UNIVERSITY of FLORIDA
Herbert Wertheim
College of Engineering

COP3502

Lab 07: The Cow Says…

Overview

This lab is designed to introduce students to the Bash Command Line Interface (CLI) and the concept of CLI arguments and give them practice writing classes. For (only) this lab, you are not allowed to use any windowed editor (such as PyCharm or Notepad++). Instead, you will use a Unix-based command line editor.

The **`cowsay`** utility is a popular Unix program from the 20th century (see https://en.wikipedia.org/wiki/Cowsay). You will write a slightly simplified **`cowsay`** program that takes in several arguments and prints out different text depending on the arguments. You must log in to the school computers using your CISE account to do this lab.

Tools

Please note that **you must use a text editor and the terminal to edit and run your program and its directories.** It is advised students learn/review basic Unix shell commands before beginning; a good run-through can be found here: https://linuxjourney.com/lesson/the-shell.

Follow these steps to get started on the lab:

1) Open a terminal and enter the **`pwd`** command to identify the path to the working (current) directory (folder)
2) Enter **`ls`** to list the contents of the current directory
3) Use the **`mkdir`** command to make a new directory called Cow*Lab*.
4) Use **`ls`** to see the change, then **`cd`** to change to the directory Cow*Lab*.
5) Do your lab work in that folder. Use your google skills to find more commands.

You can read more information about some of these commands here:
https://www.howtogeek.com/howto/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/
https://pythonbasics.org/execute-python-scripts/

Specification

Students will write two files: a driver file with a **`main()`** entry point (`cowsay`) and a data class (`cow`). Note that `heifer_generator.py` is provided for you; your code must use this class to create the cow objects.

Provided For Students - HeiferGenerator

`get_cows``()`
Static method which returns a Python list of cow objects from the built-in data set. This will use the `Cow` constructor and `image` property of the cow class to properly initialize new cow objects uniquely for each data set. *This means it is dependent on your Cow class, so you should write that before working on main!*

cowsay.py (Program Driver)

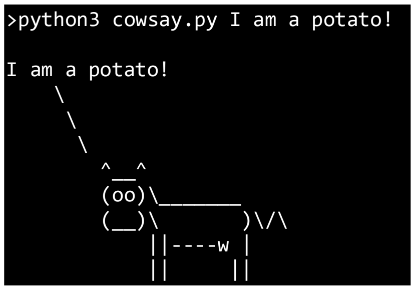
Your program must accept command line arguments. Command line arguments are captured as part of the `argv` variable found in the **sys** module. This can be accessed with `sys.argv` after you `import sys` (Review lecture slides for examples!).

The command line arguments that must be supported are as follows:

`python3 cowsay.py -l`	Lists the available cows
`python3 cowsay.py MESSAGE`	Prints out the MESSAGE using the default COW
`python3 cowsay.py -n COW MESSAGE`	Prints out the MESSAGE using the specified COW

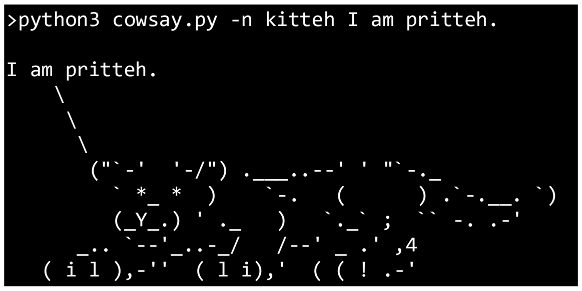
If a user calls for a cow that does not exist, the program should print out "Could not find ***[COWNAME]*** cow!"

Output Samples



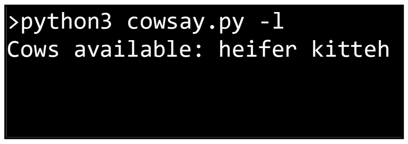
```
>python3 cowsay.py I am a potato!

I am a potato!
    \
     \
      \
        ^__^
        (oo)\_______
        (__)\       )\/\
            ||----w |
            ||     ||
```

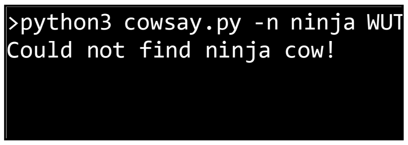


```
>python3 cowsay.py -n kitteh I am pritteh.

I am pritteh.
    \
     \
      \
       ("`-'  '-/") .___..--' ' "`-._
         ` *_ *  )    `-.   (      ) .`-.__. `)
         (_Y_.) ' ._   )   `._` ;  `` -. .-'
      _.. `--'_..-_/   /--' _ .' ,4
   ( i l ),-''  ( l i),'  ( ( ! .-'
```



```
>python3 cowsay.py -l
Cows available: heifer kitteh
```



```
>python3 cowsay.py -n ninja WUT
Could not find ninja cow!
```

Suggested Methods

The following methods are suggested to make development easier, but are not required:

`list_cows``(cows)`
Displays the available cows from a Python list of **`Cow`** objects.

`find_cow``(name, cows)`
Given a name and a Python list of **`Cow`** objects, return the **`Cow`** object with the specified **`name`**. If no such Cow object can be found, return **`None`**.

Cow Class

The **`Cow`** class facilitates the creation and use of cow objects by providing the following methods (which students must implement):

`__init__``(self, name)`
Initializes a cow object with name and image to be None

`get_name``(self)`
Returns the name of the cow. *Note: the name property should NOT have a setter.*

`get_image``(self)`
Returns the image used to display the cow (this should be called after the message has been displayed).

`set_image``(self, image)`
Sets the image used to display the cow.

Submissions

NOTE: Your output must match the example output *exactly*. If it does not, ***you will not receive full credit for your submission***!

Files: cowsay.py, cow.py, heifer_generator.py
Method: Submit on ZyLabs

Sample Output

(On next page)

```
>python3 cowsay.py Hello World!

Hello World!
    \
     \
      \
        ^__^
        (oo)\_______
        (__)\       )\/\
            ||----w |
            ||     ||

>python3 cowsay.py -n kitteh Hello World!

Hello World!
    \
     \
      \
       ("`-'  '-/") .___..--' ' "`-._
         ` *_ *  )    `-.   (      ) .`-.__. `)
         (_Y_.) ' ._   )   `._` ;  `` -. .-'
      _.. `--'_..-_/   /--' _ .' ,4
   ( i l ),-''  ( l i),'  ( ( ! .-'

>python3 cowsay.py -l
Cows available: heifer kitteh

>python3 cowsay.py -n ninja Hello world!
Could not find ninja cow!

>python3 cowsay.py Hello -n kitteh

Hello -n kitteh
    \
     \
      \
        ^__^
        (oo)\_______
        (__)\       )\/\
            ||----w |
            ||     ||
```