Name : _____

1. *(2 points)*
TA Name: _____

- 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 7 questions on 8 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
TA Name	2	
2. Vocabulary	9	
3. Fill in the Blank	4	
4. Pig	11	
5. GUI coding	14	
6. CSV Add Evens	12	
7. List Question	4	
Total:	56	

2. *(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] sequence

Solution: Any of the data types that consist of an ordered collection of elements, with each element identified by an index.

(b) [3 pts] class

Solution: A user-defined compound type. A class can also be thought of as a template for the objects that are instances of it. (A class is used to instantiate objects.)

Looking for: Compound Data Type , can specify functions and group data, used to instantiate objects.

(c) [3 pts] object

Solution: A compound data type that is often used to model a thing or concept in the real world. It bundles together the data and the operations that are relevant for that kind of data. Instance and object are used interchangeably. (An object whose type is of some class.)

Looking for: Compound Data Type, Includes data & operations/functions/methods.

3. *(4 points)*
Complete each statement below by filling in the blank with either 'object(s)' or 'class(es)'.

1. _____ are defined in python files.

2. _____ are instantiated.
3. Using common Python convention, the 'self' parameter in a method defined in a(n) _____ refers to a(n) _____.
4. You may have any number of _____ created from one _____.
5. A(n) _____ variable affects all _____ of that _____.

Solution: Blanks: classes,objects, class, object, class, objects, class.

Full text:
Classes are defined in python files.
Objects are instantiated.
Using common convention, the 'self' parameter in a method defined in a class refers to an object.
You may have any number of objects created from one class.
A class variable affects all objects of that class.

Grading: Total problem worth 4 points. Minus one point for each incorrect blank. (e.g. student with one incorrect answer receives a 3) (overall total may not go below zero).

4. *(11 points)*
Examine the code below. Write down exactly what would be printed if the code was executed. (If it would not run, explain why not.)

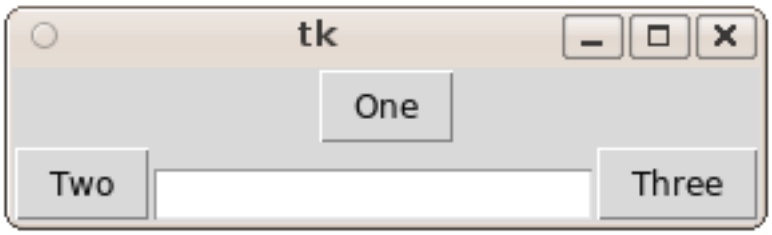
```
class Pig:
    classCounter = 0
    def __init__(self, diceRolls):
        self.diceRolls = diceRolls
        self.points = 0

    def roll(self):
        rolled = self.diceRolls[Pig.classCounter]
        Pig.classCounter = Pig.classCounter + 1
        print("You rolled a {}".format(rolled))
        self.points = rolled + self.points
        print("Score: {}".format(self.points))
        if rolled == 1:
            self.points = 0
            return 0
        else:
            return rolled

    def keepGoing(self, answer):
        try:
            if answer == 'Yes':
                self.roll()
            elif answer == 'No':
                return
            else:
                return self.keepGoing()
                print("Please enter a enter either Yes or No")
        except:
            print("You did something wrong")

rollList = [6,1,4,3]
game1 = Pig(rollList)
game1.roll()
game1.keepGoing('Yes')
game2 = Pig(rollList)
game2.roll()
game2.keepGoing('Maybe')
game2.roll()
print(game2.points)
print(game1.points)
```

Solution:

```
You rolled a 6
Score: 6
You rolled a 1
Score: 7
You rolled a 4
Score: 4
You did something wrong
You rolled a 3
Score: 7
7
0
```

Grading:
+1 for each correct line in the correct order.
-1 for each extra line, line w/ incorrect number, or out of order line.
-1 (once) for writing quotes around any of the printouts.

5. *(14 points)*
Write python code that will display a GUI that looks like the following when executed.



Note that the entry is the default width and in normal state and that the entry is against the bottom of the window.

The example GUI was made with the pack layout manager, but you may use the grid layout manager if you can make it produce the same results.

When the user clicks the "One" button, it must cause the string "Done!" to appear in the entry. (If it is clicked twice, the entry should only display a single "Done!").

Write python/tkinter code for this program. You may use classes, but are not required to do so.

Solution:

```
from tkinter import *

win  = Tk()
sv = StringVar()
def aFunc():
    sv.set("Done!")
Button(win, text="One",command=aFunc).pack(side=TOP)
Button(win, text="Two").pack(side=LEFT)
Button(win, text="Three").pack(side=RIGHT)
Entry(win, textvariable=sv).pack(side=BOTTOM)

win.mainloop()



Grading:
+1 point for importing tkinter correctly
+1 point for creating the window.
+4 points for creating the 4 widgets (3 buttons, 1 entry)
+1 point for aligning TWO button to the left of the entry. (grid or pack)
+1 point for aligning the THREE button to the right of the entry. (grid or pack)
+1 point for Aligning the ONE button above the entry and centered. (grid or pack)
+1 point for placing the ENTRY at the bottom, centered, touching the bottom of
  the window (pack Bottom or anchor South)
+1 point for correctly linking the one button command to a function/method.
+2 points if that function/method causes "Done!" to be displayed in the
entry when called.
+1 point for running the mainloop.



Misc minuses:
-1 point for adding a widget to a nonexisting container (undefined variable)
```

6. *(12 points)*

Write a function `addEvens` that takes in one parameter, the name of a CSV file as a string. This CSV file will contain integers; you should add up only the EVEN numbers on each row, then **return** a list with the totals. The first item in the list is the sum for the first row, the second item in the list is the sum for the second row, etc...

Solution:

```
def addEvens(fn):
   import csv
   f = open(fn)
   reader = csv.reader(f)
   rowsums = []
   for row in reader:
      rowtotal = 0
      for item in row:
         value = int(item)
         if value%2 == 0:
            rowtotal += value
         rowsums.append(rowtotal)
   f.close()
   return rowsums
```

Grading:
+1 Correct function definition
+1 Opens the file
Using CSV Reader:
+1: Imports CSV
+1: Correctly creates CSV reader
+1: Iterates through the reader/gets rows out.
Not using CSV Reader:
+2: Gets each line from the file.
+1: Splits the line on commas

+2: Converts each item to an int
+1: Checks to see if items are even
+1: Sums even item(s) in the row
+1: Appends row total to list of sums in correct order
+1: Closes the file
+1: Returns a list of sums.

7. *(4 points)*
Examine the code below. Next to it, write exactly what would be printed when it is executed.

```
aList = [1,2,3]
bList = []
bList.append(aList)
bList.append(aList[:])
bList.append(aList[1:3])
aList[2] = 777
bList.append(aList)
for item in bList:
    print(item)
```

Solution:

```
[1, 2, 777]
[1, 2, 3]
[2, 3]
[1, 2, 777]

Grading:
+1 for each line that is exactly correct.
-1 for any extra lines.
```

The rest of 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!