CIS 210 Fall 2017
Midterm 1

Concept: Python objects have types.
What is the result when the following Python code is executed:

1. `>>> type(99.9)`

a) `<class 'int'>` **b) `<class 'float'>`** c) `<class 'bool'>`
d) `<class 'str'>` e) `<class 'python'>`

2. `>>> type(True)`

a) `<class 'int'>` b) `<class 'float'>` **c) `<class 'bool'>`**
d) `<class 'str'>` e) `<class 'python'>`

3. `>>> type('false')`

a) `<class 'int'>` b) `<class 'float'>` c) `<class 'bool'>`
d) **`<class 'str'>`** e) `<class 'python'>`

4. `>>> type(len('hello') == len('goodbye'))`

a) `<class 'int'>` b) `<class 'float'>` c) **`<class 'bool'>`**
d) `<class 'str'>` e) `<class 'python'>`

Concept: Objects are stored in memory.
5. Given the following:

```
>>> id(99.9)
4298470336
```

`4298470336` refers to a

a) Python type b) Python built-in function **c) memory location** d) `None` type

Concept: Python assignment associates a name with a Python object (the value of the expression on the rhs of the = assignment operator. Assignment is not an expression/does not return a value.
6. `>>> a = 1200` is an example of a Python

a) expression **b) assignment statement** c) conditional d) loop

Concept: Objects are combined in expressions. Expressions are evaluated and return a value.
7. `>>> len('CIS 210')` is an example of a Python

a) expression b) assignment statement c) conditional d) loop

Concepts: Operators have an order of operation. Objects have types. Reading sequential code.
8. Given the following:

```
1 - >>> ftemp = 212
2 - >>> ctemp = (ftemp - 32) * 5/9
3 - >>> ctemp = ftemp - 32 * 5/9
```

The *value* of `ctemp` will **[??]** from line 2 to line 3; The *type* of `ctemp` will **[??]** from line 2 to line 3

a) stay the same/change **b) change/stay the same** c) stay the same/stay the same
d) change/change

Concept: Assignment associates variable names with objects (only).
9. Given the following Python code, what will be printed in the Shell:

```
>>> b = 20
>>> a = b + 1
>>> b = 30
>>> a
```

a) `20` **b) `21`** c) `30` d) `31` e) nothing will be printed

Concepts: Python for loop, accumulator pattern; dynamic typing can lead to logic errors.
10-11. What will be printed when the following Python code is executed?

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

print(yellow_ct)
```

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

11. The code in question 10 does not work as intended. This is due to Python's

a) static typing **b) dynamic typing** c) strong typing d) weak typing

Concepts: reading/executing code: for loop, tracking updates to variables.
12. What are the values of `a` and `b` after the following Python code executes:

```
a = 10
b = 3
t = 0
for i in range(1, 4):
    t = a
    a = i + b
    b = t - 1
```

a) `10, 3` b) `11, 3` **c) `6, 10`** d) `10, 11` e) `3, 11`

Concept: accumulator pattern.
13. Order the lines of Python code to implement an accumulator pattern (ignore lack of indents):

```
1 - p = p * i
2 - p = 1
3 - for i in range(10):
```

a) 1, 2, 3 **b) 2, 3, 1** c) 3, 1, 2 d) 3, 2, 1 e) 2, 1, 3

Concepts: calling a function; parameter passing; functions return values; return statement; indefinite iteration.
14. What value is returned when the following Python code is executed:

```
def qx(n):
    '''(integer) -> ??

    Test function.
    '''
    ctr = 0
    while n > 1:
        n = n // 2
        ctr += 1

    return 'The end.'

qx(7)
```

a) `1` b) `2` c) `3` **d) `'The end.'`** e) `None`

Concepts: calling a function; parameter passing; functions return values; return statement; indefinite iteration.
15. What value is returned when the following Python code is executed:

```
def qx(n):
    '''(integer) -> ??

    Test function.
    '''
    ctr = 0
    while n > 1:
        n = n // 2
        ctr += 1

    return ctr

qx(7)
```

a) `1` **b) `2`** c) `3` d) `'The end.'` e) `None`

Concepts: type contract; returning a Boolean value; lazy ("short circuit") evaluation.
16-18. Given the following Python code:

```
def q16(age, salary):
    '''
    (number, number) -> ??

