CIS 122 Practice Final Exam KEY
Spring 2017
(45 questions all same point value)

Notes in italics indicate which concepts each question or group of questions is checking for understanding. Each concept is listed only the first time it appears, though many concepts are repeated multiple times throughout the exam.

(1-6) CONCEPTS:
-- assignment statement
-- variables name values
-- evaluating an expression returns a value
-- literal values evaluate to themselves
-- variables evaluate to their value
-- calling/evaluating a simple built in function
-- string data type
-- integer data type
-- strings are a sequential data type; index operator
-- runtime errors (distractors)

(1-6) Given the following Python code:

```python
>>> lake = 'Odell'
```

What is the result when the following Python code is executed?

1) >>> lake
   a) 'Odell'   b) 'lake'   c) lake   d) NameError   e) Odell

2) >>> print(lake)
   a) 'Odell'   b) 'lake'   c) lake   d) NameError   e) Odell

3) >>> 'lake'
   a) 'Odell'   b) 'lake'   c) lake   d) NameError   e) Odell

4) >>> len('lake')
   a) 3   b) 4   c) 5   d) 6   e) NameError

5) >>> len(lake)
   a) 3   b) 4   c) 5   d) 6   e) NameError

6) >>> lake[1]
   a) 'Odell'   b) NameError   c) 'O'   d) IndexError   e) 'd'
(7) CONCEPTS:
-- function names are variables that name a function value
-- evaluating a variable (function name) v. calling a function

7) What will be printed when the following Python code is executed in the Shell?

```python
>>> pow
```

a) pow  b) <built-in function pow>  c) NameError  d) <class 'int'>
e) 'pow'

(8) CONCEPTS:
-- object type (here: integer) determines functions that apply
-- runtime error

8) What is the result when the following Python code is executed?

```python
>>> len(97403)
```

da) TypeError  b) NameError  c) IndexError  d) 4  e) 5

(9-10) CONCEPTS:
-- Boolean expressions and values; relational operators

(9) Given the following Python code:

```python
>>> greeting = 'hi'
```

What is the result when the following Python code is executed?

9) ```python
>>> greeting[0] == greeting[-1]
``` 

a) greeting  b) 'hi'  c) True  d) False  e) SyntaxError

10) ```python
>>> greeting[1] == greeting[-1]
``` 

a) greeting  b) 'hi'  c) True  d) False  e) SyntaxError

(11) CONCEPTS:
-- for loop
-- print function

11) When the following Python code is executed, what will be printed?

```python
nextclass = '123'
for ch in nextclass:
    print(ch*2, end='')
```
When the following Python code is executed, what will be printed?

```python
days_in_month = input()
type(days_in_month)
```

a) int  b) float  c) str  d) list  e) function

What will be the value of `ttlc` after the following Python code is executed?

```python
r = 4
g = r // 2
p = r + g
ttlc = r + g + p
```

a) 12  b) 12.0  c) 0  d) 1  e) 1.0
Given the following Python code:

```python
from turtle import *
def q14(n):
    '''(int) -> None
    Exam function.'''
    le = 100
    angle = 360 / n
    for i in range(n):
        fd(le)
        lt(angle)
    return None

>>> q14(4)
```

14) At line 8 the value of `n` is
   a) undefined       b) 4       c) 4.0     d) 100

15) After lines 8 and 9 execute, the values of `le` and `angle` are
   a) undefined       b) 100, 90   c) 100.0, 90.0   d) 100, 90.0

16) After line 10 executes the first time, the value of `i` is
   a) undefined       b) 0       c) 1       d) 0.0     e) 1.0

17) When the function is done executing,
   a) a drawing of a square is returned       b) None and a drawing of a square are returned
   c) None is returned and a square is drawn as a side effect
   d) function is an infinite loop

18) To implement forward movement for a variety of lengths, an additional parameter (`le`) is added to the function header. What other lines in the function would need to be changed?
   a) No other lines       b) 4, 8, 11, 12   c) 4, 8, 11   d) 4, 8       e) 4, 11

19) **CONCEPTS:**
    -- floating point operations
19) What will be the result when the following Python code is executed?

```python
bill = 100.00
tip18 = bill * .18
tip20 = bill * .20
print(tip18, tip20)
```
   a) 18 20       b) 18.0 20     c) 18 20.0     d) 18.0 20.0
(20-22) CONCEPTS:
-- parameter passing and local variables

