CIS 122 Spring 2017 Example Final Exam Questions

Note: These questions are not a comprehensive study guide! They are given here to provide a sense of the types of questions that may be on the exam.

To prepare thoroughly for the exam you should review projects and project solutions, midterms 1 and 2 and example questions/keys, your own and posted class and lab notes, and readings and exercises from the text.

The final exam will be in-class, multiple choice questions (Scantron). No outside resources are allowed during the exam, with the exception of one index card of handwritten notes.

(1) Which of the following are Python expressions?
   a) $44 + 33$  b) $\text{round}(3.14159)$  c) $x = 99$  d) $a$ and $b$  e) all of these

(2) What is the result of executing the following Python code?

```
>>> example = 99
>>> nextvar = example / 9
>>> nextvar
```

a) 11  b) 11.0  c) 0  d) syntax error  e) none of these

(3-6) Given:

```
>>> temperature = 99.1
```

(3) What is the result of executing the following code?

```
>>> temperatur
```

a) name error  b) type error  c) float  d) number  e) 99.1

(4) What is the result of executing the following code?

```
>>> temperature *= 20
```

a) syntax error  b) type error  c) int  d) 119  e) 119.1

(5) What is the result of executing the following code?

```
>>> temperature += 20
```

a) syntax error  b) type error  c) float  d) 119  e) 119.1
(6) What is the result of executing the following code?

```python
>>> round(abs(temperature))
```

a) syntax error  b) type error  c) float  d) 99  e) 99.1

(7) Given:

```python
>>> x = 0
>>> x = x + 1
>>> x += 2
```

What is the result of executing the following code?

```python
>>> x
```

a) error  b) int  c) 2  d) 3  e) 3.0

(8) What is the result of executing the following code?

```python
>>> type(97403)
```

a) integer  b) float  c) string  d) boolean  e) list

(9) What is the result of executing the following code?

```python
>>> type('97403')
```

a) integer  b) float  c) string  d) boolean  e) list

(10) What is the result of executing the following code?

```python
>>> type(4 < 5)
```

a) integer  b) float  c) string  d) boolean  e) list

11) What is the result of executing the following code?

```python
>>> type(['hello', 'goodbye'])
```

a) integer  b) float  c) string  d) boolean  e) list
12) What is the result of executing the following code?

```python
>>> type(['hello', 'goodbye'][0])
```

a) integer  b) float  c) string  d) boolean  e) list

(13-14) Given:

```python
def q13a(x, y):
    '''(int, int) -> None

  Example question.
  '''
    a = x * y
    b = q13b(a)
    print(b)  #checkpoint 2
    return None

def q13b(x):
    '''(int) -> None

  Example question.
  '''
    print(x)  #checkpoint 1
    x *= 10
    print(x)
    return None
```

```python
>>> q13a(1, 2)
```

(13) What is printed at checkpoint 1?

a) 1   b) 2   c) 20   d) None   e) none of these

(14) What is printed at checkpoint 2?

a) 1   b) 2   c) 20   d) None   e) none of these
(15) Given (change from code in problem 13 is highlighted):

def q15a(x, y):
    '''(int, int) -> None

    Example question.
    '''
    a = x * y
    b = q15b(a)
    print(b) #checkpoint
    return None

def q15b(x):
    '''(int) -> None

    Example question.
    '''
    x *= 10
    print(x)
    return x #change

>>> q15a(1, 2)

What is printed at the checkpoint?

a) 1   b) 2   c) 20   d) None   e) none of these

(16-19) Given:

def q16(time):
    '''(number) -> ??

    Example question.
    '''
    if time < 10:
        return 'very nice'
    if time < 8:
        return 'terrific'
    if time < 6:
        return 'excellent'
    if time < 4:
        return 'outstanding'
    return 'ok'

(16) What should replace ?? in the function docstring of function q16?

a) number   b) string   c) integer   d) None   e) none of these
(17) What is the result of executing the following code?

>>> q16(2)

a) 'very nice'  b) 'terrific'  c) 'excellent'  d) 'outstanding'  e) 'ok'

(18) Given (changes from function q16 are bold):

```python
def q18(time):
    '''(number) -> ??
    Example question.
    '''
    message = 'ok'
    if time < 10:
        message = 'very nice'
    if time < 8:
        message = 'terrific'
    if time < 6:
        message = 'excellent'
    if time < 4:
        message = 'outstanding'
    return message
```

What is the result of executing the following code?

```python
>>> q18(2)

a) 'very nice'  b) 'terrific'  c) 'excellent'  d) 'outstanding'  e) 'ok'
```
(19) Given (changes from q18 are bold):

def q19(time):
    '''(number) -> ??
    Example question.
    '''
    if time < 10:
        message = 'very nice'
    elif time < 8:
        message = 'terrific'
    elif time < 6:
        message = 'excellent'
    elif time < 4:
        message = 'outstanding'
    else:
        message = 'ok'
    return message

What is the result of executing the following code?

```python
>>> q19(2)
```

a) 'very nice'  b) 'terrific'  c) 'excellent'  d) 'outstanding'  e) 'ok'

(20-22) Given the following Python code:

```
1) def two_digits(s):
2)     '''(str) -> boolean
3)     If there are fewer than two digits in
4)     a string, return False, else return True
5)     '''
6)     digitct = 0
7)     for ch in s:
8)         if ch.isdigit():
9)             digitct += 1
10)        else:
11)            return False
12)        if digitct <= 2:
13)            return False
14)        else:
15)            return True
```
(20) Choose the better set of tests for function two_digits:

(a) >>> two_digits('1234') >>> two_digits('5674') >>> two_digits('11aa') >>> two_digits('aa11') >>> two_digits('z9z') >>> two_digits('') >>> two_digits('nodigits')

(b) >>> two_digits('1234') >>> two_digits('123') >>> two_digits('12') >>> two_digits('5678') >>> two_digits('567') >>> two_digits('56') >>> two_digits('') >>> two_digits('')

The code as written does not work as described in the docstring.

