Name : _____

Grading TA: _____

- INTEGRITY: By taking this exam, you pledge that this is your work and you have neither given nor received inappropriate help during the taking of this exam in compliance with the Academic Honor Code of Georgia Tech. Do NOT sign nor take this exam if you do not agree with the honor code.
- DEVICES: If your cell phone, pager, PDA, beeper, iPod, or similar item goes off during the exam, you will lose 10 points on this exam. Turn all such devices off and put them away now. You cannot have them on your desk.
- ACADEMIC MISCONDUCT: Academic misconduct will not be tolerated. You are to uphold the honor and integrity bestowed upon you by the Georgia Institute of Technology.
  - Keep your eyes on your own paper.
  - Do your best to prevent anyone else from seeing your work.
  - Do NOT communicate with anyone other than a proctor for ANY reason in ANY language in ANY manner.
  - Do NOT share ANYTHING during the exam. (This includes no sharing of pencils, paper, erasers).
  - Follow directions given by the proctor(s).
  - Stop all writing when told to stop. Failure to stop writing on this exam when told to do so will result in a substantial grade penalty.
  - Do not use notes, books, calculators, etc during the exam.
- TIME: Don't get bogged down by any one question. If you get stuck, move on to the next problem and come back once you have completed all of the other problems. This exam has 7 questions on 10 pages including the title page. Please check to make sure all pages are included. You will have 50 minutes to complete this exam.

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community. I have also read and understand the requirements outlined above.

Signature: _____

Question	Points	Score
1. Vocabulary	9	
2. Multiple Choice	5	
3. List Questions	6	
4. Buzz Wins	4	
5. Bravarian Strings	5	
6. GetTemp	10	
7. Reformat	10	
Total:	49	

1. *(9 points)*
   For each of the following vocabulary terms, write a concise 1-2 sentence definition. Be brief, and to the point.

   (a) [3 pts] argument

   Solution: argument - A value provided to a function when the function is called. This value is assigned to the corresponding parameter in the function.

   (b) [3 pts] dictionary

   Solution: A mutable compound data type that associates keys with values. They map keys, which can be any immutable type, to values, which can be any type, just like the values of a list or tuple.

   (c) [3 pts] parameter

   Solution: parameter - A name used inside a function to refer to the value passed as an argument.

2. *(5 points)*
   For each of the following multiple choice questions, indicate the single most correct answer by circling it!

   (a) [1 pt] What is the value printed when the following code is executed?

```
def func(aNum,bNum):
    cNum = aNum/2-4
    if cNum > 0 and bNum:
        return aNum
    else:
```

```
        return bNum

print( func(15,5) )
```

A. `15` B. `5` C. `3` D. N/A. This code raises an exception.

(b) [1 pt] Which of the following data types are NOT sequences?

A. Tuples **B. Dictionaries** C. Strings D. Lists E. All of these are sequences

(c) [1 pt] Examine this code. What is the value referenced by `aVar` after the code is executed?

```
def func():
   greetings = "Hello"
   salutations = "Salutations!"
   for i in range(len(greetings)):
      greetings[i] = salutations[i]
   return greetings

aVar = func()
```

A. 'Salutations'!

B. 'Salut'

C. 'Hello'

D. N/A. This code raises an exception.

Use the following code to answer the next two questions.

```
aList = [5, 10, 15, 20]
bList = 2 * aList
cList = bList
```

(d) [1 pt] What list does bList reference?

A. `[2, 5, 10, 15, 20]`

B. `[10, 20, 30, 40]`

C. `[5, 10, 15, 20, 5, 10, 15, 20]`

D. `[5, 10, 15, 20, 20, 15, 10, 5]`

(e) [1 pt] Which of the following statements is true?

A. cList is an alias of bList

B. bList is an alias of aList

C. cList is an alias of aList

D. cList is a copy of bList

3. *(6 points)*
Examine the following code:

```
aList = [1,2,3,4,5]
bList = []
cList = []

for i in range(0,6,2):
   aList[i] = i
   bList.append(i)
   cList.append(bList)
```

After the code executes, what does each of the variables point at?

aList -

bList -

cList -

Solution:

```
aList = [0, 2, 2, 4, 4]
bList = [0, 2, 4]
cList = [[0, 2, 4], [0, 2, 4], [0, 2, 4]]
```

Grading:
+3 points for correct aList. +2 for one error, +1 for 2 errors.
+1 point for correct bList.
+1 point for 3 lists in cList.
+1 point for all sublist in cList matching bList.

