COGS 18 Exam #2

Fill out your Name and PID here:

Name: ________________________________

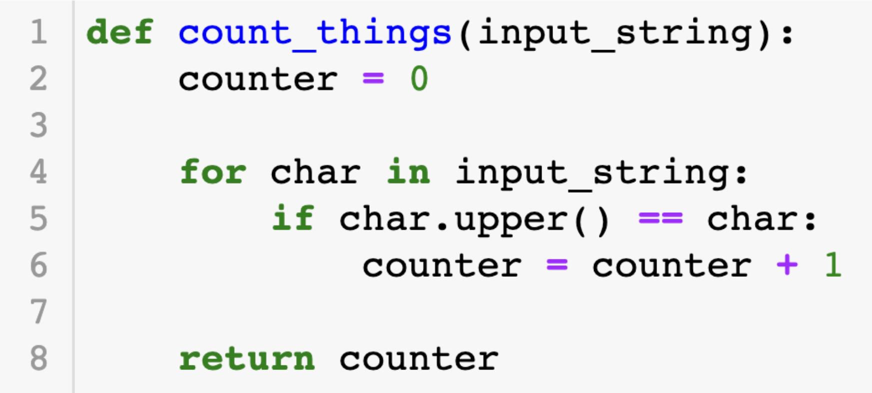
PID: ________________________________

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, your writing utensil, and a single-sided 8.5x11 sheet of paper cheat sheet.
- All work should be your own. Keep your eyes on your own exam.
- If you are unsure of what any instructions mean, raise your hand to ask a TA, IA or the Professor.
- You'll have 1h 20 min 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 30 possible points.

Q1-3 Recall that `upper()` is a string method that will convert all letters in the string it operates on into capital letters. Consider the following function to answer the following three questions:



```python
def count_things(input_string):
    counter = 0

    for char in input_string:
        if char.upper() == char:
            counter = counter + 1

    return counter
```

1. If you were to execute the function `count_things('AaBbCc')`, what would the output of the function be? (1 pt)
   - **A. 3**
   - B. 6
   - C. 'ABC'
   - D. 'AABBCC'

2. If you were to execute the function `count_things('AaBbCc')`, how many iterations of the loop would execute? (1 pt)
   - A. 0
   - B. 3
   - **C. 6**
   - D. infinite

3. Which of the following would lead to the counter being zero after the loop terminates? (1 pt)
   - A. If the `input_string` was 'AA' and the `counter` was moved inside the for loop but before the conditional
   - B. If the `input_string` were 'AA' and the `counter` was left where it is in the function provided
   - C. If the `input_string` was 'aa' and the `counter` was initialized with the value 1 instead of 0
   - D. **If the `input_string` were 'aa' and the `counter` was left where it is in the function provided**

---

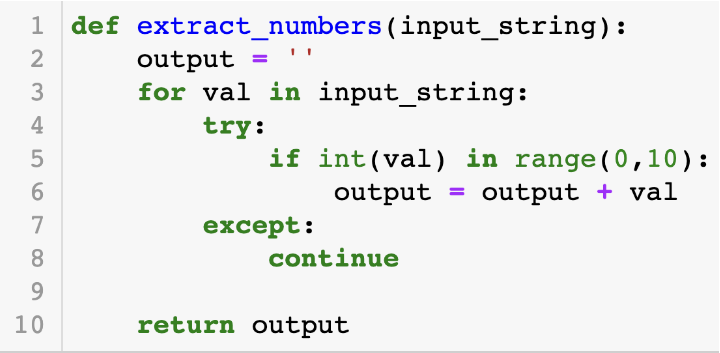
4. For each of the following statements, <u>circle *either* True or False</u> (3 pts):

Classes organize attributes and methods that operate on those attributes	**True**/False
In Python, everything is a conditional.	True / **False**
`cd` is a shell command that changes your current working directory	**True** / False
Absolute Paths specify location relative to the computer's root directory	**True** / False
Class attributes are useful for data that will change for each instance of a class	True / **False**
self is used in python to refer to a specific iteration of a loop	True / **False**

CONTINUE TO THE NEXT PAGE

PID: ______________________

Q5-7 Reference the following function to answer the following three questions:



```python
def extract_numbers(input_string):
    output = ''
    for val in input_string:
        try:
            if int(val) in range(0,10):
                output = output + val
        except:
            continue

    return output
```

5. Assuming the input to the function is a string, the `for` loop in `extract_numbers()` will (1 pt):
   A. Determine if the string contains only numeric values and then loop through those numeric values
   B. Determine if the string contains only letters of the alphabet as values and then loop through values
   C. Loop through each character in the `input_string` and carry out the `try/except` block on each character.
   D. Loop through each character in the `input_string` and convert every value in the input into numbers.

