CIS 210 Fall 2019 Example Exam Questions  KEY

Note: These questions are not a comprehensive study guide! They are given here to provide a sense of the types of questions that will be on the CIS 210 final exam.

To prepare thoroughly for the final exam you should review projects and project solutions, the midterm exam, class notes, labs, and readings from the text.

The final exam will be given in the regular classroom, and will comprise multiple choice questions, short-answer questions, and questions where the solution will require you to write Python code according to the usual CIS 210 style guidelines.

No outside resources are allowed during the exam, with the exception of one 3x5” index card of handwritten notes.

(1)
(a) Replace the ?? with the result of executing the following code (indicate 'error' if the result would be a Python error message):

```python
>>> a = 'good morning'
>>> b = 'good evening'
>>> c = a
>>> a = a[0:4]
>>> b = 'good night'
>>> a
'good'
>>> c
'good morning'
```

(b) Given:

```python
if cost > 100:
    sign = '$$$$'
elif cost > 50:
    sign = '$$$'
elif cost > 25:
    sign = '$$
else:
    sign = '$'
```

Replace the ?? with the result of executing the following code when `cost` has the value 110 (indicate 'error' if the result would be a Python error message):

```python
>>> sign
```
(c) Replace the ?? with the result of executing the following code (indicate 'error' if the result would be a Python error message):

```python
g>>> vowels = ''
g>>> line = 'the quick brown'
g>>> for c in line:
    if c in ['a', 'e', 'i', 'o', 'u']:
        vowels += c
g>>> vowels
g'euio'
```

(d) Replace the ?? with the result of executing the following code (indicate 'error' if the result would be a Python error message):

```python
g>>> factors = [x for x in range(1,10) if x % 3 == 0]
g>>> factors
g[3, 6, 9]
```

(e) Replace the ?? with the result of executing the following code (indicate 'error' if the result would be a Python error message):

```python
g>>> x = 1
g>>> (x ** 2) == (x * 2)
False
```

(f) Replace the ?? with the result of executing the following code (indicate 'error' if the result would be a Python error message):

```python
g>>> name = "University of Oregon"
g>>> name = name.split()
g>>> new = name[0][0] + name[2][0]
g>>> new
'UO'
```
(2) Given:

def q2(slist):
   '''(list of str) -> ??'''
   slen = len(slist)
   ct = 0
   for s in slist:
      ct += s.count('x')
   return ct

li = ['CIS 415', 'CIS 422', 'CIS 425', 'CIS 4xx']

Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

>>> type(q2(li))
int
>>> q2(li)
2

(3) Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

>>> equakes = [['OR', 5.2], ['WA', 5.4], ['CA', 5.0]]
>>> equake = ['OR', 5.0]
>>> equakes.append(equake)
>>> equakes[0]
['OR', 5.2]

>>> equake = ['OR', 5.3]
>>> equakes = equakes.append(equake)
>>> equakes[-1]
error
(4) Given:

```python
def twice(x):
    '''(int) -> int
    Exam function.
    '''
    m = 2
    x = m * x
    return x
```

Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

```python
>>> x = 5
>>> twice(x)
10

>>> x
5

>>> m
error
```

(5) Given:

```python
def thrice(x):
    '''(int) -> int
    Exam function.
    '''
    m = 3
    print(m, x)
    x = m * x
    return x
```

Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):
>>> y = 10
>>> thrice(y)
3 10
30

>>> x
error

(6) Given:

```python
def q6(date):
    '''(str) -> str
    Exam function.
    '''
    d = date.split('/'
    newd = '{}-{}-{}'.format(d[2], d[0], d[1])
    return newd
```

Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

```bash
>>> q6('3/21/2016')
'2016-3-21'
```
(7) Supply the docstring (using CIS 210 format) for the following code:

You may use the file sample.txt:

Header1
Header2
Header3
44 55 66
77 88 99
#This line signals end of data.
#This line is a footer.

def q7(f, c):
    '''(str, str) -> None

    Print each line of file f that begins with c
    (not including header lines and after whitespace
    is removed). None value is returned.

    For example,
    >>> q7('sample.txt', '#')
    #This line signals end of data.
    #This line is a footer.
    '''
    with open(f, 'r') as myf:
        for i in range(3):  #move past header lines
            myf.readline()

        for nextline in myf:
            nextline = nextline.strip()
            if nextline[0] == c:
                print(nextline)

    return None
(8) Given:

```python
def q8(astr):
    '''(str) -> str

    Exam function.
    '''
    lenstr = len(astr)
    if lenstr == 0:
        return astr
    elif lenstr == 1:
        return astr
    else:
        return astr[-1] + q8(astr[0:lenstr-1])
```

Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

```python
>>> q8('CIS 210')
'012 SIC'
```

Give three good test cases for the function; for each test case, indicate its equivalence class.