(20-22) Given the following Python code:

```python
1 def q20(w, x):
2     ''' (int, int) -> int
3     Exam function.
4     '''
5     w = 200
6     x = 300
7     result = w + x
8     return result
```

20) What is the result of executing the following Python code?

```python
>>> q20(2, 3)
```

a) undefined  b) 5  c) 500  d) NameError  e) TypeError

21) What is the result of executing the following Python code?

```python
>>> q20(3, 4)
```

a) 5  b) 7  c) 500  d) NameError  e) TypeError

22) What changes to function q20 are needed for it to return different results for different argument inputs?

a) change lines 1, 2  b) delete lines 6, 7  c) change lines 8, 9  d) delete lines 8, 9

(23-25) CONCEPTS:
-- random module
-- functions calling functions
(23-25) Given the following Python code:

```python
0  import random
1  def q23():
2      '''() -> (?)
3      Exam function
4      '''
5      n = random.randint(1, 290)
6      return n
7
8  def q24():
9      '''() -> (?)
10     Exam function
11     '''
12      n = random.randint(-178, 1)
13      return n
14
15  def q25():
16      '''() -> None
17     Exam function
18     '''
19      wc_data = q23()
20      temp_data = q24()
21
22      print(wc_data, '"t",
23          temp_data, '"t"
24      )
25
26  return None

>>> q25()

23) What value is returned?
   a) None    b) -178   c) 1   d) 290   e) 1

24) What are possible values for wc_data and temp_data at line 24?
   a) undefined   b) 5, 0   c) 5.0, 0.0   d) 100, 100   e) True, True

25) Complete the type contracts (right hand side) for functions q23 and q24:
   a) int, int   b) int, float   c) float, int   d) None, None   e) Boolean, Boolean
CONCEPTS:
-- functions returned values can be used in an expression
-- Python namespaces (separate x for each function)

(26-28) Given the following Python code:

```python
1  def q26(x):
2      '''() -> (??)
3
4      Exam function
5      '''
6      result = x * x
7      return result
8
9  def q27(x):
10     '''() -> (??)
11
12     Exam function
13     '''
14     x += 1
15     return x
16
17  def main():
18     '''() -> None
19
20     Exam function
21     '''
22     result = q27(q26(3))
23     print(result)
24     return None

>>> main()
```

26) What is the value of x after line 6 is executed?

a) 3 b) 6 c) 9 d) undefined

27) What is the value of x after line 14 is executed?

a) 3 b) 4 c) 9 d) 10 e) undefined

28) What is printed in the Shell?

a) 9 b) 9 c) 10 d) None e) nothing is printed

None
(29-32) **CONCEPTS:**
-- conditional statements
-- docstring (type contract)
-- testing

(29-32) After executing the following Python code:

```python
1 def q29(weight):
2     '''(number) -> (??)
3     '''
4     Exam function.
5     '''
6     cost = weight * 0.05
7     if weight > 100:
8         cost = cost - 1.50
9
10     return cost

>>> q29(100)
```

29) How many times was line 8 executed?

a) 0  
b) 1  
c) 2  
d) 0 or 1 or 2

e) cannot be determined

30) Complete the type contract:

a) number  
b) int  
c) float  
d) str  
e) None

31) Choose the best set of test cases for q29:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>b)</td>
<td>c)</td>
</tr>
<tr>
<td>q29 (0)</td>
<td>q29 (10)</td>
<td>q29 (0)</td>
</tr>
<tr>
<td>q29 (1)</td>
<td>q29 (20)</td>
<td>q29 (1)</td>
</tr>
<tr>
<td>q29 (20)</td>
<td>q29 (30)</td>
<td>q29 (2)</td>
</tr>
<tr>
<td>q29 (100)</td>
<td>q29 (40)</td>
<td>q29 (3)</td>
</tr>
<tr>
<td>q29 (200)</td>
<td>q29 (50)</td>
<td>q29 (4)</td>
</tr>
</tbody>
</table>

32) Which line(s) of code need(s) to be changed for the discount to apply to weights of 100 or more?

a) 7  
b) 2,7  
c) 7,8  
d) 6,7,8,10  
e) no code changes are needed

(33) **CONCEPTS:**
-- string method find and slice operator

(33) What is the result when the following Python code is executed?

