CS 2316 Exam 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	0	-	=	
Page 2	0	-	=	
Page 3	0	-	=	
Page 4	0	-	=	
Page 5	0	-	=	
Page 6	0	-	=	
Page 7	0	-	=	
Page 8	0	-	=	
TOTAL	72	-	=	

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: 0 - 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. Be exact.

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`

 Points available: 0 - points lost: _____ = points earned: _____. Graded by: _____

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 `len(set(['A', 'b', 'b', 'a']))`

A. 2
B. 3
C. 4
D. 0

[2] (c) 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.

[2] (d) What's the value of the expression `''.join('h a n d s'.split())`

A. `'hands'`
B. `'h a n d s'`
C. `['h', 'a', 'n', 'd', 's']`
D. `None`

 Points available: 0 - points lost: _____ = points earned: _____. Graded by: _____

4. **Tracing**

Consider the following program:

```
counter = 0;

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

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

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

Solution: Boo
ya!
2

Consider the following program:

```
mystery = "mnerigpaba"
solved = ""
for i in range(len(mystery) // 2):
    j = -i - 1
    solved += mystery[i] + mystery[j]
print(solved)
```

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

Solution: manbearpig

 Points available: 0 - points lost: _____ = points earned: _____. Graded by: _____

5. **Short Answer**

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

Solution:
`'gfedcba'`

[2] (b) Write a list comprehension that returns a list of the first 5 squares where the first square is 1.

Solution:
```
[x * x for x in range(1, 6)]
```

[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'}
```

[2] (e) Write a dictionary comprehension that converts `e2f` to a dictionary from French words to their english equivalents and assigns this new dictionary to a variable `f2e`

Solution:
```
f2e = {f: e for e, f in e2f.items()}
```

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: 0 - points lost: _____ = points earned: _____. Graded by: _____

7. **Write the method.** Assume valid input.

[10] (a) Given a $m \times n$ matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

The transpose $\mathbf{A}^T$ is defined as: $\left[\mathbf{A}^T\right]_{ji} = [\mathbf{A}]_{ij}$. Think "the rows of a matrix are the columns of its transpose." One way to represent matrices in Python is as a list of lists, for example:

```
m = [
    [1, 2, 3],
    [4, 5, 6]
]
```

Write a method `transpose` that takes a single parameter `m` representing a 2-dimensional matrix as a list of lists and returns its transpose as a list of lists. Hint: it's possible to do this in one line, but you may use `for` statements instead.

```
def transpose(m):
    mt = [[0] * len(m) for column in m[0]]
    for row in range(len(m)):
        for column in range(len(m[row])):
            mt[column][row] = m[row][column]
    return mt

def transpose2(m):
    return [[row[i] for row in m] for i in range(len(m[0]))]

def transpose3(m):
    return [list(row) for row in zip(*m)]
```

 Points available: 0 - points lost: _____ = points earned: _____. Graded by: _____

[5] (a) Write a class `Person` with three instance variables: `name`, `age`, and `email` and two methods:

- `is_senior()`, which returns `True` if the `Person` instance's `age` is greater than 59, and
- `user_name()`, which returns the user name portion of the instance's `email`, that is, the part before the `@` symbol.

```
class Person():
    def __init__(self, name, age, email):
        self.name = name
        self.age = age
        self.email = email

    def is_senior(self):
        return self.age > 59

    def user_name(self):
        return self.email.split('@')[0]
```

[5] (b) Write function, `oldest`, that takes a variable number of `Person` (from previous question) parameters (that is, a variable number of single `Person` objects) and returns the oldest `Person` among the arguments. Assume `oldest` is always called with at least one argument.

```
def oldest(*persons):
    return sorted(persons, key=lambda p: p.age)[-1]
```

 Points available: 0 - points lost: _____ = points earned: _____. Graded by: _____