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 is academic misconduct.
  - 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 10 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. Audio	0	
3. Short Answer	18	
4. True or False	4	
5. CountUp	6	
6. Loving While	4	
7. Fill in the Blank	6	
8. Same IFs	4	
9. DooWaa	6	
10. Who Won?	16	
Total:	73	

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] evaluate

Solution: To simplify an expression by performing the operations in order to yield a single value.

(b) [3 pts] float

Solution: A Python data type which stores floating-point numbers. Unlike INT's, floating point numbers have a fractional component (value after the decimal place).

(c) [3 pts] iteration

Solution: Repeated executation of a set of statements.

2. *(0 points)*
Circle the approprate answer for you:
In lecture, I typically sit in the **front** / **middle** / **back** of the lecture hall.

Jay's voice is **too quiet** / **just right** / **to loud** for me.

3. *(18 points)*

For each of the following questions, give a brief answer:

(a) [10 pts] Pretend you are the python interpreter. Evaluate each of the expressions below. Write down the value that they evaluate to, and the type of that value in the provided columns. If the expression is not valid python syntax, or will throw an exception, simply write "Error". The first line has been provided as an example.

Expression	Result	Type
1 + 2 * 3	7	int
6 % 4 / 3		
"a" + "b" * 2		
len(range(4,5))		
3.5 * 2		
4**(3.0/2)		

Solution:

Expression	Result	Type
1 + 2 * 3	7	int
6 % 4 / 3	0	int
"a" + "b" * 2	"abb"	str
len(range(4,5))	1	int
3.5 * 2	7.0	float
4**(3.0/2)	8.0	float

Grading: +1 point for each correct answer.

(b) [2 pts] What is printed to the console if the following code is evaluated?

```
print "%.2f" % (4/2)
```

Answer: ____________________

Solution: Answer: **2.00** Grading: +2 for the correct answer. +1 if they know it has 2 decimal places after the . but get the value wrong.

(c) [6 pts] Pretend you are the python interpreter. Write down what would be printed out when the following code is executed:

```
def foo(n):
     print n*3
     return n*2

print "Begin"
y = foo(7)

if (y > 15):
    print "Breakfast"
elif (y > 10):
    print "Lunch"
if (y > 5):
    print "Dinner"
else:
    print "Dessert"

if ('a' < 'b'):
    print "Apple"
elif ('a' < 'A'):
    print "Peanut"
else:
    print "Butter"
print "Finished"
```

Solution: Begin
21
Lunch
Dinner
Apple
Finished

Grading: +1 for each correct line.
-1 if they don't write them vertically.
-1 for each incorrect line, or line out of order.

4. *(4 points)*
Examine the following function:

```
def trueOrFalse(a,b,c,d):
  a = raw_input("Enter a: ")
  b=b
  d=c
  c = b/d
  d=d

  #Print all variables
  print a, b, c, d
```

Assume the function is called as follows:

```
trueOrFalse(4,1,2,3)
```

and the number 30 is typed for the raw_input prompt: *Enter a:*

At the end of the function, we print the four variables. At this point in the flow of exectuion, indicate if each of the following statements is True or False:

T / F 1. The variable a will point at 30 whose type is int.
T / F 2. The variable b will point at 2 whose type is int.
T / F 3. The variable c will point at 0.5 whose type is float.
T / F 4. The variable d will point at 3 whose type is int.

Solution: F - Type is string
F - b will be 1
F - c will be zero due to integer division
F - d will be 2
Grading: +1 points for each correct answer.

5. *(6 points)*

Write a function called **countUp** that accepts two integer parameters. The function will print out all integers between the two parameters (excluding both parameters!) in ascending order. You may assume that the two parameters are valid integers, and that the first parameter will always be smaller than the second parameter.

Example test case:

```
>>>countUp(1,5)
2
3
4
>>>
```

Solution:

```
def countUp(a,b):
   a = a + 1
   while a < b:
      print a
      a = a+1

or:
def countUp(a,b):
   for x in range(a+1,b):
      print x
```

Grading: 2 points for a correct header.
2 points for starting the printout at a+1
2 points for stopping correctly just before b.

