), then a space, add the remainder of the line's text to a collection of names.• If the line begins with the text "item" (again, ignore quotes), then a space, add the remainder of the line's text to a collection of items.	
After processing the full contents of the file (every line) print each name from the names collection along with a randomly selected item from the items collection. No two names should be printed with the same item.	10 points
At the top of your file must be a comment the text of which is your name	1 point
For each function, loop, and any other potentially complicated section of code write a comment explaining what the code is doing	2 points

You must complete the assignment by **Monday, November 25 at 11:59 PM** and submit the `.py` file containing Python code on Canvas to Assignment 6. Do not submit any other files, or a ZIP of the `.py` file, or anything else. Should Canvas be down or unavailable, you may e-mail the file to nicgrounds@ou.edu

Rubric for grading:

	Acceptable	Needs Improvement	Fail
Functionality	Meets all requirements listed above. May have additional commands or functions to better organize the code.	One or two basic requirements are unmet (e.g., some of the commands given by user input).	Program fails to execute or is missing even basic functionality (e.g., prompting the user for input, printing simple messages).
Non-Functional	Clear comments explaining what each function is doing. Clear variable names that intuitively relate what the variable's purpose is.	Unclear comment(s) or variable names which are arbitrary and uninformative.	Missing comment or comment that merely re-stated the requirement. Variable names are misleading for what their purpose actually is.

Sample contents of a file to be processed by the program, e.g., gifts.txt:

```
name Nic
name Steve
name John
name Rainn
name Jenna

item Ipod
item Teapot
item Oven Mitt
item Poster
item Paintballs
item Shot Glasses
```

Sample potential execution of a program meeting the requirements:

```
What file would you like to process?
gifts.txt
```

```
Nic - Teapot
Steve - Shot Glasses
John - Poster
Rainn - Oven Mitt
Jenna - Paintballs
```

```
Thanks for using my program.
```