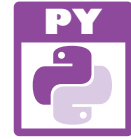


...cont'd

- 5 Start another Python script by making features of the class file available, then display its document string  
`from Bird import *`  
`print( '\nClass Instances Of:\n' , Bird.__doc__ )`
- 6 Next, add a statement to create an instance of the class and pass a string argument value to its instance variable  
`polly = Bird( 'Squawk, squawk!' )`
- 7 Now, display the common class variable value and call the class method to display this instance variable value  
`print( '\nNumber Of Birds:' , polly.count )`  
`print( 'Polly Says:' , polly.talk() )`
- 8 Create a second instance of the class, passing a different string argument value to its instance variable  
`harry = Bird( 'Tweet, tweet!' )`
- 9 Finally, display the common class variable value and call the class method to display this instance variable value  
`print( '\nNumber Of Birds:' , harry.count )`  
`print( 'Harry Says:' , harry.talk() )`
- 10 Save the file in your scripts directory, then open a Command Prompt window there and run this program – to see two instances of the Bird class get created



instance.py



Bird instance - polly



Bird instance - harry

```
Command Prompt
C:\MyScripts>python instance.py
Class Instances Of:
A base class to define bird properties.

Number Of Birds: 1
Polly Says: Squawk, squawk!

Number Of Birds: 2
Harry Says: Tweet, tweet!

C:\MyScripts>
```



The class variable `count` can also be referenced with `Bird.count`, but the encapsulated instance variable `sound` can only be accessed by calling an instance's `talk()` method.