

COGS 18 Exam #2

Fill out your Name and PID here:

Name: _____

PID: _____

Do not begin until instructed by Professor Ellis to do so.

Exam Notes:

- Put your PID at the top of each page.
- This is a closed book test. You may not use any resources other than your own brain and your writing utensil.
- All work should be your own. Keep your eyes on your own exam.
- If you are unsure of what any instructions means, raise your hand to ask a TA, IA or the Professor.
- You'll have until 9:50 to complete the exam.
- Answer all questions.
- For any questions that would not execute/would return an error, your answer can be: "Error"
- Your exam should have 5 pages.
- There are 60 possible points.

Part I: Variables & Operators (6 pts)

Q1. *Variables* - In each line of code below, what type of variable would be stored in my_variable (3 pts)?

my_variable = 17.6	_____ float _____
my_variable = True	_____ bool or boolean _____
my_variable = 'name'	_____ str or string _____
my_variable = [1, 2, 3]	_____ list _____
my_variable = (1, 2, 3)	_____ tuple _____
my_variable = 'None'	_____ str or string _____

Q2. *Operators* - Write out how each expression will evaluate. (3 pts):

6 / (2 + 4)	_____ 1 or 1.0 _____
18 % 9	_____ 0 _____
2 = 2	_____ Error _____
(22 < 5) or (20 == 20)	_____ True _____
False and not False	_____ False _____
'COG' in 'COGS 18'	_____ True _____

Part II: Indexing (4 pts)

Q3. Given the list below, write the line of code you would use to index the list and return the specified output (4 pts):

```
cogs18 = ['Devendra', 'Shreenivas', 'Andrew', 'Chau', 'Duolan',
          'Byungkwon', 'Severine', 'Stephen', 'Zekria']
```

For output A, use **negative indexing**: 'Zekria'

```
cogs18[ -1 ]
```

For Output B, use **positive (forward) indexing**: ['Duolan', 'Byungkwon', 'Severine']

```
cogs18[ 4:7 ]
```

Part III: Control Flow - Conditionals & Loops (4 pts)

Q4. Write the code for a **for loop** that **loops through a list** (assume my_list has already been defined). This loop must **contain** an **if** statement that checks if the value in the list is positive, an **elif** that checks if the value is negative, and an **else** statement. The code within each conditional should just be the word: **pass**. (4 pts)

```
for value in my_list:
    if value > 0:
        pass
    elif value < 0:
        pass
    else:
        pass
```

- **1 pt: colons and indentation correct**
- **1 pt: if, elif, and else used**
- **1 pt: value specified to loop over used in conditionals**
- **1 pt: conditions specified correctly**

PID:

Part IV: Functions (27 pts)

Q5. Assuming the following function has been defined, what would each of the following statements return? : (4 pts)

```
def subtraction(num1, num2):  
    return num1 - num2
```

```
subtraction(num1 = 19, num2 = 17) : _____ 2 _____  
subtraction(19, 17) : _____ 2 _____  
subtraction(num2 = 3, num1 = 6) : _____ 3 _____  
subtraction(num1 = 19, 17) : _____ Error _____
```

Q6A. In real code, write a function called `state_country` that takes one input parameter, `country` (we'll assume the input is a string). Set the default value for `country` to be the string of your home country. Within the function, concatenate 'I am from ' with the input parameter. Store this in the variable `output` and return `output` from this function. (5 pts)

```
def state_country(country = 'USA'):  
    output = 'I am from ' + country  
    return output
```

- 1 pt: function setup correctly (def, function name, colon, indent)
- 0.5 pt: includes country parameter
- 1 pt: default for input parameter correct
- 1 pt: string concatenation correct
- 0.5 pt: string stored in output
- 0.5 pt: return correct
- 0.5 pt: would execute

Q6B. If your `state_country()` function were defined (meaning the code part A of this question was executed), what would `state_country()` return? (1 pt)

'I am from USA' (Note: USA would be replaced with whichever country they specified as their default value)

Q6C. Assume the `state_country()` function has been defined. *In real code*, how would you use (meaning call or execute) your `state_country()` function so that it would return 'I am from outer space'? (3 pts)

```
state_country(country = 'outer space')  
OR  
state_country('outer space')
```

- 1 pt: function called correctly
- 1pt: input value is a string
- 1pt: input value passed correctly

Q7. There is a theoretical Python module called `unicorn`. This module has already been installed. Write a line of code that would import the `grow_horn` method from this module using the short name `horn`. (2 pt)