(1) 'x' - string length 1

(2) '' - string length 0 (empty string)

(3) 'ab' - simple string
(9) Given:

def addlist(m, n):
    '''(list, list) -> None
    Exam function.
    '''
    for i in range(len(m)):
        if i < len(n):
            m[i] += n[i]
    return None

def q9(x, y):
    '''(list, list) -> None'''
    z = x
    addlist(x, y)
    print(x)
    print(y)

    w = []
    for item in y:
        w.append(item / 10)
    print(w)

    return None

Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

```python
>>> a = [30, 20, 10, 0]
>>> b = [10, 20, 30]
>>> q9(a, b)
[40, 40, 40, 0]
[10, 20, 30]
[1.0, 2.0, 3.0]
```
Given:

```python
import random

def createCentroids(k, datadict):
    '''(int, dict) -> list (of dict values)

    Create a starter list of k centroids for a k-cluster algorithm by randomly choosing from the items in datadict. The starter list is returned.
    '''
    centroids = []
    centroidCount = 0
    centroidKeys = []

    while centroidCount < k:
        rkey = random.randint(1, len(datadict))
        if rkey not in centroidKeys:
            centroids.append(datadict[rkey])
            centroidKeys.append(rkey)
            centroidCount += 1

    return centroids
```

Replace the ??s with the results of executing the following code in the Python Shell, for example (indicate ‘error’ if the result would be a Python error message):

```python
>>> d = {1: [5.2], 2: [2.9], 3: [3.4], 4: [2.7]}
>>> d[1] = [4.4]
>>> d[5] = [2.5]
>>> len(d)
5
>>> createCentroids(2, d)
[[4.4] [2.7]]
```
def even(i):
    '''(int) -> Boolean

    return True if i is an even number

    >>> even(4)
    True
    >>> even(3)
    False
    '''
    return (i % 2) == 0

def median(alist):
    '''(list of numbers) -> number

    Return median of alist (for alist of len > 0).

    >>> median([5, 7, 1, 3]) # even number of items
    4.0
    >>> median([1, 2, 2, 3, 99]) # odd number of items
    2
    >>> median([99]) # list with 1 item
    99
    >>> median([99, 101]) # list with 2 items
    100.0
    >>> median([0, 0, 0, 0]) # all items are the same
    0.0
    '''
copyl = alist[:]
copyl.sort()
copylen = len(copyl)

if even(copylen):
    rmid = copylen // 2
    lmid = rmid + 1
    medi = (copyl[lmid] + copyl[rmid]) / 2
else:
    mid = copylen // 2
    medi = copyl[mid]

return medi

Function **median** in Python file **example.py** contains an error (in the code, not the docstring).

a) add Python code to **example.py** that will automatically test the **median** function.

```python
import doctest
print(doctest.testmod())
```

b) circle the test case(s) in the docstring of **median** that will cause (an) error(s):

```python
>>> median([5, 7, 1, 3]) # even number of items
    4.0
>>> median([99, 101]) # list with 2 items
    100.0
```

c) what kind error is this / errors are these (circle all that apply):

- documentation
- syntax
- runtime
- logical

d) fix the bug in the code of function **median**

```python
lmid = rmid - 1
```
(12) Given:

```python
from turtle import *

def q12(x, y):
    '''(number, number) -> None
    Exam function.
    '''
    pendown()
    hideturtle()
    setpos(0,0)
    setpos(x,0)
    setpos(x,y)
    setpos(0,y)
    return None
```

Replace the `??`s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

```python
>>> q12(50, 50)
```

```python
+---+
|   |
|   |
+---+
```
def q13(astr):
    '''(str) -> (list of str)
    Exam function.
    '''
    countdict = {}
    for item in astr:
        if item in countdict:
            countdict[item] += 1
        else:
            countdict[item] = 1

    countlist = countdict.values()
    maxcount = max(countlist)

    mli = []
    for item in countdict:
        if countdict[item] == maxcount:
            mli.append(item)

    return mli

mystr = 'Simple substitution encryptions can be broken by \nlooking for frequently used characters.'

teststr = 'abbcdeffg'

Replace the ??s with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message):

```python
>>> q13(teststr)
['b', 'f', 'd']
```