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 9 questions on 11 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. Fill in the Blanks	3	
3. Short Answer	20	
4. Short Answer 2	12	
5. Greeting Card	8	
6. isNegative	5	
7. CountUp	8	
8. SmallestOf3	8	
9. Parrot	9	
Total:	82	

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] keyword

Solution: A reserved word that is used by the compiler to parse program; you cannot use keywords (such as if, def, and while) as variable names.

(b) [3 pts] runtime error

Solution: An error raised by the python runtime while the program is executing if something goes wrong. For example, a divide by zero error.

(c) [3 pts] parameter

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

2. *(3 points)*
Fill in the blanks:

Python has several ways to repeatedly execute a block of code.

A ____________________ loop is used to iterate through a sequence, or repeat a block of code a specific number of times while a ____________________ loop will repeat a block of code as long as its' boolean expression evaluates to True.

A function that uses ____________________ can also repeat blocks of code, but must call itself to do so.

Solution: Python has several ways to repeatedly execute a block of code. A **for** loop is used to iterate through a sequence, or repeat a block of code a specific number of times while a **while** loop will repeat a block of code as long as its' boolean expression evaluates to True. A function that uses **recursion** can also repeat blocks of code, but must call itself to do so.

Grading: one point per correct blank.

3. *(20 points)*
   For each of the following questions, give a brief answer:
   (a) [12 pts] Write the value that the following expressions evaluate to, as well as the type of the result. Write "ERROR" in both columns if the expression would result in an error. The first line has been completed for you as an example.

Expression	**Value**	**Type**
3+4	7	int
1.5 + 3		
05/01/10		
float(5/4)		
"a" < "c"		
(True or False) and (not False)		
(3000 * 3 + 1)		

Solution:

Expression	**Value**	**Type**
1.5 + 3	4.5	float
05/01/10	0	int
float(5/4)	1.0	float
"a" < "c"	True	Bool(ean)
(True or False) and (not False)	True	Bool(ean)
(3000 * 3 + 1)	9001	int

Grading: +1 point for each correctly filled in table square.

   (b) [5 pts] Assume the following code is run:

```
def foo(x):
   print x
   x = int(x)
   print x + 4
   y = str(x)
   print y + "5"
   return y

z = foo(6.5)
```

What is displayed to the screen?

Solution: Answer:
6.5
10
65

Grading: +1 for each line. (-1 if they don't write them vertically, or for each incorrect item.)

What is stored in (refered to by) the variable z? (Be sure to specify it's type!)

Solution: Answer: "6"

Grading: +1 point for 6, +1 point for quotes or the word "string" indicating its type.

(c) [3 pts] What is printed when the following lines of code are evaluated? Be sure to format your output exactly as Python would.

```
l = ["open", "close", "in", "out", "up", "down" ]
for i in range(0,6,2):
    print l[i]
```

Solution: Answer:

```
open
in
up
```

Grading: +1 for each word. (-1 if they don't write them vertically, or for each incorrect word.)

4. *(12 points)*
For each of the following questions, give a brief answer:

(a) [5 pts] You have been asked to finish the function below by filling in the blanks. The function will accept an integer parameter, and is supposed to return the string "odd" if the number is odd, or "even" if the number is even.

```
____ oddOrEven( number ):
   if number ___ 2 == 0:
      _______ "even"
   _____ :
      _______ "odd"
```

Solution:

```
def oddOrEven( number ):
   if number % 2 == 0:
      return "even"
   else:
      return "odd"
```

Grading: +1 for each correct blank.

(b) [4 pts] What is printed when the following lines of code are evaluated?

```
if  (0 + 1 == True):
    print "false"
if (5/2 > 2):
    print "banana"
else:
    print "pear"
if (type("2") == int):
    print "apple"
elif (23 % 7 <= 2):
    print "peach"
if (int(3.1) == float(3.0)):
    print "watermelon"
```

Solution: Answer:
false
pear
peach
watermelon

Grading: +1 for each word. (-1 if they don't write them vertically, or for each incorrect word.)

