**CIS 210 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 final exam you should review projects and project solutions, quizzes, midterm and solution, class notes and exercises, lab notes and exercises, and readings and exercises from the text.

The final exam 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 index card of handwritten notes.

1-2. What will be the result of executing the following Python code:

```python
def q1(raindata):
    '''Exam function'''
    datactr = 0
    for rain in raindata:
        if rain >= 0:
            datactr += 1
    return datactr
```

```python
>>> q1([-1, 1, 3.0, -5, 2])
```

1. a) 0  b) 1  c) 2  d) 3  e) 5

2. An appropriate type contract for q1 is:

   a) (list) → list
   b) (list) → float
   c) (float) → float
   d) (list) → int
   e) (int) → int

3-4. What will be printed when the following Python code is executed?

```python
yellow_ct = 0
for ctr in range(3):
    yellowCt = yellow_ct + 1
print(yellow_ct)
```

3. a) 0  b) 1  c) 2  d) 3  e) nothing will be printed

4. The code in question 3 does not work as intended. This is due to Python's:

   a) static typing  b) dynamic typing  c) strong typing  d) weak typing
5-6. Given the following Python code:

```python
def q5(li, idx, item):
    '''Exam function.'''
    newli = []

    if li == []:
        newli = [item]
    elif idx >= len(li):
        newli = li + [item]
    elif idx == 0:
        newli = [item] + li
    else:
        for i in range(idx):
            newli.append(li[i])
        newli = [i for i in li[:idx]]  #checkpoint 1
        newli.append(item)
        newli = newli + li[idx:]

    return newli  #checkpoint 2
```

When the following code is executed

```python
>>> q5([1, 2, 3, 4], 2, 5)
```

5. What is the value of item at checkpoint 1?

a) []  b) [5]  c) 1  d) 2  e) 5

6. What value is returned (what is the value of newli at checkpoint 2)?

a) []  b) [1, 2, 5, 3, 4]  c) [1, 2, 3, 4, 5]

d) [1, 2, 2, 3, 4, 5]  e) [1, 3, 4]

7. Given the following Python code, what is likely to be the case for f1, f2, and f3?

```python
def foo():
    res = f1()
    if f2(res):
        f3()
    return res
```

a) f1 returns a value  b) f2 returns Boolean value

c) f3 causes a side effect  d) a) and b) only  e) a) and b) and c)
8. The binary representation of the decimal number 30 is:
   a) 210  b) 11110  c) 01111  d) 10110  e) 01101

9. The decimal equivalent of binary number 1010 is:
   a) 5  b) 8  c) 10  d) 15  e) 20

10-11. Given the following Python code:

   def q10(s):
       ''' UNTESTED exam function '''
       vowels = ['a', 'e', 'i', 'o', 'u']
       vowelct = 0
       for item in s:
           if item in vowels:
               vowelct += 1
       return vowelct

10) What will be the result of executing the following code:

>>> q10('the quick brown fox')

   a) 0  b) 1  c) 5  d) 19  e) None

11) (A) test(s) that would reveal the bug in the code is (are):

   a) >>> q10('aeiou')  b) >>> q10('bcdfg')
   c) >>> q10('the quick brown fox')  d) (a) and (b)  e) (a) and (b) and (c)

12-13. Given the following Python function:

   def getmax(li):
       '''(list of ints) -> int

       UNTESTED - Returns max int from non-empty list li.
       '''
       max_so_far = li[0]
       idx = 1
       while idx < len(li):
           if li[idx] > max_so_far:
               max_so_far = idx
               idx += 1
       return max_so_far
12. A test that would reveal the bug in this code is

a) `getmax([0, 1, 2, 3, 4])`

b) `getmax([4, 3, 2, 1])`

c) `getmax([99])`

d) all of these

e) none of these

13. This is an example of what type of bug?

a) syntax error   b) runtime error – NameError   c) runtime error – IndexError

d) runtime error – TypeError   e) logic (semantic) error
16-17. Given the following Python code:

```python
nums = (1, 3, 4, 6, 8, 9, 11, 13, 15, 25, 99, 100, 102)
```

and a target integer, t:

16. Using a linear search algorithm, what is the maximum number of items in `nums` that would need to be searched to locate (or not locate) t? (Note: `len(nums) == 13`)

   a) 0   b) 1   c) 4   d) 7   e) 13

17. Using a binary search algorithm, what is the maximum number of items in `nums` that would need to be searched to locate (or not locate) t?

   a) 0   b) 1   c) 4   d) 7   e) 13

18-19. What will be the result of executing the following code:

```python
>>> parks = ['Crater Lake', 'Rainier', 'Olympia']
>>> parks.remove('Rainier')
>>> parks[0]
```

18. a) 'Crater Lake'   b) 'Rainier'   c) 'Olympia'   d) 'Glacier'   e) error

```python
>>> parks = parks.append('Glacier')
>>> parks[0]
```

