CIS 210 Fall 2020 Example 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 CIS 210 final exam.

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

(1)

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

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

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

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

```python
>>> x = 1
>>> (x ** 2) == (x * 2)
??
```

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

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

def q2(slist):
    '''(slist: list) -> ??'''
    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):

```python
>>> type(q2(li))
??
>>> q2(li)
??
```

(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):

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

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

```python
def twice(x):
    '''(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)
??

>>> x
??

>>> m
??
```

(5) Given:

```python
def thrice(x):
    '''(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)
??

>>> x
??

(6) Given:

def q6(date):
    '''(date: str) -> str
    Exam function.
    '''
    d = date.split('/
    newd = f'{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):

>>> q6('11/26/2020')
??
(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):
    '''
    ...
    with open(f, 'r') as myf:
        for i in range(3):
            myf.readline()

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

    return
def q8(astr):
    '''(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')

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

(1) ??

(2) ??

(3) ??

(9) Given:

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

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

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

    return

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):

>>> a = [30, 20, 10, 0]
>>> b = [10, 20, 30]
>>> q9(a, b)
??
Given:

```python
import random
def createCentroids(k, datadict):
    '''(k: int, datadict: 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)
??
>>> createCentroids(2, d)
??
```
def even(i):
    """(i: int) -> bool
    return True if i is an even number
    >>> even(4)
    True
    >>> even(3)
    False
    ""
    return (i % 2) == 0

def median(alist):
    """(alist: list) -> 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.

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
??
>>> median([99, 101]) # list with 2 items
??
```

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**

??
(12) Given:

```python
from turtle import *

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

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):

```
(13) Given:

```python
def q13(astr):
    '''(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

myst = 'Simple substitution encryptions can be broken by looking 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):

```
When the following code is executed in the Python Shell,

```python
>>> mypi
??-1
>>> mypi = 3
>>> mypi
??-2
>>> w = mypi
>>> id(w) == id(mypi)
??-3
>>> type(w)
??-4
>>> mypi = [3]
>>> id(w) == id(mypi)
??-5
```

(a) Replace ??-1 with the correct value:

a) 3  
b) [1, 2, 3, 4]  
c) True  
d) False  
e) NameError

(b) Replace ??-2 with the correct value:

a) 3  
b) [1, 2, 3, 4]  
c) True  
d) False  
e) NameError

(c) Replace ??-3 with the correct value:

a) 3  
b) [1, 2, 3, 4]  
c) True  
d) False  
e) NameError

(d) Replace ??-4 with the correct value:

a) int  
b) float  
c) str  
d) boolean  
e) tuple
(e) Replace ??–5 with the correct value:

a) 3  

b) [1, 2, 3, 4]  
c) True  
d) False  
e) NameError

(f) Which statement is correct?

a) `mypi` is defined in the `global` namespace; `w` is defined in the `global` namespace.

b) `mypi` is defined in the `global` namespace; `w` is defined in the `local` namespace.

c) `mypi` is defined in the `local` namespace; `w` is defined in the `global` namespace.

d) `mypi` is defined in the `local` namespace; `w` is defined in the `local` namespace.

e) `mypi` is defined in the `local` namespace; `w` is no longer defined.

(15) `TypeError`, `NameError`, `ZeroDivisionError` are examples of which type of error?

a) syntax  
b) runtime  
c) logical/semantic  
d) regression  
e) integrated

(16) The following Python code

```python
>>> x = 'hi'
>>> x = 0
>>> x = x < 0
```

demonstrates which characteristic of Python?

a) strong typing  
b) dynamic typing  
c) operator overloading  
d) static typing  
e) conditionals
(17) The following Python code

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

demonstrates which characteristic of Python?

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

(18) 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
```

demonstrates which characteristic of Python?

a) strong typing    b) dynamic typing    c) operator overloading
d) static typing    e) conditionals
(P1) Write a function, `mypow`, that returns \( x \) to the \( n \)th power, where \( x \) and \( n \), non-negative integers, are function parameters. Do not use Python's built-in power operator (** or `pow`).

a) Write a simple example of use for `mypow`:

b) Write one edge test case for `mypow`:

c) Write the function header for `mypow`:

d) Write the type contract for `mypow`:

e) Write an algorithm for solving `mypow` (natural language/pseudocode):

f) Write the rest of (i.e., do not include header and docstring) the Python code for `mypow`:

(P2) Write function `strReverse`, which will take a single string argument, \( s \), and return a string which is the reverse of \( s \). (Do NOT use `[:: -1]` or any type conversion functions.) Code must be written according to CIS 210 style guidelines, including a docstring with a type contract, brief description and at least two examples of use – basic and edge/boundary.