4. *(4 points)*
Examine this code:

```
n=0
while(n<10):
    output=""
    if(n%2==0):
        output= output + "Buzz buzzes."
    if(n%3==0):
```

```
        output= output + "Buzz wins."
        print(output)
    n= n + 1
```

What is printed to the screen when this code is executed?

Solution:

```
Buzz buzzes.Buzz wins.
Buzz wins.
Buzz buzzes.Buzz wins.
Buzz wins.
```

Grading: 1 point for each correct line in correct position.
-1 point for each incorrect line.

5. *(5 points)*
Examine the following code.

```
def iSeeYou():

    aString = "THIS IS NOT A TEST"
    bString= aString[::-1]
    aLetter = bString[12]
    bLetter = bString[0]

    cString = "MULL IT OVER"
    dString = cString[3:0:-1]

    nonsenseString = "MLOIOWNCXA"
    eString = nonsenseString
    fString = eString[0:7:3]

    print(aLetter + dString + fString + nonsenseString[9]+ bLetter + aLetter)

iSeeYou()
```

What is printed to the screen when this code is executed?

Solution:

```
ILLUMINATI
```

Grading: 1 point for "I" in front and back
1 point "LLU"
1 point for "MINA"
1 point for "T".
1 point for getting the full word correctly
-1 point for any "extra" incorrect letter(s) between correct segments.

6. *(10 points)*

You are hired to write the control software for a bio-reactor. As part of this job, you need to write a function named `getTemp` that takes no parameters. It should display a message (prompt) to the user asking them to enter a temperature between 30 and 90 degrees C. (inclusive) ("Please enter a temp between 30 and 90 degrees C") Note that the user may enter a temp such as 45.8 which is valid.

Your function should return the temperature the user entered as a float. If the user does not enter a valid temperature (not a number e.g. "Fish", a number lower than 30 or higher than 90) you should print out "Invalid Temp, try again!", and then repeat the prompt asking for a temperature until they get it right.

Solution:

```
def getTemp():
   userStr = input("Please enter a temp between 30 and 90 degrees C")
   try:
     userFlt = float(userStr)
     if userFlt > 90 or userFlt < 30:
        print("Invalid Temp, try again!")
        return getTemp()
     else:
        return userFlt
   except:
     print("Invalid Temp, try again! ")
     return getTemp()
```

Note: Students could also use a while loop instead of recursion to solve this problem.

Grading:
1 point for prompting the user with the correct text
1 point for trying to convert to a float.
2 points for correctly catching float conversion exceptions
2 points for correctly checking >=30 and <=90. (-1 for not allowing 30 or 90)
2 points for asking again and again until the user gets it right.
2 point for returning the correct float value.

7. *(10 points)*

Write a function called **reformat** that will take in a list of items as a parameter. Check the number of elements in the list. If the list has an odd number of elements, copy the list, reverse the copy, remove the center/middle element from the copy, and return the copy.

If the list has an even number of elements, you should return a tuple that contains all of the elements in the list (in the same order) with the following exceptions: do not copy zeros (int/float) or empty strings into the returned tuple.

Example test cases:

```
>>>reformat([1,"happy",7,False,0])
[0, False, 'happy', 1]
>>> reformat([0,"",12,"Learning is fun!",44.8, True])
(12, 'Learning is fun!', 44.8, True)
```

Solution:

```
def reformat(myList):
    if(len(myList)%2==1):
        cpL = myList[:]
        cpL.reverse()
        del cpL[len(myList)//2]
        return cpL
    else:
        newTup=()
        for item in myList:
            if(not(item==0 or item==""")):
               newTup+=(item,)
        return newTup
```

Grading:
1 point for getting length of input list
1 point for detecting if it is even or odd
1 point for making a COPY of the list. (on odd number lists)
1 point for correctly reversing the copy
1 point for correctly removing center element.
1 point for returning the COPY of the list.
1 point for making a tuple (on even number lists)
. 1 points for correctly adding elements to the tuple
1 points for omitting the 0 and "" elements!
1 point for returning the new tuple.

This page intentionally left blank. You may use it for scratch paper. If you place an answer on this page, box it, indicate which problem it is for by number, and BE SURE TO WRITE "Answer on last page" at the problem location!