

Name : _____

1. (2 points)

Section 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 7 questions on 9 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. CSV 1,2,3	3	
4. Multiple Choice	13	
5. W vs A	4	
6. Readline Reading	6	
7. QB Rating	8	
Total:	45	

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] delimiter

Solution: A character that is used to separate (or delimit) separate data items in a string. The default delimiter for comma separated value (CSV) files is the comma.

Looking for: Separation of data items.

(b) [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.

(c) [3 pts] slice

Solution: A subsequence copied from a sequence specified by a range of indices. The slice operator is: `sequence[start:stop]`.

3. (3 points)

Write down exactly what the following code would write out to the file named exam.csv:

```
import csv
data = [ ["One,", "Two,", "Three,"], ["Four,", "Five,", "Six,"] ]

file = open("exam.csv", "w", newline="")
writer = csv.writer(file)
```

```
writer.writerow(data)
file.close()
```

Solution:

```
"One,","Two,","Three,"
"Four,","Five,","Six,"
```

Grading:

+3 has the correct lines, with all commas. +2 Only has 1/2 of the commas. +1 Doesn't have any of the commas.

4. (13 points)

For each of the following multiple choice questions, indicate the best answer. Indicate your selected answer by circling it.

- (a) [3 pts] Examine the following code which defines a class and then creates an instance of the object, then answer the following questions about it:

```
class Test2:
    def __init__(self,x,y):
        self.x=x
        y=self.recursionFun(y)

    def recursionFun(self,x):
        print("recursionFun called, x is:", x)
        for i in x:
            if int(i) > 5:
                return self.x
            else:
                self.x=self.x+self.recursionFun(i[0])
        return self.x
```

```
app=Test2(3,['12',8,[99],21])
```

What is app.y? A. None B. 0 C. 3 D. 6 **E. Not Defined.**

What is app.x? A. None B. 0 **C. 3** D. 6 E. Not Defined.

How many times total was the recursionFun method called, either by itself or other methods?

A. 0 **B. 1** C. 2 D. 3 E. 4 F. 5

- (b) [1 pt] After this code runs:

```
aList = [9, 4, 7, 5, 1]
```

```
bList = aList.sort()
```

```
aList[-1] = "Falcons"
```

Which of the following statements is True?

A. `aList == [9, 4, 7, 5, 'Falcons']`

B. `aList == [1, 4, 5, 7, 'Falcons']`

C. `bList == [1, 4, 5, 7, 9]`

D. Both B and C are True.

- (c) [1 pt] An instance variable:

A. is a variable which is accessible from any function in your Python file.

B. is always declared inside of an init method.

C. is contained within a class rather than an object.

D. is accessible from any method inside of the class.

E. belongs to all objects.

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

```
aTuple = ('GaTech', 'U[sic]GA', 'Georgia Southern')
bTuple = ('Emory', 'G State')
cTuple = ('SCAD')
finalTuple = aTuple + bTuple[ : ]
finalTuple = cTuple + finalTuple
```

- A. ('GaTech', 'U[sic]GA', 'Georgia Southern', 'Emory', 'G State')
- B. ('SCAD' , 'GaTech', 'U[sic]GA', 'Georgia Southern', 'Emory', 'G State')
- C. ('SCAD' , 'GaTech', 'U[sic]GA', 'Georgia Southern', 'Emory')
- D. This code throws an error.**
- (e) [1 pt] Strings are immutable because...?
- A. Once created, you can not change any items in a string.**
- B. Once created, you can only append to the end of a string.
- C. You can not add two strings together.
- D. You can not multiply two strings together.
- (f) [1 pt] Which of the following correctly assigned the variable X to refer to a tuple?
- A. `x = (1, [1,2], "three")`
- B. `x = ()`
- C. `x = (1)`
- D. Both A and B**
- E. A, B, and C.
- (g) [1 pt] Which of the following is true about the keys in a dictionary?
- A. An integer can be a key.**
- B. A list can be a key.
- C. A dictionary can be a key.
- D. All keys must be mutable.
- E. All of the above are False.
- (h) [1 pt] Assume you have an empty dictionary created as follows:
- ```
aDict = {}
```
- Which one of the following statements would not raise an error?
- A. `aDict[ ["1"] ]= "one"`
- B. `aDict.items()`**
- C. `aDict[aDict]="two"`
- D. `aDict={1;1}`

