concepts: Python variables, literals, assignment, expressions
1-2. Given the Python statement

greeting = 'Hello, CIS 210.'

1. [1 pt.] greeting is a Python
   a. variable       b. algorithm       c. operator       d. function

2. [1 pt.] 'Hello, CIS 210.' is a(n)
   a. expression       b. string literal       c. neither of these       d. both of these

concepts: using Python standard modules
3. [1 pt.] To use the sqrt() function, you must first use the following Python statement:
   a. use math       b. import math       c. math::sqrt()       d. return sqrt()

concepts: Python string data type, operations; assignment; expression evaluation
4. [2 pts.] Given the following Python code:

```python
1. >>> name = 'Roy G Biv'
2. >>> i = name[1]
3. >>> i
4. >>> m = name[4]
5. >>> m
6. >>> name[-1]
```

What is printed after lines 3, 5, and 6?
   a. 'R', 'G', 'v'
   b. 'R', '', 'v'
   c. 'o', 'G', 'i'
   d. 'R', 'G', 'v'

concepts: Python for loop
5-6. Given the following Python code:

```python
for i in range(10):
    print('check')
```

5. [1 pt.] How many times will check be printed?
   a. 0       b. 1       c. 9       d. 10

6. [1 pt.] What will be the value of i the last time the loop is executed?
   a. 0       b. 1       c. 9       d. 10
concepts: reading Python code; conditionals
7. [2 pts.] We've heard from the client who requested the federal income tax estimator. Recall that the estimated tax is calculated as taxable income (gross income minus the sum of the standard deduction and the exemption amount * number of exemptions) multiplied by the tax rate:

```python
exemption = 4050
std_deduction = 10000
tax_rate = .20

taxable_income = gross_income - std_deduction
exempt_adjust = exemption * number_exempt
taxable_income = taxable_income - exempt_adjust
estimated_tax = taxable_income * tax_rate
```

They forgot to mention that if the gross income is less than the sum of the standard deduction and exemption adjustment, the function should return an estimated tax of zero (not a negative number, as it currently does).

To implement this new functionality, we will need to use

a. math module       b. random module       c. conditional statement       d. for loop

concepts: Python user-defined functions, data types, return values, iterative process
8-10. Given the following Python code:

```python
def mysqrt(n, k):
    '''(integer, integer) -> ??-8

    Generates k successive, better approximate square roots of n, a positive integer.

    The approximate square root is ??-9.

    >>> mysqrt(25, 5)
    5.000023178253949
    '''
    approx_val = 1
    for ctr in range(k):
        approx_val = .5 * (approx_val + n/approx_val)
    return approx_val
```

8. [1 pt.]
a. integer       b. float       c. string       d. None

9. [1 pt.]
a. returned       b. printed       c. returned and printed       d. neither returned nor printed

10. [1 pt.]
a. iterative       b. Monte Carlo       c. cipher       d. infinite loop
concepts: scope, function returned values and side effects

11-12. Given the following Python code:

```python
def qx1(a):
    a += 1
    return a

def qx2(a):
    a += 1
    print(a)
    return 100

def qx3(x):
    result = qx1(x) + qx2(x)
    return result
```

11. [2 pts.] Which is correct?

<table>
<thead>
<tr>
<th>a.</th>
<th>b.</th>
<th>c.</th>
<th>d.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>a = 5</code></td>
<td><code>a = 5</code></td>
<td><code>a = 5</code></td>
<td><code>a = 5</code></td>
</tr>
<tr>
<td><code>qx1(a)</code></td>
<td><code>qx1(a)</code></td>
<td><code>qx1(a)</code></td>
<td><code>qx1(a)</code></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td><code>a</code></td>
<td><code>a</code></td>
<td><code>a</code></td>
<td><code>a</code></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

12. [2 pts.] Which is correct?

<table>
<thead>
<tr>
<th>a.</th>
<th>b.</th>
<th>c.</th>
<th>d.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>qx3(0)</code></td>
<td><code>qx3(0)</code></td>
<td><code>qx3(0)</code></td>
<td><code>qx3(0)</code></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>101</td>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>

concepts: accumulator pattern, for loop, conditional, string data type

13. [2 pts.] Given the following Python code:

```python
def q13(s1):
    s2 = ''
    for ch in s1:
        if ch not in s2:
            s2 += ch
    return s2
```

Which is correct?

<table>
<thead>
<tr>
<th>a.</th>
<th>b.</th>
<th>c.</th>
<th>d.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>q13('123456')</code></td>
<td><code>q13('123456')</code></td>
<td><code>q13('123456')</code></td>
<td><code>q13('123456')</code></td>
</tr>
<tr>
<td>'123456'</td>
<td>'1'</td>
<td><code>q13('999999')</code></td>
<td><code>q13('999999')</code></td>
</tr>
<tr>
<td><code>q13('999999')</code></td>
<td><code>q13('999999')</code></td>
<td><code>q13('999999')</code></td>
<td><code>q13('999999')</code></td>
</tr>
<tr>
<td>'9'</td>
<td>'9'</td>
<td>'9'</td>
<td>'9'</td>
</tr>
</tbody>
</table>