(21) Executing function two_digits as written will cause what type of error?

a) syntax    b) runtime    c) logic    d) list    e) none of these

(22) Which lines of code need to be changed to correct the bugs in two_digits?

a) 11, 12, 14    b) 8, 15    c) 11, 12, 17    d) 9, 15    e) 10, 11

(23-25) Given:

```python
def q23(s):
    '''(str) -> ??
    Example question.
    '''
    res = []
    for ch in s:
        if ch.islower():
            res.append(ch)
        else:
            return res
name = 'Ducks'
```

(23) What should replace ?? in the function docstring?

a) int    b) string    c) boolean    d) list    e) none of these

(24) What will be the result of when the following code is executed?

```python
>>> q23(name)
```

a) []    b) ''    c) ['u', 'c', 'k', 's']    d) 'Ducks'    e) None
(25) Given (changes from q24 are bold):

def q25(s):
    '''(str) -> ??
    Example question.
    '''
    res = []
    for ch in s:
        if ch.islower():
            res.append(ch)
    return res

name = 'Ducks'

(a) What will be the result when the following code is executed?

>>> q25(name)

a) []  b) ''  c) ['u', 'c', 'k', 's']  d) 'Ducks'  e) None

(26-28) Given the following Python code:

def q26(myl):
    '''(list of numbers) -> ??
    Example question.
    '''
    check = 0
    for item in myl:
        if item < check:
            return True
    return False

numlist = [0, 10, 10, 20]

(26) What should replace ?? in the function docstring?

a) int   b) string   c) boolean   d) list   e) none of these

(27) What will be the result when the following code is executed:

>>> q26(numlist)

a) 4   b) 'numlist'   c) False   d) [10, 10]   e) True
(28) What will be the result when the following code is executed?

```python
>>> numlist = numlist.append(-50)
>>> print(numlist)
```

a) [0, 10, 10, 20]  b) [0, 10, 10, 20, -50]  c) [-50, 0, 10, 10, 20]  d) None  

e) none of these

(29-32) Given:

```python
t = [['name', 'Duk'], ['color', 'blue'], ['pos', [0, 0]]]
```

(29) What will be the result when the following Python code is executed?

```python
len(t)
```

a) 0  b) 3  c) 6  d) 7  e) cannot be determined

(30) What will be the result when the following Python code is executed?

```python
for item in t:
    if item[0] == 'color':
        print(item)
```

a) 'color'  b) 'blue'  c) ['color', 'blue']  

d) 'color', 'blue'  e) none of these

(31) How many times was the for loop in question (30) executed?

a) 0  b) 3  c) 6  d) cannot be determined  e) infinite

(32) What is the value of `t` after the following Python code is executed?

```python
t.append(['heading', 0])
```

a) [['name', 'Duk'], ['color', 'blue'], ['pos', [0, 0]]]  
b) [['name', 'Duk'], ['color', 'blue'], ['pos', [0, 0]], ['heading', 0]]  
c) None  d) error  e) cannot be determined

(33) A problem that may be difficult for a computer to solve is one that is

a) precisely specified  b) ambiguous  c) uses lists  d) uses a large data file  e) none of these
Text file `pykeywds.txt` contains the following data:

```
and    elif    if    print
as     else    import    raise
assert  except  in    return
break  exec    is    try
class  finally  lambda    while
continue  for    not    with
def    from    or    yield
del    global    pass
```

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

```python
with open('pykeywds.txt', 'r') as keyf:
    contents = keyf.readlines()
    print(type(contents))
```

a) int  b) float  c) string  d) boolean  e) list

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

```python
with open('pykeywds.txt', 'r') as keyf:
    for i in range(3):
        keyf.readline()

    nextline = keyf.readline
    print(type(nextline))
```

a) int  b) float  c) string  d) boolean  e) list
(36-38) Given the following Python code:

```python
1) def simple_blackjack():
2)     '''() -> None
3)     Final exam function
4)     '''
5)     card = random.randint(1, 11)
6)     total = card
7)     print('your total so far is:', total)
8)     resp = input('another card (y for yes)? ')
9)     while resp == 'y':
10)        card = random.randint(1, 11)
11)        total += card
12)        print('your total so far is:', total)
13)        resp = input('another card (y for yes)? ')
14)     if total > 21:
15)         result = 'lose'
16)     else:
17)         house = random.randint(1, 21)
18)         print('house total is:', house)
19)         house_score = 21 - house
20)         my_score = 21 - total
21)         if my_score < house_score:
22)             result = 'win'
23)         else:
24)             result = 'lose'
25)     print(result)
26)     return None
27) simple_blackjack()
```

(36) A possible value for `total` after line 7 is executed is

a) 0     b) 4     c) 'y'     d) 'n'     e) None

(37) How many times will the `while` loop that begins at line 11 be executed?

a) 0     b) 1     c) 2     d) cannot be determined     e) infinite

(38) Possible values for `result` at line 31 are

a) no value     b) 'win'     c) 'lose'     d) a or b or c     e) b or c