Name:_____

CSCI-UA.0002-008 – Midterm Exam #1

October 17th, 2016

Instructor: Joseph Versoza

Ask the person to your left for their first name
(leave blank if next to empty seat or wall):

_____

Ask the person to your right for their first name
(leave blank if next to empty seat or wall):

_____

Keep this test booklet closed until the class is prompted to begin the exam

- Computers, calculators, phones, textbooks or notebooks are **not allowed** during the exam
- Please turn off your phone to avoid disrupting others during the exam
- The back of this cover can be used as scratch/scrap paper

1. Write the result, `True` or `False` (or error if applicable) for the following boolean expressions and statements. (3 points)

a) `42 < '42'` **error**

b) `'draw' < 'drip'` **True**

c) `42 != '42'` **True**

d) `False or 'hello world'` **True / hello world**

e) `5 // -3 == -2` **True**

f) `False or True and Not True` **False / error**

2. Read the code in the first column. Answer questions about the code in the second and third columns. (6 points total)

Code	Question #1	Question #2

`Count = 1 // = 0`
 	
`while count < 5: // <= 5`
 	
`    if count != 3:`
 	
`        print(count)`
`        count += 1   // un-indent`
	What is the output of the code on the left? (1 point)
	
	1
	2
	(repeatedly no output)
		Change/fix the program (you can do this directly in the code in left-most column) so that the output is the same as the following (**do not use** a for loop, and do not use multiple consecutive if statements): (2 points)
		0
		1
		2
		4
		5

`result = 1`
`for num in range(11, 4, -3):`
`    if num % 2 == 0:`
`        result += 1`
`    else:`
`        result += num`
`print(result)`
	How many times will this loop run? (1 point)
	
	3
		What is the output of this program? Show your calculations / work for partial credit. (2 points)
		
		18

3. Your friend is part of an avant-garde acapella group, and they've written a program to write the lyrics to their next song. The song's lyrics consists of numbers, "mmmm" and "bzzzz" (um, what? Art!). Your friend's program is supposed to:

a) **print out** numbers from **50** down to (and including) **0**, by **5**'s...
b) after each number, **add** a **random number** (1-5) of **exclamation** points
c) if the number is **greater** than or **equal** to **40**, always print out **bzzzz** (instead of the number and instead of mmmm)
d) however, for the remaining numbers, if the **number ends** in a **0**, print out **mmmm** instead of the number

Unfortunately, their program (shown below) is full of errors. It does not produce the expected output! **Circle 3 errors** (there are more than 3), **identify** if they're a syntax, runtime or logical error... and **briefly explain** why. Draw arrows or label with numbers to associate error with explanation (6 points)

Expected Output

```
bzzzz
bzzzz
bzzzz
35!!!!!
mmmm
25!!!!!
mmmm
15!!!
mmmm
5!
mmmm
```

Broken Code (should produce output on left, but does not!)

```
for i in range(50, 0, -5): 1
    if i % 10 == 0: 2
        print("mmmm")
    else if i => 40: 3, 4
        print("bzzzz")
    else:
        num = random.randint(0, 5) 5
        print(i + num * '!') 6
```

Error #	Type	Explanation
1	**Logical**	**Should be range(50, -1) to include 0**
2	**Logical**	**Switch condition with i >= 40 (>= 40 takes priority for bzzzz)**
3	**Syntax**	**else if should be elif**
4	**Syntax**	**=> should be >=**
5	**Logical**	**Use randint to generate 1 to 5 (not 0 to 5)**
6	**Runtime**	**TypeError: adding non-string (i) to string**

4. Circle the boolean expression that is equivalent to: **`x <= 10 and y <= 10`** ... when the given the following values for x and y: `(9, 7)` and `(10, 12)`. (1 point)

(a) `x > 10 or y > 10`

(b) `x > 10 and y > 10`

(c) `not x > 10 or not y > 10`

(d) `not (y > 10 or x > 10)`

5. Convert the following numbers . Show work for partial credit. (2 points)

a) 01000011 is ___67_____ in decimal. b) 18 is __0__0__0__1__0__0__1__0__ in binary.

