Python OO Practice Solution

Download our solution file

Serial

serial.py

```
"""Python serial number generator."""

class SerialGenerator:
    """Machine to create unique incrementing serial numbers.

    >>> serial = SerialGenerator(start=100)

    >>> serial.generate()
    100

    >>> serial.generate()
    101

    >>> serial.generate()
    102

    >>> serial.reset()

    >>> serial.generate()
    100
    """

    def __init__(self, start=0):
        """Make a new generator, starting at start."""

        self.start = self.next = start

    def __repr__(self):
        """Show representation."""

        return f"<SerialGenerator start={self.start} next={self.next}>"

    def generate(self):
        """Return next serial."""

        self.next += 1
        return self.next - 1

    def reset(self):
        """Reset number to original start."""

        self.next = self.start
```

Random Word

simple.txt

```
cat
dog
porcupine
```

complex.txt

```
# fruits

pear

# veggies

carrot
kale
```

wordfinder.py

```
"""Word Finder: finds random words from a dictionary."""

import random


class WordFinder:
    """Machine for finding random words from dictionary.

    >>> wf = WordFinder("simple.txt")
    3 words read

    >>> wf.random() in ["cat", "dog", "porcupine"]
    True

    >>> wf.random() in ["cat", "dog", "porcupine"]
    True

    >>> wf.random() in ["cat", "dog", "porcupine"]
    True
    """

    def __init__(self, path):
        """Read dictionary and reports # items read."""

        dict_file = open(path)

        self.words = self.parse(dict_file)

        print(f"{len(self.words)} words read")

    def parse(self, dict_file):
        """Parse dict_file -> list of words."""

        return [w.strip() for w in dict_file]

    def random(self):
        """Return random word."""

        return random.choice(self.words)


class SpecialWordFinder(WordFinder):
    """Specialized WordFinder that excludes blank lines/comments.

    >>> swf = SpecialWordFinder("complex.txt")
    3 words read

    >>> swf.random() in ["pear", "carrot", "kale"]
    True

    >>> swf.random() in ["pear", "carrot", "kale"]
    True

    >>> swf.random() in ["pear", "carrot", "kale"]
    True
    """

    def parse(self, dict_file):
        """Parse dict_file -> list of words, skipping blanks/comments."""

        return [w.strip() for w in dict_file
                if w.strip() and not w.startswith("#")]
```