



Lab 08: The Cow Strikes Back

Overview

This lab's purpose is to provide students with experience in inheritance of classes and working with multiple classes. It is recommended that students use command line tools and editors for this lab (though it is not strictly speaking required). This lab will require students to build on their previous lab experience, in which a version of the **cowsay** utility was created.

Specification

Students will update the driver program file (cowsay.py) and also create two new classes - Dragon and IceDragon. The Dragon class should extend the Cow class, and IceDragon must be derived from Dragon. As before, heifer_generator.py is provided for you – but updated to handle the new Dragon class and its subtypes. (Please refer to specification for previous lab for a refresher.) Students may implement private attributes and methods if they chose to do so. This is not required – it is purely optional. No public attributes / methods should be added to the specification!

cowsay.py (Program Driver)

Your program must accept command line arguments as follows:

In addition, this version of the utility handles a special set of Cow-derived Dragon class (and its subclasses). Whenever a dragon-type cow is selected, the display of the message must be followed by a line stating whether or not the dragon is fire-breathing:





Cow Class

The Cow class must have all of the same methods as previously required (though students may add private methods). The methods are repeated here, briefly, for reference.

Dragon Class

The **Dragon** class must be derived from the **Cow** class and must make all of its methods available. In addition, Dragon must provide the following methods:

```
init (self, name, image)
```

Constructor; creates a new **Dragon** object with the given **name** and **image**.

can_breathe_fire()

This method should exist in **every Dragon** class. For the default **Dragon** type, it should always return **True**.

IceDragon Class

The **IceDragon** class must be derived from the **Dragon** class and must make all of its methods available:

```
__init__(self, name, image)
```

Constructor; creates a new **IceDragon** object with the given **name** and **image**.

can_breathe_fire()

For the **IceDragon** type, this method should always return False.

Submissions

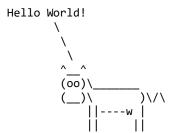
NOTE: Your output must match the example output *exactly*. If it does not, *you will not receive full credit for your submission*!

Files: cowsay.py, cow.py, dragon.py, ice_dragon.py, heifer_generator.py

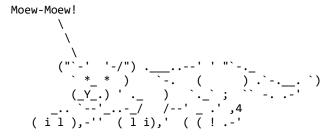
Method: Submit on ZyLabs

Sample Output

>python3 cowsay.py Hello World!



>python3 cowsay.py -n kitteh Moew-Moew!

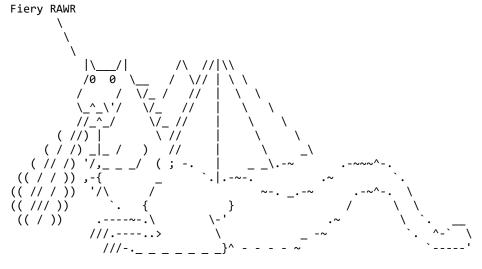


>python3 cowsay.py -1

Cows available: heifer kitteh dragon ice-dragon

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

>python3 cowsay.py -n dragon Firey RAWR



This dragon can breathe fire.

>python3 cowsay.py -n ice-dragon Ice-cold RAWR



This dragon cannot breathe fire.