6. Circle all of the **valid variable names** (1 point): $foo **Foo** **_foo** 2_foo **foo2**

7. Answer the following questions about loops. (3 points)

   a) In Python, **define count-controlled loop**. What is the construct/control structure (keyword) that represents it?

   A loop that repeats a specific number of times... a for loop in Python

   b) **Define condition-controlled loop**. What is the construct/control structure (keyword) that represents it?

   A loop that repeats as long as a condition is true... a while loop in Python

   c) Briefly explain why you would use one kind of loop over the other?

   Use a for loop when you know how many iterations you want; use a while loop when the number of iterations is based on a condition (number of iterations is not known beforehand)

8. What is the output of the following code (no output and error are possible)? Note the number of spaces if there is left or right padding. (3 points)

   a) `print(format('hello', '.2f'))` **__error_____**

   b) `print(format('hello', '<8s') + format(42, '.1f'))` **__hello___42.0_____(3 spaces)_____**

   c) `print(format(0.90, '.2%'))` **__90.00%_____**

9. In the **truth table** below, fill out all of the possible Boolean values for **p** and **q**, as well as the result of **p and not q**. That is, under columns p and q, write out all possible combinations of True and False... and in the third column write out the result of **p and not q** using the boolean values under the columns p and q in that row. (2 points)

p	**q**	**p and not q**
True	**True**	**False**
True	**False**	**True**
False	**True**	**False**
False	**False**	**False**

10. Determining what the following program will print out based on the user input specified in the 1st column of the table below. **Show your work for partial credit**. (4 points)

```
n = int(input('Gimmeh a number!\n> '))
if n == 1 or n == 2:
    print(n - 1)
elif n > 0:
    prev = 0
    cur = 1
    for i in range(0, n - 2):
        cur, prev = (prev + cur), cur
    print(cur)
else:
    print('Invalid Input')
```

User Input	**Resulting Output to Screen**
2	**1**
-2	**Invalid Input**
5	**123**

11. Name two data **types** in Python that **are not** *numeric*, and give a **syntactically correct** literal example of each. (2 points)

- type: **str, bool** example: **'hello', True**
- type: **range** example: **range(5)**

12. Write a program that calculates the factorial of a number given by the user. The factorial of a number is the product all positive integers less than or equal to the number. Factorial is indicated by a number followed by an exclamation point. For example, the factorial of 4 is: `4! = 4 x 3 x 2 x 1 = 24`. (6 points)

a) Ask for a number to calculate the factorial of (`'Enter a number to calculate the factorial of'`)
b) You can assume that the number the user enters is a whole number (you don't have to worry about letters or floating point)
c) However, if the number is less than 0, print out the following message: `Number must be >= 0`
d) Otherwise, calculate the factorial of the number entered and print it out as: `n! is result`
   - the factorial of 0 is 1
   - the factorial of numbers greater than 0 is the product of all of the positive numbers less than or equal to that number
e) The question only needs to be asked once (there is no repetition required)
f) Example output below:

```
# Example run 1 (product of positive #'s):
Enter a number to calculate the factorial of
> 4
4! is 24

# Example run 2 (factorial of 1 is 1):
Enter a number to calculate the factorial of
> 1
1! is 1

# Example run 3 (factorial of 0 is also 1):
Enter a number to calculate the factorial of
> 0
0! is 1

# Example run 4 (negative number):
Enter a number to calculate the factorial of
> -4
Number must be >= 0
```

```
num = int(input('Enter a number to calculate the factorial of\n> '))
product = 1
if num < 0:
    print('Number must be >= 0')
elif num >= 0:
    for i in range(1, num + 1):
        product *= i
    print(product)
```