```python
>>> s = 'University of Oregon'
>>> s[s.find('O') :]
```

a) 14  
b) 'UO'  
c) 'Oregon'  
d) ''  
e) 'O'
(34) CONCEPTS:
-- Boolean data type, values, relational operators

(34) Given the following Python code:

```python
1 if a > b:
2     c = 10
3 elif a < b:
4     c = 15
5 else:
6     c = 20
7 print(c)
```

If after the code is executed, the value 20 has been printed, what must be True?

a) a > b      b) a < b      c) a == b      d) c > b      e) c > a

(35-37) CONCEPTS:
-- revising code

(35-37) After executing the following Python code:

```python
1 def q35(s1, s2):
2     '''(str, str) -> (??)
3     '''
4     Exam function
5     field_width = 18
6     ttl_len = len(s1) + len(s2)
7     fits = ttl_len <= field_width
8     return fits

>>> q35('Hood', 'Sandy')
```

35) Complete the type contract:

a) str       b) int       c) float      d) Boolean      e) None

36) What is printed in the Shell?

a) 18       b) 9        c) True       d) False       e) nothing is printed

37) Which line(s) of code need(s) to be changed such that the field width is specified at function call time?

a) 1, 2      b) 2, 8     c) 1, 2, 8    d) 1, 2, 8, 9   e) 1, 2, 6

(38) CONCEPTS:
-- Python indenting is syntax; note common whitespace bug
Given the following Python code:

```python
def any_upper(astring):
    '''(str) -> Boolean
    Returns True if there is any upper-case character in astring, a string
    of alphabetic characters. Otherwise return False.
    '''
    for ch in astring:
        if ch.isupper():
            return True
    return False
```

Executing `any_upper` will return the correct result.

a) always  b) sometimes  c) never  d) cannot be determined

(39-40) CONCEPTS:
-- testing and debugging

Given the following UNTESTED Python code:

```python
def calc_avg(dataset):
    '''(string) -> float
    returns average of values in input string values, but zeros do not
    count at all
    >>> calc_avg('23')
    2.5
    >>> calc_avg('203')
    2.5
    >>>
    count = 0
    total = 0
    for n in dataset:
        if n != '0':
            total += int(n)
            count += 1

    avg = total / count
    return avg
```

When the following code is executed

```python
>>> calc_avg('303')
```
39) n will have the value(s)
   a) '3', '4', '5'  b) '3'  c) '3', '0', '3'  d) '3', '3'
   e) '1', '2', '3'

40) What value will be returned?
   >>> calc_average('303')
   a) 2.0  b) 2.5  c) 3.0  d) None

(41-42) CONCEPTS:
-- lists; list methods (often return None and mutate list as a side effect)

41) What is the result when the following Python code is executed?
   >>> uli = ['University of Oregon', 'Cal', 'Oregon State']
   >>> uli.sort()
   >>> uli
   a) ['University of Oregon', 'Cal', 'Oregon State']
   b) ['Oregon State', 'Cal', 'University of Oregon']
   c) ['Cal', 'Oregon State', 'University of Oregon']
   d) []
   e) None

42) What is the result when the following Python code is executed?
   >>> uli = ['University of Oregon', 'Cal', 'Oregon State']
   >>> uli = uli.sort()
   >>> uli
   a) ['University of Oregon', 'Cal', 'Oregon State']
   b) ['Oregon State', 'Cal', 'University of Oregon']
   c) ['Cal', 'Oregon State', 'University of Oregon']
   d) []
   e) None
Given the following Python code to process a text file:

```python
def processf(fname):
    '''Exam function'''
    with open(fname, 'r') as examf:
        ??
    return None
```

43) Complete the type contract:
   a) str, str  b) None, None  c) str, None  d) int, None  e) list, None

44) If `fname` is a text file where the first line is a header that identifies what kind of file it is, what code should be included in the function to move the file pointer past the file header?
   a) examf.read()  b) examf.readline()  c) examf.readlines()  d) examf.write()

45) What will be the result of executing the following Python code:

```python
def best(greeting):
    '''(str) -> None
    print a greeting
    '''
    start = greeting.find('Best')
    print(greeting[start:]
    return None

>>> best('CIS122: Best wishes for a pleasant summer.')</code>

   a) Best wishes for a pleasant summer.
   b) Best wishes for a pleasant summer.
   c) Best wishes for a pleasant summer.
   d) Best wishes for a pleasant summer.
   e) Best wishes for a pleasant summer.