Your answer: _____ **from unicorn import grow_horn as horn** _____

PID:

Q8A. In real code, write a function called `sum_tuple` that takes an input (assumed to be a tuple of numeric values). Inside the function, include a loop that iterates across the items in the tuple summing all the values in the tuple. The function should return the summed value. (5 pts)

<pre>def sum_tuple(my_tuple): total = 0 for item in my_tuple: total += item return total</pre>	<ul style="list-style-type: none">- 1 pt: function defined correctly (def, colon, indent) and named correctly (sum_tuple)- 1 pt: has input- 1 pt: sum value initialized- 0.5 pt: for loop loops through input (uses input list)- 0.5 pt: input summed correctly- 1 pt: return statement; returns outside for loop
--	--

Q8B. If your `sum_tuple()` function were defined (meaning the code part A of this question was executed), what would `sum_tuple((1, 2, 3))` return? (2 pts)

Your answer: _____ 6 _____

Q8C. If your `sum_tuple()` function were defined (meaning the code part A of this question was executed), explain what would happen if you tried to execute the following: `sum_tuple(my_tuple = [1, 2, 3])`. (2 pts)

If input parameter is `my_tuple`, It would return 6. (Note that if they add checks for whether the input were a tuple or not, this answer could change); If different input parameter, would error.

Q9. In the Python standard library, there is a module called `random`. One of the methods from `random` is `choice()`, which returns one item from a collection at random. In real code, write what you would need to do to: (3 pts)

- 1) import the `random` module so you can use the `choice()` method
- 2) Use the `choice()` method return a value at random from a list containing the integers 1 through 4 inclusive.

<pre>import random random.choice([1, 2, 3, 4] or range(0, 5)) from random import choice choice([1, 2, 3, 4] or range(0, 5))</pre>	<ul style="list-style-type: none">- 1 pt: import statement correct- 1 pt: choice used correctly; has list as input- 1pt: import and choice would work together
--	--

Part V: True or False (7 pts)

Q10. For each of the following statements, circle either True or False (1 pt each):

In Python, everything is a class.	True / False
<code>my_class</code> is a better <u>class</u> name than <code>MyClass</code>	True / False
<code>do_thing</code> is a better <u>function</u> name than <code>DoThing</code>	True / False
<code>import math as unicorn</code> is valid Python syntax	True / False
Functions, Classes, and your Notebook share a single namespace	True / False
<code>pwd</code> is a shell command that prints your current working directory	True / False
Absolute Paths specify location relative to the computer's root directory	True / False

Part VI: Objects & Classes (12 pts)

Q11. You have created a class `Rocket` with two attributes, `position_x` and `position_y`. This class also has two methods: `lift_off()` and `land()`. You've created an instance of this class called `my_rocket`. Indicate how you would access information stored in the attribute `position_x` for `my_rocket` and how you would call the `lift_off()` method for `my_rocket`. (2 pts)

Access attribute: _____ **my_rocket.position_x** _____

Call method: _____ **my_rocket.lift_off()** _____

Q12A. In real code, create a class called `BasketballGame()`. This class should have four (4) instance attributes, : `home_team`, `away_team`, `home_points`, and `away_points`. It should also have one method, `play_game()`. Within the `play_game` method, add code that would determine who the winner of the game is (determined by the team with the most points), returning 'Winner: ' and the winning team's name. For ties, return 'Winner: tie' (9 pts)

```
class BasketballGame():

    def __init__(self, home_team, away_team,
                 home_points, away_points):
        self.home_team = home_team
        self.away_team = away_team
        self.home_points = home_points
        self.away_points = away_points

    def play_game(self):
        if self.home_points > self.away_points:
            out = "Winner: " + self.home_team
        elif self.away_points > self.home_points:
            out = "Winner: " + self.away_team
        else:
            out = "Winner: tie"

        return out
```

- **Class - (0.5 pt)**
- **Instance attributes**
 - **self** (1 pt)
 - **def __init__** (1pt)
 - **self.** (1 pt)
 - **Inherit correctly** (1 pt)
- **Method**
 - **def** (1 pt)
 - **conditional** (1 pt)
 - **self** (1 pt)
 - **string** (1 pt)
- **return**(0.5 pt)

Q12B. Assuming the class described in part A of this question were defined, what would the following return: (1 pt)

```
my_game = BasketballGame(home_team = "Warriors", away_team = "Sixers",
                          home_points = 100, away_points = 105)
my_game.play_game()
```

Would return: _____ **Winner: Sixers** _____