13. Write a tiny betting game (wait a second, is this even legal!?). The player will **choose** either **(L)ower** or **(H)igher**, and place a bet. The computer will **generate** a **random number** between **1 and 7** inclusive. If the resulting **number matches** the player's **choice relative** to **4** (that is, lower or higher than 4), you keep your bet, and you **win the same amount you bet**. However, if it's the **opposite**, you **lose** the **amount** you **bet**. Finally, if it's a **tie**, you **don't lose anything**. Do this until you have no longer have any money to bet (you must have at least $1 to bet)... or until you've doubled your money (you start with $100). (8 points)

a) Start the player with **$100**
b) Ask the player to choose L or H for lower or higher (assume input is always valid): '`(L)ower or (H)igher than 4?`'
c) Ask the player how much they'd like to bet (allow floating point numbers); if they don't enter a positive number, **default to 1**
d) Generate a random number between 1 and 7 (inclusive) and print it out
e) If the player guesses correctly, print out '`You won`', add the player's bet to their total
f) If the opposite occurs, print out '`You lost!`', and subtract the player's bet from their money
g) If it's a tie, **no money** is added or deducted from the player
h) Print out the new total on the same line as e/f: '`Total <total>`'
i) **All dollar amounts should have two decimal places and a dollar sign**: `$2.00`
j) **Repeat again, starting with step b** ... until the player's money is less than 1 or greater than or equal to $200
k) Once the game is finished, print out '`Game ended.`', and the player's total (negative amounts are ok)
l) Example output below:

```
(L)ower or (H)igher than 4?
> L
How much are you betting?
> 50
1
You won! Total: $150.00
(L)ower or (H)igher than 4?
> L
How much are you betting?
> 53
7
You lost! Total: $97.00
```

```
<continued from left collumn>
(L)ower or (H)igher than 4?
> L
How much are you betting?
> 105
2
You won! Total: $202.00
Game ended. You have $202.00
```

```
+0.5: import random
+1:   while loop and condition
+1:   ask for inputs (choice and bet)
+0.5: generate random number correctly
+0.5: print the roll
+1:   condition for loss
+1:   condition for win
+0.5: print you won or you lose
+1:   increase or decrease money
+0.5: format money
+0.5: game over print
+0.5: total after every round
```

```
import random
money = 100
while money > 0 and money <= 200:
    choice = input('(L)ower or (H)igher than 4?\n> ')
    bet = int(input('How much are you betting?\n> '))
    roll = random.randint(1, 7)
    print(roll)
    if roll == 4:
        print('no one won!')
    elif choice == 'L' and roll < 4 or choice == 'H' and roll > 4:
        print('You won!')
        money += bet
    elif choice == 'L' and roll > 4 or choice == 'H' and roll < 4:
        print('You lost!')
        money -= bet
    print('Total ', '$' + format(money, '.2f'))
print('Game ended. You have ', '$' + format(money, '.2f'))
```

14. You're a mad scientist, and one of your hobbies is creating mutants. To assist in your creation of magnificent mutant minions, you write a program to generate DNA sequences. DNA is made up of triplets (codons) of nucleotides: guanine, adenine, cytosine and thymine (G, A, C and T). Each triplet, or codon, is made up of 3 nucleotides (nucleic acids). Generate a random sequence of G, A, C, and T based on a number of codons specified by the user. (6 points)

    Continually ask the user for the number of codons if they do not specify a number greater than 0:

```
How many nucleotide triplets (codons)?
> -1
Please enter a number greater than 0...
How many nucleotide triplets (codons)?
> 7
Your DNA is ready! ATCAATGGAATCGTGATATAC
```

```
0.5: import random
0.5: ask for input, convert to int
1:   continually ask for valid input
1:   sting accumulator
0.5: generate random number
2:   appropriate if/elif, along with addition
0.5: print
```

```
import random
triplets = int(input('How many nucleotide triplets (codons)?\n> '))
while triplets < 0:
    triplets = int(input('Please enter a positive number\n> '))

dna = ''
for i in range(triplets * 3):
    gact = random.randint(1, 4)
    if gact == 1:
        dna += 'G'
    elif gact == 2:
        dna += 'A'
    elif gact == 3:
        dna += 'C'
    elif gact == 4:
        dna += 'T'
print('Your DNA is ready! ' + dna)
```