

The strategy is to convert the generator's variables and execution position into object fields and make `__next__` return each value one at a time, raising `StopIteration` at the end.

```
class CountUpTo:  
    def __init__(self, n):  
        self._n = n  
        self._i = 1  
  
    def __iter__(self):  
        return self  
  
    def __next__(self):  
        if self._i <= self._n:  
            result = self._i  
            self._i += 1  
            return result  
        else:  
            raise StopIteration  
  
def count_up_to(n):  
    return CountUpTo(n)
```