- (i) [1 pt] When reading in CSV files using the CSV Reader module:
- A. you do not have to manually close the file after you are finished.
  - B. you can use readlines to retrieve all of the rows from the file as a list of lists.
  - C. data is returned as a tuple of strings, where each string is one element in a row.
  - D. you need to iterate through the reader object to retrieve each row.**
- (j) [1 pt] Which of the following is/are valid functions to use with writing to a CSV file using a CSV Writer?
- A. write
  - B. writeline
  - C. writerow
  - D. writelines
  - E. writerows
  - F. A and C are valid choices.
  - G. B and D are valid choices.
  - H. C and E are valid choices.**
- (k) [1 pt] When reading in a CSV file using the CSV Reader from the CSV module, which of the following statements is true?
- A. Numeric data in the CSV file is automatically converted from a string to a numeric type.
  - B. We can use readlines to read in all rows.
  - C. We need to check for an empty row to signal the end of the CSV file.
  - D. The CSV Reader will not automatically close the file after it is finished reading.**

5. (4 points)

Briefly describe the difference between opening a file in "W" mode vs opening a file in "A" mode.

**Solution:**

Opening a file with "W" (write) mode will erase anything in the file if it exists (2pt), or will create it if it doesn't exist.(0pt) Opening a file with "A" (append) mode will add to the end of an existing file (2pt), or create a new file if the named file does not exist.(0pt)

6. (6 points)

The text file `sample.txt` has three lines, with the numbers 1, 2, 3 followed by a newline on each line respectively. Specifically, it looks like the following:

```
1
2
3
```

Write down exactly what would be printed when the following code is executed, being sure to include line breaks where appropriate:

```
a = open("sample.txt")
for item in a.readline():
 print(item)
b = a.read()
b.split('\n')
print(b)
c = a.readlines()
a.close()
print(c)
```

**Solution:**

```
1

2
3

[]
```

Grading: 1 point for the 1 alone

1 point for a larger space (3 newlines) between the 1 and the 23

2 points for the 2,3 (one point each)

1 point for newline between 3 and the []

1 point for the empty list [].

Subtract a point for every extra item written down after a wrong item appears.

EXAMPLE: Just having 1,2,3 in a line vertically with nothing else is worth 2 pts.

7. (8 points)

Write a function called `qbRating` that accepts a dictionary as a parameter. The dictionary will be structured as follows:

```
nfl = { 'Falcons':[360,'Matt Ryan'], 'Broncos' : [450, 'Peyton Manning'], ... }
```

such that each key/value pair is of the form: `team : [passing yards, "Players Name"]`

The dictionary may have the player with the top passing yardage from many different teams, not just the 2 shown in our example.

The `qbRating` function will return a list of tuples, in the following form:

```
[(passing yards, "Player Name"), (passing yards, "Player Name"), ...]
```

The tuples must be sorted by the number of passing yards (largest first). For the example above, the output would look like:

```
[(450, 'Peyton Manning'), (360, 'Matt Ryan'), ...]
```

**Solution:**

```
def qbRating(adict):
 new = []
 for i in adict:
 listy = adict[i]
 listy = (listy,)
 new.append(listy)
 new.sort() # or new.sort(reverse=True) and not new[::-1] below...
 return new[::-1] #Reverse the list before returning.
```

Grading:

- +1 correct function header
- +2 iterates through the dictionary
- +2 correctly adds a tuple of values to a new list.
- +2 sorts the final list.(+1 for any sort, +1 for correct order)
- +1 returns the correct list



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!