CC 1.3.1: Dynamic Typing

Dynamic Typing: Question 1

0/1 point (graded)

Consider the following code:

```
x=3
```

x did not exist in memory prior to this code. Which of the following does NOT occur?

Review the video at 2:15 if you need a reminder.

The object 3 is created.

A variable with name x is created.

The object 3 refers to the variable name x.

correct

The variable x is referred to object 3.

Dynamic Typing: Question 2

1/1 point (graded)

Consider the following code:

```
x=3
y=x
y=y-1
```

What does x equal?

3

correct

2

`y`

Dynamic Typing: Question 3

1/1 point (graded)

Consider the following code:

```
L1 = [2,3,4]
L2 = L1
L2[0] = 24
```

What does `L1` equal?

`[24, 3, 4]`

correct

`[2,3,4]`

This code contains an error.

Dynamic Typing: Question 4

1/1 point (graded)

Consider the following code:

```
L = [2,3,4]
M1 = L
M2 = L[:]
M1 is M2
```

What will this return?

`True`

`False`

correct

CC 1.3.2: Copies

Copies: Question 1

1/1 point (graded)

Consider the following code:

```
import copy
x = [1,[2]]
y = copy.copy(x)
z = copy.deepcopy(x)
y is z
```

What will this return?

`True`

`False`

correct

This code contains an error.

CC 1.3.3: Statements

Statements: Question 1

1/1 point (graded)

Consider the following code:

```
if False:
    print("False!")
elif True:
    print("Now True!")
else:
    print("Finally True!")
```

What does this print?

`"False!"`

`"Now True!"`
correct

`"Finally True!"`

`"Now True!"; "Finally True!"`

Statements: Question 2

1/1 point (graded)

Consider the following code:

```
if n%2 == 0:
    #blank#
else:
    print("odd")
```

Assume that n is a previously defined integer. Can you replace the `#blank#` line so that the code prints `"even"` if `n` is even, and `"odd"` if `n` is odd?

Enter your code here.

```
    print(“even")
```

CC 1.3.4: For and While Loops

For and While Loops: Question 1

1/1 point (graded)

Consider `bears = {"Grizzly":"angry", "Brown":"friendly", "Polar":"friendly"}`. Can you replace `#blank#` so the code will print a greeting only to friendly bears? Your code should work even if more bears are added to the dictionary.

```
for bear in bears:
    if #blank#:
    print("Hello, "+bear+" bear!")
else:
print("odd")
```

Enter your code here.

```
    if(bears[bear] == “friendly"):
```

For and While Loops: Question 2

1/1 point (graded)

Consider the following code:

```
is_prime = True
for i in range(2,n):
    if n%i == 0:
        #blank#
print(is_prime)
```

Can you fill in the #blank# line so the code will only print True if n is prime?

Enter your code here.

is_prime = True for i in range(2, n //2):   if n % i == 0:
print("impossible")   is_prime = False   break
print(is_prime)

For and While Loops: Question 3

1/1 point (graded)

Consider the following code:

```
n=100
number_of_times = 0
while n >= 1:
    n //= 2
    number_of_times += 1
```

```
print(number_of_times)
```

What will this print?

Enter your numerical answer here.

7

CC 1.3.5: List Comprehensions

List Comprehensions: Question 1

1/1 point (graded)

Consider the following code:

```
sum([i**2 for i in range(3)])
```

What will this output?

```
[0, 1, 2, 3, 4, 5, 6, 7, 8]
```

```
[0, 1, 4]
```

```
5
```

correct

```
36
```

List Comprehensions: Question 2

1/1 point (graded)

How can you use a list comprehension, including `if` and `for`, to sum the odd numbers from `0` through `9`?

Enter your code here.

sum(num for num in range(10) if num % 2 == 1)

CC 1.3.6: Reading and Writing Files

Reading and Writing Files: Question 1

1/1 point (graded)

Consider the following code:

```
F = open("input.txt", "w")
F.write("Hello\nWorld")
F.close()
lines = []
for line in open("input.txt"):
    lines.append(line.strip())
print(lines)
```

What does this print?

`Hello World`

`'Hello World'`

`['Hello', 'World']`

correct

`'Hello', 'World'`

This code contains an error.

CC 1.3.7: Introduction to Functions

Introduction to Functions: Question 1

1/1 point (graded)

Consider the following function:

```
def modify(mylist):
    mylist[0] *= 10
    return(mylist)
L = [1, 3, 5, 7, 9]
M = modify(L)
M is L
```

What is the value of the final line?

`True`

correct

`False`

This code contains an error.

CC 1.3.8: Writing Simple Functions

Writing Simple Functions: Question 1

1/1 point (graded)

Consider the function `intersect()` defined in the previous video, 1.3.8: Writing Simple Functions. What will `intersect([1,2,3], [3,4,5,6,7])` return?

`[1,2,3,4,5,6,7]`

`[3]`
correct

`3`

This code contains an error.

Writing Simple Functions: Question 2

1/1 point (graded)
Consider the following code:

```
def is_vowel(letter):
    if #blank#:
        return(True)
    else:
        return(False)
```

Can you replace `#blank#` in the second line so `is_vowel` becomes a function that takes a letter as input and prints whether a letter is a vowel (in `"aeiouy"`)?

Enter your code here.

letter in (“aeiouy")

Writing Simple Functions: Question 3

1/1 point (graded)

Consider the function call `is_vowel(4)`. Why would this not work?

`4` is not a vowel, leading to an error, and the function returns `False`.

`4` is not a letter, and Python only tests if single letters are in a `string`.

`4` is not a `string`, and Python cannot test if an `int` is in a `string`.
correct

All the above.

None of the above.

Writing Simple Functions: Question 4

1/1 point (graded)

Consider the following proposed emendation of `is_vowel`:

```
def is_vowel(letter):
    if type(letter) == int:
        letter = str(letter)
    if letter in "aeiouy":
        return(True)
    else:
        return(False)
```

Does this properly accommodate objects of type `int` for use with `is_vowel`? For example, will `is_vowel(4)` produce a correct answer?

Yes

correct

No

Writing Simple Functions: Question 5

1/1 point (graded)

Recall that

$n!$

("

n

factorial") is defined as the product of all integers

$1,\ldots,n$

. Additionally, by definition,

$0!\equiv 1$

.

Let's create a factorial function. Consider the following code:

```
def factorial(n):
    if n == 0:
        return 1
    else:
    N = 1
        for i in range(1, n+1):
            #blank#
        return(N)
```

Can you fill in the `#blank#` to complete the function as described above?

Enter your code here.

N = N*I

CC 1.3.9: Common Mistakes and Errors

Common Mistakes and Errors: Question 1

1/1 point (graded)

When you encounter an error in Python, what should you do?

Read the error message.

Try `help()` or `dir()`.

Use Google or StackOverflow to find an answer.

Search the course discussion forum and post a question if yours hasn't been asked.

All of the above.
correct