6. Which of the following values for `input_string` would return a string of length 4 as the output from `extract_numbers`()? (1 pt)
   A. `'A10987b'`
   B. `'Aabbccdd'`
   C. `'1a2b3c4d'`
   D. `'11223344'`

7. If you wanted the output from this function to be the string '678' which of the following would accomplish that? (1 pt)
   A. `extract_numbers(input_string='hello876')`
   B. `extract_numbers(input_string='876hello')`
   C. `extract_numbers(input_string='h6e7ll8o')`
   D. `extract_numbers(input_string='sixseveneight')`

---

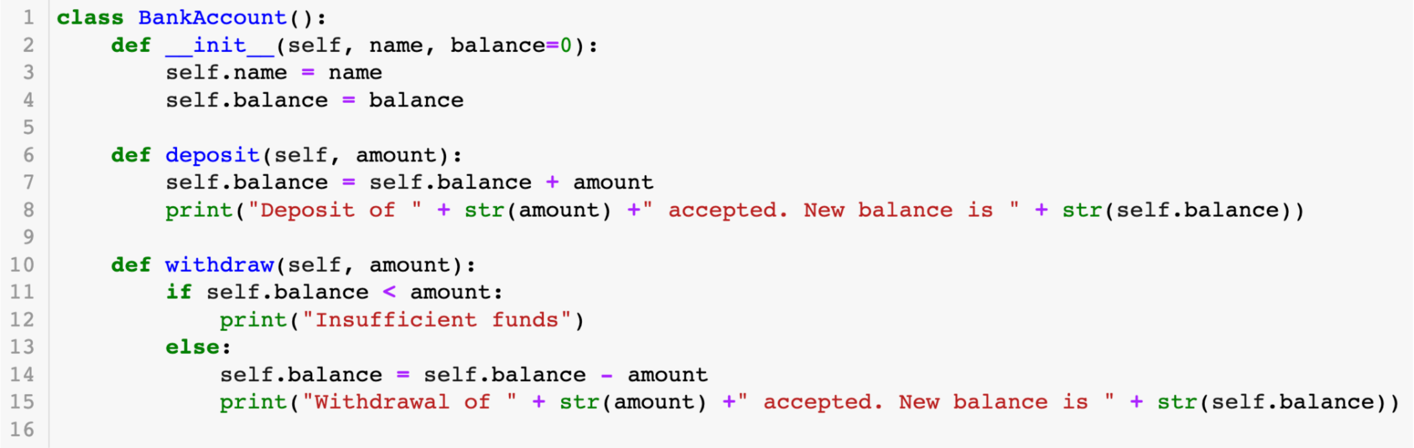
8. **Which Code Construct?** For each provided situation at left below, enter the abbreviation of the most appropriate code construct you would use for the specified situation. Choose from among the following code constructs: while loop (WL), for loop (FL) function (F), class (CL), conditional (CO). Constructs may be used more than once. (6 pts)

Situation	**Construct**
You want to carry out a set of operations on a specified input.	F
You want to repeat a specified set of operations multiple times for each item in a tuple.	FL
You want to carry out a set of operations directly on an object itself.	CL
You want to control the flow of your code, having certain code execute when one thing is true and other code execute when another thing is true.	CO
You want to repeat a specified set of operations over and over again, stopping once a condition changes (and no longer evaluates as True).	WL
You want to store information about an object so that you can carry out associated operations on that information directly.	CL

