```
# counter.py
#
# ICS 32 Winter 2014
# Code Example
#
# This module defines a class called Counter, which is a kind of object
# called a "counter", whose goal is to count how many times its "count"
# method has been called.  Each time you call "count", it increments
# its count and returns it, so each time you call "count", you'll receive
# a value one greater than the value you received the previous time.  It
# is also possible to reset the counter.
#
# Counter objects have one private attribute, _count, which specifies
# how many times its count() method has been called.
#
# If you run this module in IDLE, you would be able to do this in the
# interpreter:
#
#     >>> c1 = Counter()
#     >>> c1.count()
#     1
#     >>> c1.count()
#     2
#     >>> c1.count()
#     3
#     >>> c2 = Counter()
#     >>> c2.count()
#     1
#     >>> c1.count()
#     4
#     >>> c1.reset()
#     >>> c1.count()
#     1
#     >>> c2.count()
#     2


class Counter:
    def __init__(self):
        '''Initializes a Counter with a count of zero'''
        self._count = 0


    def count(self) -> int:
        '''Increments and returns the count'''
        self._count += 1
        return self._count


    def reset(self) -> None:
        '''Resets the counter, so that its value is zero'''
        self._count = 0
```