    Test function.

    >>> q16(18, 5000)
    ??
    '''
    return (age < 18) and (salary < 10000)
```

16. Complete the type contract:

a. number b. integer c. float **d. Boolean**

17. What value is returned when `q16(18, 5000)` is executed?

a. `18` b. `5000` c. `5018` d. `True` **e. `False`**

18. To determine this value, Python evaluated

a.	**b.**	c.
`(age < 18)`	**`(age < 18)`**	`(salary < 10000)`
`(salary < 10000)`		

Concepts: implementing an algorithm, type contract, accumulator pattern, for loop.
19-22. Given the following Python code:

```
def mysqrt(n, k):
    '''(integer, ??-19) -> ??-19

    Generates an approximate square root of n,
    a positive number, via an iterative process
    that runs k times.

    The approximate square root is returned.

    >>> mysqrt(25, 5)
    5.0
    '''

    approx_val = 1
    for ctr in range(k):
        approx_val = .5 * (approx_val + n/approx_val)

    return round(approx_val, 2)

mysqrt(25, 5)
```

19. Complete the type contract:

a. bool/float b. float/None c. integer/integer **d. integer/float**

20. The first time the `for` loop executes, the value of `k` is
a. `0` b. `1` c. `4` **d. `5`** e. `k` is not defined

21. The first time the `for` loop executes, the value of `ctr` is
a. `0` b. `1` c. `4` d. `5` e. `ctr` is not defined

22. After the for loop has finished executing, the value of `n` is
a. `0` b. `1` c. `4` **d. `25`** e. `n` is not defined

Concepts: Python namespaces; variable scope.
23-26. Given the following Python code:

```
1 - def twice(x):
2 -   '''(int) -> int

3 -   Return x multiplied by 2.

4 -   >>> twice(3)
5 -   6
6 -   '''
7 -   y = 2
8 -   result = y * x
9 -   return result
```

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

`>>> y = 5` then

23. `>>> twice(y)`

a. `0` b. `2` c. `5` **d. `10`** e. `NameError`

24. `>>> y`

a. `0` b. `2` **c. `5`** d. `10` e. `NameError`

25. `>>> x`

a. `0` b. `2` c. `5` d. `10` **e. `NameError`**

26. When the following Python code is executed

```
>>> z = 10
>>> twice(z)
```

what is the value of `x` at line 8 of the `twice` function?
a. `0` b. `2` c. `5` **d. `10`** e. `error`

Concepts: Using Python standard library; docstring; turtle graphics.
27-28. You are given the following Python code:

```
from turtle import *

def square():
    '''() -> None

    Use Python turtle graphics to
    draw a square.

    >>> square()
    [draw square on turtle Canvas]
    '''
    for i in range(4):
        fd(100)
        lt(90)

    return None
```

with the assignment to make changes so the function can draw any polygon shape.
Some progress has already been made; finish the work:

```
def poly(s):
    '''(int) -> None

    Use Python turtle graphics to
    draw an s-sided polygon.

    >>> ??-27
    [draw square on turtle Canvas]
    '''
    for i in range(??-28):
        fd(100)
        lt(360 / s)

    return None
```

27. a. `poly` **c. `poly(4)`** b. `poly()` d. `poly(s)`

28. **b. `s`** a. `4` c. `ctr` d. `10`

Concepts: type contract; understanding Python code/underlying algorithm.
29-30. Given the following Python code:

```
def q29(s):
    '''
    (??) -> ??

    Test function.

    >>> q29('The quick brown fox')
    5
    >>> q29('Hello, world.')
    3
    '''
    vowels = 'aeiou'
    result = 0
    for i in range(len(s)):
        if s[i] in vowels:
            result += 1

    return result
```

29. Complete the type contract:

a) str/str b) int/int c) str/float
d) str/bool **e) str/int**

30. Executing this function will

a) Return count of vowels in s.

b) Return count of characters in s.

c) Print count of vowels in s.

d) Print count of characters in s.

e) Causes an infinite loop.