

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

Ice-cold RAWR

      /\_/\
     /     \
    /  \   / \
   /    \ /   \
  /      ^     \
 /        ^      \
( // ) |         \
( / / ) | _ /      \
( // / ) ' \ _ _ _ / ( ; - .
(( / / )) ,-{
(( // / )) ' \ / _ _ _ _ _
(( /// )) . { }
(( / )) .-----\ \-'
          ///.-----.> \
          ///-.-.-.-.-}^-----~

This dragon cannot breathe fire.
```

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.

```
__init__(self, name)           // Constructor

get_name(self)                 // Returns name of this cow object

get_image(self)                // Return image for this cow object

set_image(self, image)         // Sets the image for this cow object to image
```

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!
```

Hello World!

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

Moew-Moew!

(" _ ' ' _ / ") . _ _ _ . . . _ ' ' _ " _ . _ . _
 (* *) _ _ _ . _ _ _ (_ _ _) _ _ _ . _ _ _ ; _ _
 (Y _ .) ' _
 (i l) , - ' ' (l i) , ' (! _ . ' _

```
>python3 cowsay.py -l
```

Cows available: heifer kitten dragon ice-dragon

```
>python3 cowsay.py -n ninja Hello world!
```

Could not find ninja cow!

```
>python3 cowsay.py -n dragon Firey RAWR
```

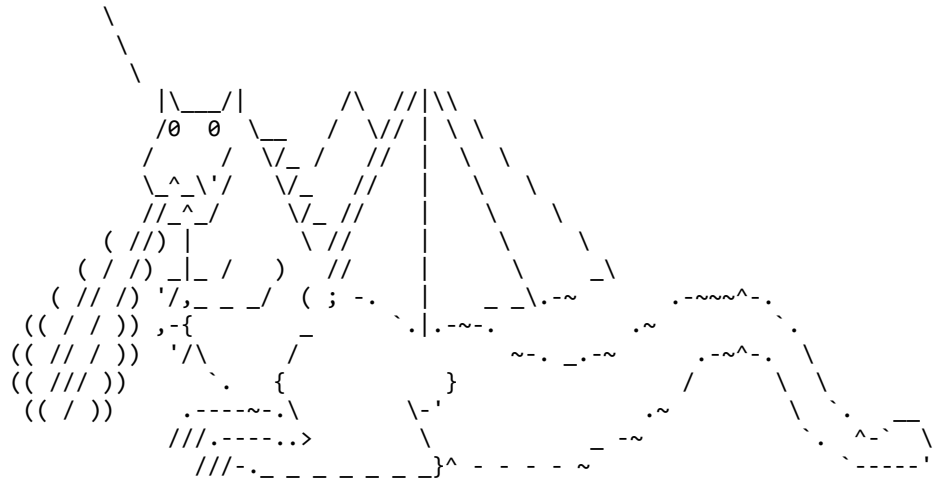
Fiery RAWR

The diagram illustrates a neural network architecture with multiple layers of nodes and connections. The nodes are represented by various symbols, including '0', '^', and '}', and are connected by solid and dashed lines. The architecture includes a dashed line at the top left, a vertical dashed line in the middle, and various symbols like '0', '^', and '}' indicating specific nodes or operations. The connections are represented by solid and dashed lines.

This dragon can breathe fire.

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

Ice-cold RAWR



This dragon cannot breathe fire.