Learning Quiz 15: More for Loop and Iteration Patterns

Due Oct 30 at 1pm **Points** 5 **Questions** 5 **Available** until Dec 4 at 11:59pm
Time Limit None **Allowed Attempts** Unlimited

Instructions

This quiz deals with loops. To keep track of your loops, always write out on paper what your values equal each time you go through a loop.

Prior to completing this quiz, be sure to read:

- Section 5.2, for Loop and Iteration Patterns (p. 131-139)

Please also go over Practice Problems 5.2-5.8 in the textbook (solutions at the end of the chapter) before attempting this quiz.

This quiz was created for learning purposes. You may attempt this quiz as many times as you would like. The highest score *prior to the deadline* will count towards the final course grade. No late submissions will be accepted.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	1 minute	5 out of 5
LATEST	Attempt 2	1 minute	5 out of 5
	Attempt 1	25 minutes	2.33 out of 5

Score for this attempt: **5** out of 5
Submitted Oct 28 at 3:06pm
This attempt took 1 minute.

Question 1

1 / 1 pts

In Knowledge and Learning Quiz 10, we went over nested for loops.

Consider the following nested for loop.

```
color = ['red', 'yellow', 'green', 'brown']
for i in range(3):
    for currentcolor in color:
        print(currentcolor[i], end = " ")
```

What prints?

○ red red red red yellow yellow yellow yellow green green green green

○ e e r r d l e o

○ 0 0 0 0 1 1 1 1 2 2 2 2

Correct!

● r y g b e e r r d l e o

○ red red red yel yel yel gre gre gre bro bro bro

○ red red red red yel yel yel yel gre gre gre gre

Question 2

1 / 1 pts

In Lecture 1, we briefly ways to update a variable by adding a number (say 1) to itself. Consider the following two lines of code:

```
num = 4
num = num + 1
```

At this point, num is 5.

You can similarly use the following **augmented assignment operator** to achieve the SAME result:

```
num = 4
num += 1
```

Here, num is still 5.

Starting with num = 4, test out EACH of the following augmented assignment operators to see what each one does:

```
num += 1
num *= 2
num -= 1
num /= 0.5
num %= 3
num **= 2
```

Fill in the blank with single symbol so that we end up with thisnum equal to 1.

thisnum = 13

thisnum % = 4

print(thisnum)

Answer 1:

Correct!

%

Question 3

1 / 1 pts

In Learning and Knowledge Quiz 10, we saw for-loops work through lists, strings, and range(). However, a common pattern in loops is to accumulate "something" in every iteration of the loop. That is, while we can use loops to help us work through lists, we can also use it to help us keep count of things. See **innerloopcount** and **outerloopcount** (both variables add 1 to itself each iteration of the loop).

```
mylist = ['apple','banana','pear']
color = ['red', 'yellow', 'green', 'brown']
```

```
outerloopcount = 0
innerloopcount = 0
for currentfruit in mylist:

    ## add 1 to outerloopcount BEFORE starting the inner loop
    outerloopcount += 1

    for currentcolor in color:

        ## add 1 to innerloopcount BEFORE printing
        innerloopcount += 1

        print('I have a', currentcolor, currentfruit)

        ## print text to help us keep track of the inner loop
        print('***** Inner loop count:', innerloopcount)

    ## after finishing the inner loop, print text
    print('***** Outer loop count:', outerloopcount)
```

Variables **innerloopcount** and **outerloopcount** are known as accumulators.

What is the resulting value of **savings** after the following lines are complete?

```
savings = 1357
for i in range(12):
    savings += 10
print(savings)
```

Correct!

1,477

orrect Answers

1,477 (with margin: 0)

Question 4

1 / 1 pts

In the following example, we (inefficiently) check all of the letters in word1 against all of the letters in word2 and print it only if they are equal. Note the nesting of the loop.

```
word1 = 'agglomerate'
word2 = 'confiscate'
comparisons = 0
```

```
for i in word1:
    for j in word2:
        if i == j:
            print(i, end = " ")
        comparisons += 1

print('\nWe made a total of', comparisons, 'comparisons.')
```

We can use print statements to help us keep track of our program.

```
word1 = 'agglomerate'
word2 = 'confiscate'
comparisons = 0
for i in word1:
    print('We are looking at letter', i,'in word1')
    for j in word2:
        print('We will compare', i, 'with', j)
        if i == j:
            print(i, end = " ")
        comparisons += 1

print('\nWe made a total of', comparisons, 'comparisons.')
```

Using the help of print() how many comparisons have been made by the time the second 'a' prints?

Correct!

88

orrect Answers

88 (with margin: 0)

Question 5

1 / 1 pts

Write a program that uses nested for-loops to print the following output:

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

```
for i in range(5, 0 ,-1):
    for j in range( i ,0,-1):
        print( j , end = " ")
    print()
```

Answer 1:

Correct!

0

Answer 2:

Correct!

i

Answer 3:

Correct!

j

Quiz Score: **5** out of 5