CONTINUE TO THE NEXT PAGE

PID: ______________________

Q9-12 Reference the following Class (BankAccount()) to answer the following four questions:



```
class BankAccount():
    def __init__(self, name, balance=0):
        self.name = name
        self.balance = balance

    def deposit(self, amount):
        self.balance = self.balance + amount
        print("Deposit of " + str(amount) +" accepted. New balance is " + str(self.balance))

    def withdraw(self, amount):
        if self.balance < amount:
            print("Insufficient funds")
        else:
            self.balance = self.balance - amount
            print("Withdrawal of " + str(amount) +" accepted. New balance is " + str(self.balance))
```

9. Which of the following is true regarding the attributes for this class? (1 pt)

   A. There are two class attributes, each of which has a default value specified.

   B. There are two instance attributes, each of which has a default value specified.

   C. There are two class attributes, one of which has a default value specified.

   D. There are two instance attributes, one of which has a default value specified.

10. If you were to create an instance of the `BankAccount` object, call the `deposit()` method on that instance once with an `amount` of 20 and then the `withdraw()` method once with an `amount` of 10, what would be the value stored in that object's `balance` attribute after? (1 pt)

    A. 0

    B. 10

    C. 20

    D. 30

11. What is the best explanation as to why these methods do not have a `return` statement? (1 pt)

    A. Because these methods have `print` statements that do the same thing as a `return` statement

    B. Because these methods update the value of an attribute

    C. Because these methods don't carry out any operations

    D. Because these methods replace the need for any class attributes

12. Again using the `BankAccount`() class, write a brief description of 1) what each of the following components is and 2) its role in the class. (5 pts).

Component	What	Role
`BankAccount()`	class name	Allows us to create a BankAccount type object with attributes and associated methods
`withdraw()`	method	Takes in amount from user when executing method and checks if withdrawal can be made. If balance larger than amount to withdraw, balance attribute updated and updated balance displayed.
`amount`	parameter	Allows user to specify value in both the deposit and withdraw methods to increase or decrease balance, respectively
`balance`	instance attribute	Allows object to keep track of amount of money in the bank account - starts at zero by default; updated via the methods withdraw() and deposit()

CONTINUE TO THE NEXT PAGE

PID: ____________________

Concepts (that you will implement technically on the take-home portion)**:** *For these next three questions, we're going to work to build a* `MusicPlayer` *type object. The idea here is that there will be a single class (MusicPlayer) with a few attributes (described below) and two methods: 1)* `add_song()`*, a method to add a song to our playlist and 2)* `change_volume()`*, a method that changes the value of the volume for our* `MusicPlayer`.

For the following three questions, the question will be at left and your answer will be at right. Your answer should include code and/or detailed pseudocode explaining all the components you would need to accomplish the goal at left. When completed the three boxes should include all necessary code/pseudocode for your `MusicPlayer()`:

13. To start building your **`MusicPlayer`**( ) type object, you'll first need to **create the class *and* specify the necessary instance attributes**: 1) **`playlist`** (which will be an empty list) and 2) **`volume`** (which will be 50) when an instance of the object is first created.
(2 pts)
	0.5 pts each
	- Class name
	- Instance attributes + self
	- Playlist - empty list
	- Volume - 50
	`class MusicPlayer():`
	`    def __init__(self):`
	`        self.playlist = []`
	`        self.volume = 50`

14. Now, your **`MusicPlayer`**( ) needs an **add_song**() method. This method should take a song in as input and add that song to the object's **`playlist`** attribute (from the previous question)
(2 pts)
	0.5 pts each
	- Method add_song
	- self
	- Parameter as input
	- Updates playlist attribute w/ self specified
	`    def add_song(self, song):`
	`        self.playlist.append(song)`

15. Finally, your object needs an **`change_volume()`** method. This method should take in a value specifying by how much the `volume` should change. Your code should ensure that the `volume` should only be a value between 0 and 100. (For example, if your `volume` was 90 and you specified to change_volume by 20, the value stored in `volume` should be 100…the max possible value.)
(3 pts)
	0.5 pts each
	- Method change_volume
	- Self
	- Parameter as input
	- Code to change value
	- Conditional (or other approach) to keep value between 0 and 100
	- Updates volume attribute with self specified
	`    def change_volume(self, change):`
	`        self.volume = self.volume + change`
	`        if self.volume > 100:`
	`            self.volume = 100`
	`        elif self.volume < 0:`
	`            self.volume = 0`

END OF EXAM