CIS 210 Fall 2018  Week 4 Review

>>> isinstance(101, float) == True       #booleans, built-in func, type
False

>>> isinstance(101, float)               #coding style
False

Given the following Python code:

>>> x = 'CIS 210'                           #assignment, built-in func, memory
>>> id(x)
4391509160
>>> y = x
>>> id(y)
??-1
>>> x = 'the end'
>>> id(x)
??-2
>>> y
??-3

4391509160 refers to a(n)

a) assignment statement     b) function     c) None type     d) keyword     e) memory location

The value printed at ??-1 will also be 4391509160 (yes or no);
the value printed at ??-2 will also be 4391509160 (yes or no).

a) yes/yes                  b) no/no                  c) yes/no                  d) no/yes

The value printed at ??-3 will be

a) 4391509160            b) 'CIS 210'            c) 'the end'             d) None

The following Python code                          # dynamic typing, strong typing, overloaded operators

>>> x = 'hi'
>>> x = 0
>>> x = x < 0

is an example of what feature of Python?

a) strong typing          b) dynamic typing          c) operator overloading
                          d) loops                      e) conditionals

The following Python code

>>> x = 'hi' + '-' + 'bye'
>>> y = 99 + 100

demonstrates what feature of Python?

a) strong typing  
b) dynamic typing  
c) operator overloading  
d) weak typing  
e) static typing

The following Python code

```python
>>> x = 'hi' + 99
Traceback (most recent call last):
  File "<pyshell#41>", line 1, in <module>
    x = 'hi' + 99
TypeError: must be str, not int
```

is an example of what feature of Python?

a) strong typing  
b) dynamic typing  
c) operator overloading  
d) weak typing  
e) static typing

What will be printed when the following Python code is executed?

```python
n = 5
mysum = 0
for ctr in range(1, n):
    myctr = mysum + ctr

print(mysum)
0
```

This code does not work as intended. This bug may be attributable to Python’s

a) strong typing  
b) dynamic typing  
c) operator overloading  
d) weak typing  
e) static typing

What will be printed when the following Python code is executed?

```python
n = 5
mysum = 0
for ctr in range(1, n):
    mysum = mysum + ctr

print(mysum)
10
```

This code is an example of  

# accumulator pattern

a) accumulator pattern  
b) TypeError  
c) conditional  
d) indefinite iteration  
e) infinite loop
Given the following Python code:

```python
0 import math

1 def isInCircle(x, y, r):
   2     '''(number, number, number) -> ??
   3     Returns True if point (x, y) is in
   4     the circle with radius r.
   5     >>> isInCircle(0, 0, 1)
   6     True
   7     >>> isInCircle(.5, .5, 1)
   8     True
   9     >>> isInCircle(1, 2, 1)
  10     False
  11     '''
  12     d = math.sqrt(x**2 + y**2)
  13     isIn = d <= r
  14     return isIn

Complete the type contract:

Which code would give the same results as `isInCircle` lines 12-14 (changes are in bold)?

a) `d = math.sqrt(x**2 + y**2)
return d = r`

b) `d = math.sqrt(pow(x, 2) + pow(y, 2))
return d <= r`

c) `d = math.sqrt(x**2 + y**2)
return d < r`

d) `d = math.sqrt(pow(x, 2) + pow(y, 2))
isIn = d < r
return isIn`

given the following Python code:

```python
# order of operations, type

1 - >>>> ftemp = 212
2 - >>>> ctemp = (ftemp - 32) * 5/9
3 - >>>> ctemp = ftemp - 32 * 5/9

The value of ctemp will [??] from line 2 to line 3; the type of ctemp will [??] from line 2 to line 3

a) stay the same/change
b) change/stay the same
c) stay the same/stay the same
d) change/change
```
Given the following Python code:

```
def q24(s):
    '''
    (??) -> ??
    
    Test function.
    