6. *(4 points)*
Pretend you are the python interpreter and the following code is executed.

```
x = 3
while x >= 0:
    if x%2 == 0:
        print "chocolate"
    elif x-3<0:
        print "flowers"
    else:
        print "valentine"
    x = x - 1
```

What is the output that is printed to the screen?

Solution: Printed to screen (vertically):
valentine
chocolate
flowers
chocolate

Grading: +4 points for getting the printout 100% correct.
-1 for each incorrect line or line in incorrect order.
-1 if not printed vertically.

7. *(6 points)*
Complete each statement below by filling in the blank:

1. You use ____________________ in your python programs, denoted by the # symbol, to explain in natural language how your program works.
2. In python, the single equal sign is used for ____________________, while the double equal sign is used for ____________________.
3. A good recursive function requires three things: First, it must call ____________________. Second and third, it must have a ____________________ condition and it must ____________________ it.

Solution: Grading: +1 for each correct answer.
Comments
Assignment, Equality Checking

Call itself, Terminating, (work towards, move towards, approach)

8. *(4 points)*

Which two (of the following four) blocks of code has the exact same meaning? [You may assume that booleanExpression1...booleanExpression3 are variables that point to a boolean value, and have the same values for all four blocks of code.]

```
A:
if booleanExpression1:
    print "Yeah"
elif booleanExpression2:
    print "Yeah"
elif booleanExpression3:
    print "Yeah"
```

```
B:
if booleanExpression1:
    print "Yeah"
elif booleanExpression2:
    print "Yeah"
elif booleanExpression3:
    print "Yeah"
else:
    print "Yeah"
```

```
C:
if booleanExpression1 or booleanExpression2 or booleanExpression3:
    print "Yeah"
```

```
D:
if booleanExpression1:
    print "Yeah"
if booleanExpression2:
    print "Yeah"
if booleanExpression3:
    print "Yeah"
```

__________________ and __________________ give identical results.

Solution: A and C have give the identical result. 4 points if correct.

9. *(6 points)*
Fill in the blanks so that, when run, the code below will output the followng:

```
>>> func1()
DooWaa
Diddy
Diddy
Dum
Diddy
Doo
```

```
def func1():
    print "DooWaa"
    for i in range(  ___________________ ):

        print __________________

        if i == ________________ :
            print "Dum"
    print "Doo"
```

Solution:

```
def func1():
    print "DooWaa"
    for i in range(___3___):
        print __"Diddy"__
        if i == ___1___ :
            print "Dum"
    print "Doo"
```

Grading: 2 points for each correct blank. -1 for any minor syntax errors. (leaving out quotes, etc)

10. *(16 points)*

Write a function named `whoWon` that takes in four parameters (team1, score1, team2, score2). The team names (team1 and team2) will be passed in as strings. The scores for the teams (score1 and score2) will be passed in as integers.

Your function should **return** a string that states who the winning team is, what their score was, how many points they won by, and who they beat. If the game was a tie, return the string “It’s a tie!”. (See examples below).

Example test cases:

```
>>> result =  whoWon("Georgia Tech", 30, "Duke", 20)
>>> result
"Georgia Tech scored 30 points and won by 10 points over Duke!"
>>> whoWon("United States", 1, "England", 1)
"It’s a tie!"
```

Solution:

```
def whoWon(team1,score1,team2,score2):

   if score1 > score2:
      wonBy = score1-score2
      aStr = team1 +" scored " + str(score1) +" points and won by " +str(wonBy)+ "
   elif score 2 > score1:
      wonBy = score2 - score1
      aStr = team2+ " scored " + str(score2) + " points and won by " +str(wonBy) +
   else:
      aStr = "It’s a tie!"

   return aStr
```

Grading: 2 points for a correct header.
4 points for correctly determining which team won
2 points for correctly calculating wonBy difference. 4 points for creating the string (1 point each for winning team, winning score, wonBy score, losing team)
2 points for correctly detecting a tie and returning appropriate tie string
2 points for returning the string.