Quiz 8

Due May 25 at 1:59am **Points** 24 **Questions** 6
Available until May 28 at 1:59am **Time Limit** 15 Minutes
Allowed Attempts 2

Instructions

Generators, first-class functions and decorators

The questions are all multiple choice or true/false. You may take the quiz twice, with 15 minutes for each attempt. Questions may differ between attempts. The highest score of the two attempts will be considered. You can only view the results once, immediately after finishing an attempt.

This is an open book test, however, you may not share the questions/answers of your quiz with others.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	8 minutes	20 out of 24

Correct answers are hidden.

Score for this attempt: **20** out of 24
Submitted May 23 at 7:43pm
This attempt took 8 minutes.

Question 1 4 / 4 pts

When should you use a decorator?

○ When you have a function that is too long.

- When you need your function to handle any errors that might occur from bad arguments from the user.
- When you want to save the function's output to a file.
- [✓ When you want to add behavior to a function that is not part of its primary purpose, which you may also want to apply to a number of different functions.]

Incorrect

Question 2

0 / 4 pts

What can you do with a generator object?

- Index into it.
- Import it so you can access it from other files.
- [✓ Iterate over it as many times as you want.]
- Iterate over it once.

Question 3

4 / 4 pts

Match the object with its type.

stuff = (c for c in "AlphAbEt" if c.isupper())	generator
junk = [x for x in range(100) if x%2]	list

paraphernalia = {'a': "aacorn", 'b': "broccolini"} dictionary

panoply = {1, 2, 3, 4, 5} set

Question 4

4 / 4 pts

A Python generator will remember the current values of its variables between calls.

[✓ True]

False

Question 5

4 / 4 pts

What does a generator do?

Supplies electrical power to the CPU.

Creates a data structure of items stored contiguously in memory.

Turns Python code into machine language.

[✓ Produces an iterable sequence of values.]

Question 6

4 / 4 pts

What would be the expected output of the following code?

```
def concat_strings():
    def cat(s1, s2):
        return s1+s2
    return cat


print(cat("pers", "picacious"))
```

- 'perspicacious'
- 'picaciouspers'
- 'pers'
- [✓ NameError: name 'cat' is not defined]

Quiz Score: **20** out of 24