    >>> q24('The quick brown fox')
    ??
    >>> q24('Hello, world.')
    ??
    '''
    result = 999
    for i in range(len(s)):
        if s[i] == 'E' or s[i] == 'e':
            result = i
    return result

q24('Hello')
```

Complete the type contract for `q24`:

```
(str) -> int
```

Executing this function will

a) Return the number of occurrences of 'e' in `s`, or 999 if none.
b) Return the number of occurrences of 'E' in `s`, or 999 if none.
c) Return the sum of a) and b), or 999 if none.
d) Return the position of the first occurrence of 'e' or 'E' in `s`, or 999 if none.
e) Return the position of the last occurrence of 'e' or 'E' in `s`, or 999 if none.

The first time the for loop executes, the value of `i` is

a) 'H'   b) 0   c) 1   d) 4   e) 5

The first time the for loop executes, the value of `s[i] == 'E' or s[i] == 'e'` is

a) 'E'   b) 'e'   c) True   d) False   e) 'False'

To determine this value, Python evaluated

a) `s[i] == 'E'
    b) `s[i] == 'E'
    c) `s[i] == 'e'
    d) `result += 1`
Given the following Python code:

def q29(s1):
    '''(str) -> str

    s2 = ''
    for ch in s1:
        if ch not in s2:
            s2 += ch

    return s2

Which brief description is appropriate for q29?

a) copies s1 to s2; returns s2
b) copies all characters except the last character in s1 to s2; returns s2
c) copies 1st occurrence of each character in s1 to s2; returns s2
d) determines whether s1 is an empty string
e) creates and returns s2, a string of the characters that repeat (occur more than once) in s1

Given the following Python code:

def q30(score):
    ''' exam function '''
    gradepoint = 0
    if score >= 90:
        gradepoint = 4
    elif score >= 80:
        gradepoint = 3
    elif score >= 70:
        gradepoint = 2
    elif score >= 60:
        gradepoint = 1

    return gradepoint

What is the result of executing `>>> q30(80)`?

a) 4      b) 3      c) 2      d) 1      e) NameError
Given the following UNTESTED Python code:

def q3(myStr):
    '''final exam function'''
    newStr = ''
    for ch in myStr:
        if ch not in newStr:
            newStr += ch
    return newStr

What will be the result of executing

```python
>>> q3('abab')
```

a) 'abab' b) 'ab' c) 'ba' d) 'a' e) 'b'

Given the following Python code:

```python
# variable scope, namespaces
1  def isOdd(i):
2      '''(int) -> bool
3      exercise
4      '''
5      return i % 2 != 0
6  def q8(msg):
7      '''(str) -> ??
8      exercise
9      '''
10     odd_ct = 0
11     for ch in msg:
12         if isOdd(int(ch)):
13             odd_ct += 1
14     return odd_ct
15  def main():
16      '''exercise'''
17     code = '001100001100'
18     print(q8(code))
19     return None
20  
21  Complete the type contract for q8:
22     int
23  What will be the result of executing
24 >>> main()
25     4
```
What would be the result of executing `print(msg)` between lines 12 and 13?

a) NameError  b) 0  c) '001100001100'  d) str

print(ch) between lines 14 and 15 the first time the for loop is executed?

a) '0'  b) '1'  c) 'm'  d) 0  e) 1

print(i) between lines 5 and 6 the first time isOdd is executed?

a) '0'  b) '1'  c) 'm'  d) 0  e) 1

print(odd_ct) between lines 5 and 6 the first time isOdd is executed?

a) '0'  b) '1'  c) NameError  d) 0  e) 1

print(code) between lines 24 and 25 when main is executed?

a) 0  b) 4  c) 8  d) NameError  e) '001100001100'

print(msg) between lines 24 and 25 when main is executed?

a) 0  b) 4  c) 8  d) NameError  e) '001100001100'

While function q8 is executing, odd_ct exists in a/the _____ namespace.

a) local  b) global ('__main__')  c) built-in

Functions main, q8, and isOdd exist in a/the _____ namespace.

a) local  b) global ('__main__')  c) built-in

The decimal representation of binary 1111 is 15

# binary representation of numbers

The binary representation of decimal 24 is 11000

Practice (re-)writing any of the code from projects so far. Code should be same or very similar to posted solutions.