(c) [3 pts] What is printed when the following lines of code are evaluated?

```
if 5<5:
    print "A"
if 5+5:
    print "B"
elif 5==5:
    print "C"
if "D":
    print "D"
if 0:
    print "E"
else:
    print "F"
```

Solution: Answer:
B
D
F

Grading: +1 for each correct letter. (-1 if they don't write them vertically, or for each incorrect letter.)

5. *(8 points)*
The following code has four errors. Clearly indicate and correct the errors on the code provided.

```
def greetingCard():
    name = raw_input("please enter your name:")
    age = raw_input("please enter your age:")

    if (age <= 12)
        message = "new toys"
    elif (age > 12) and (age <= 20):
        message = "angst and trendy stuff"
    elif (age >= 21) and (age < 40):
        message = "drinks"
    Else:
        message = "time to turn 40 again"
    print "Happy birthday %s Hope you get lots of %s!" %name %message
```

Solution: Solution (the four errors)
1 - convert age to int
2 - colon after if
3 - Else should be lowercase
4 - need only 1 percent sign and () on last line

Grading: +2 points for each error. 1pt for identifying it, 1pt for fixing it correctly.

6. *(5 points)*
Write a function named `isNegative` that takes in an integer parameter and returns a boolean. If the parameter is negative, the function should return the boolean value True. If the parameter is zero or positive, it should return the boolean value False.

Example test cases:

```
>>> print isNegative(4)
False
>>> print isNegative(-7)
True
>>>
```

Solution:

```
def isNegative( aNum):
   if aNum < 0:
return True
   else:
  return False
```

Grading: 1 point for a correct header.
2 points for testing if the number is positive/negative correctly. 1 point for returning False if the number is positive or zero.
1 point for returning True if the number is negative

7. *(8 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 *using a while loop.* You may assume that the two parameters are valid integers, and that the first parameter will always be smaller than the second parameter. Note that you **may not use a for loop** in your solution!

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
```

Grading: 2 points for a correct header.
2 points for starting the printout at a+1
2 points for printing multiple numbers with the while loop.
2 points for stopping correctly just before b.

8. *(8 points)*
Write a function called **smallestOfThree** that accepts three integer parameters. The function will return the smallest of the three parameters. If several of the paramters are the same (and the smallest) it may return either of the same (smallest) parameters.

Example test cases:

```
>>>smallestOfThree(1,5,10)
1
>>>smallestOfThree(5,5,5)
5
>>>smallestOfThree(5,5,1)
1
```

Solution:

```
def smallestOf3(x,y,z):
   smallest = x
   if b <= smallest:
      smallest = b
   if c <= smallest:
      smallest = c
   return smallest
```

Grading: 2 points for a correct header.
2 points for returning (instead of printing)
2 points for working in the case where all three numbers are different (1,5,4)
2 points for also working when some of the numbers are the same (e.g. 5,5,1), which typically means they have to check that using ¡= instead of just ¡, unless the ¡ check doesn't lead to an error when items are the same... try with 5,5,1 for example.

9. *(9 points)*
Write a function named **parrot**. When ran, this function will prompt the user "Say Something:" and then obtain a string of input from the user. The function will then print whatever the user typed back to the user, and repeat this procedure (of asking the user to "Say Something:" and repeating it) until the user types "stop". Hint: A while loop or recursion may be the way to implement this.

Solution:

```
def parrot( ):
  keepGoing = True
  while keepGoing:
    userStr = raw_input("Say Something!")
    if userStr == "stop":
       keepGoing = False
    else:
       print userString
```

Grading: 1 point for a correct header.
2 points for getting user input as a string.
2 points for printing the user input.
2 points for repeating multiple times.
2 points for stopping the repetition when the user types stop.