Quiz 1 Practice

ANSWER KEY

- Signing signifies you are aware of and in accordance with the **Academic Honor Code of Georgia Tech**.
- Calculators and cell phones are NOT allowed.
- This is a Python programming test. Where asked for Python statements or expressions you must print them exactly as they would be typed in a Python source file or interactive shell.

Question	Points per Page	Points Lost	Points Earned	Graded By
Page 1	4	-	=	
Page 2	13	-	=	
Page 3	12	-	=	
Page 4	10	-	=	
Page 5	8	-	=	
Page 6	10	-	=	
Page 7	0	-	=	
Page 8	0	-	=	
TOTAL	57	-	=	

1. **True or False**

In each of the blanks below, write "T" if the statement beside the blank is true, "F" otherwise.

[1] (a) **T** Every Python value has a type such as `float` or `int`.

[1] (b) **F** Python variables are statically typed, meaning that once you assign a value to a variable you can only assign new values of the same type. For example, after `x = 3.14` you can only assign `float` values to `x`.

[1] (c) **F** The `+` operator means the same for `str` values as it does for `int` values.

[1] (d) **F** `try = try + 1 # increment the number of tries` is a valid Python statement.

 Points available: 4 - points lost: _____ = points earned: _____. Graded by: _____

2. **Expression Evaluation**

For each expression below, write the value and then the Python data type of the evaluated legal expression in the space provided.

Expression: `7 / 2`

[1] (a) Calculated value: 3.5

[1] (b) Type: `float`

Expression: `64 - 16 * 2`

[1] (c) Calculated value: 32

[1] (d) Type: `int`

Expression: `'Ni' * 3`

[1] (e) Calculated value: 'NiNiNi'

[1] (f) Type: `str`

Expression: `1 // 2`

[1] (g) Calculated value: 0

[1] (h) Type: `int`

Expression: `True and (1 == 2)`

[1] (i) Calculated value: `False`

[1] (j) Type: `bool`

Expression: `1 if 2 else 3`

[2] (k) Calculated value: `1`

[1] (l) Type: `int`

3. **Multiple Choice** Circle the letter of the correct choice.

[2] (a) Given the following code:

```
capitals = {}
capitals['Murica'] = 'Warshington'
capitals['Germany'] = 'Bonn'
capitals['France'] = 'Paris'
capitals['Engalnd'] = 'London'
capitals['Germany'] = 'Berlin'
```

What is `capitals['Germany']`?

A. `'Berlin'`

B. `'Sweden'`

C. `'Paris'`

D. `'London'`

[2] (b) What is wrong with this code:

```
n = 5
while n > 0:
    print(n)
n -= 1
```

A. The variable `n` is declared outside the scope of the `while` loop.

B. The `while` loop never finishes.

C. The variable `n` is the wrong type.

D. There is nothing wrong with this code.

[4] (c) What is the value of `list(enumerate(['today', 'is', 'a', 'great', 'day']))`?

A. `[('today', 1), ('is', 2), ('a', 3), ('great', 4), ('day', 5)]`

B. `[todayisagreatday]`

C. `[(0, 'today'), (1, 'is'), (2, 'a'), (3, 'great'), (4, 'day')]`

D. the location of an iterable object

4. **Tracing**

Consider the following program:

```
x = 1
y = x
y = 2
print(x)
print(x == y)
```

[5] (a) What is printed when this program is run from the command line?

```
1
False
```

Consider the following program:

```
counter = 0;

def incrementCounter():
    global counter
    counter += 1
    return True


a = True
b = False;
if b or incrementCounter():
    print("Boo")
if (a or b) and incrementCounter():
    print("ya!")
print(counter)
```

[5] (b) What is printed when this program is run from the command line?

```
Boo
ya!
2
```

5. **Short Answer**

[2] (a) What is the value of "abcdefg"[1]

Solution:
`'a'`

[2] (b) What is the value of "abcdefg"[-1]

Solution:
`'g'`

[2] (c) Write an expression that computes the average of a list of numbers `nums`.

Solution:
```
sum(nums) / len(nums)
```

[2] (d) Make the dictionary variable, `e2f`, that contains mappings from English words to their French equivalents. Use these words: dog is chien, cat is chat, and walrus is morse.

Solution:
```
e2f = {'dog': 'chien', 'cat': 'chat', 'walrus': 'morse'}
```

6. **Complete the Method**

[5] (a) Fill in the code for the following method that takes a list of numbers and returns the number of even numbers in list argument. Your code should use a `for` statement.

```
def evens(nums):
```

Solution:

```
    count = 0
    for num in nums:
        if num % 2 == 0:
            count += 1
    return count
```

[5] (b) Fill in the code for the following method that takes a list of numbers and a number and returns `True` if the list contains the number, `False` otherwise. You will need a loop, and your loop must not execute more iterations than necessary, and you cannot use `break` or `continue` or the `in` operator.

```
def contains(nums, n):
    // Your code goes here
```

Solution:

```
    found = False
    i = 0
    while i < len(nums) and not found:
        if nums[i] == n:
            found = True
        i += 1
    return found
```

 Points available: 10 - points lost: _____ = points earned: _____. Graded by: _____