19. a) 'Crater Lake'   b) 'Rainier'   c) 'Olympia'   d) 'Glacier'   e) error

20-23. Given the following Python Shell session:

```python
>>> alist = [1, 2, 3]
>>> id(alist)
4328102872
>>> alist[1] = 99
>>> id(alist)
?? - x
```

```python
>>> id(alist)
?? - y
```

```python
>>> astring = '1234'
>>> id(astring)
4391692304
>>> astring = astring + '5'
>>> id(astring)
?? - z
```

```python
>>> astring[1] = 'a'
?? - w
```
20. What would be printed at ?? - x?

a) [1, 2, 3]  
   b) [99, 2, 3]  
   c) [1, 99, 3]  
   d) [1, 2, 99]  
   e) error message

21. What would be printed at ?? - y?

a) 4328102872  
   b) 4391692304  
   c) 4388677768  
   d) <class 'list'>  
   e) error message

22. What could be printed at ?? - z?

a) 4328102872  
   b) 4391692304  
   c) 4389887760  
   d) <class 'str'>  
   e) error message

23. What would be printed at ?? - w?

a) 4328102872  
   b) 4391692304  
   c) 4389887760  
   d) <class 'str'>  
   e) error message

24. What is printed at checkpoint 1?

a) ['CIS 210', 'CIS 211', 'CIS 212']  
   b) ['CIS 210', 'CIS 211']  
   c) ['CIS 212']  
   d) 'CIS 212'  
   e) NameError
25. What is printed at checkpoint 2?

a) ['CIS 210', 'CIS 211', 'CIS 212']  
b) ['CIS 210', 'CIS 211']

c) ['CIS 212']

d) 'CIS 212'

e) NameError

26. What is printed at checkpoint 3?

a) ['CIS 210', 'CIS 211', 'CIS 212']

b) ['CIS 210', 'CIS 211']

c) ['CIS 212']

d) 'CIS 212'

e) NameError

27. What is printed at checkpoint 4?

a) ['CIS 210', 'CIS 211', 'CIS 212']

b) ['CIS 210', 'CIS 211']

c) ['CIS 212']

d) 'CIS 212'

e) NameError

28-30. Given the function, raindata, with one parameter, f, the name of a file containing rain data (daily rainfall in Eugene from March 1 through March 15), and a file, 'rainfall.csv', with one header line, and additional lines of comma separated values, each recording the year, month, day, and rainfall amount for that day:

```python
#rainfall data for final exam CIS 210 F17 Eugene, OR
2019,12,1,.2
2019,12,2,.24
2019,12,3,.16
```

```python
1 def raindata(f):
2     '''Exam function
3     '''
4     with open(f) as rainf:
5         rainf.readline()
6         track_rain = {}
7         for rain in rainf:
8             rain = rain.strip().split(',')
9             date = rain[0] + '-' + rain[1] + '-' + rain[2]
10            track_rain[date] = float(rain[3])
11        return track_rain

>>> raindata('rainfall.csv')
```

```text
>>> raindata('rainfall.csv')
```
28. The type contract for raindata should be
   a) (file object) -> dict  b) (file object) -> list
   c) (str) -> str  d) (str) -> dict  e) (str) -> list

29. The type of rain after line 10 is executed is
   a) file object  b) string  c) list  d) dictionary  e) float

30. The type of date after line 11 is executed is
   a) file object  b) string  c) list  d) dictionary  e) float

31. Given:

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

What will be the result of executing the following code:

```python
>>> best('CIS 210: Best wishes for a pleasant break.')
```

a) Best wishes for a pleasant break.

```python
def dtob(n):
    '''(int) -> str
    Convert non-negative int n to binary string.
    '''
    if n == 0:
        b = '0'
    else:
        b = ''
        next_n = n
        while next_n > 0:
            r = next_n % 2
            b = str(r) + b
            next_n = next_n // 2
        return b
```

```python
def btod(b):
    '''(str) -> int
    Convert binary string to decimal integer.
    '''
    dn = 0
    bctr = len(b) - 1
    for bit in b:
        dn += int(bit) * (2**bctr)
        bctr -= 1
    return dn
```
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. Some documentation and keywords are given for each function; comments indicate where to fill in the rest of the code.

32. Complete the implementation of `strReverse`, which will take a single string argument, s, and return a string which is the reverse of s. (You may NOT use `[::-1]`.)

```python
def strReverse(s):
    #complete the function header
    #type contract needed
    Reverse s; return the reversed string.

    >>> strReverse('abc')  #basic example is given
    'cba'
    >>>  #edge (boundary) example needed

    #code to solve the problem goes here

    return  #complete the return statement```
33. Complete the implementation of `euclidD`, a function to compute and return the Euclidean distance between two points, p1 and p2, which represent as x, y coordinates on a grid. Recall that the formula for Euclidean distance between two points is

\[ d_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \]

```python
import math
def euclidD((x1, y1), (x2, y2)) -> float:
    # Brief description is needed
    Computes and returns the Euclidean distance between points (x, y) p1 and p2.

>>> # Basic example is needed

>>> # Boundary example is needed

... # Code to solve the problem goes here

return
```

# Complete the return statement