

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 7 questions on 12 pages including the title page. Please check to make sure all pages are included. You will have 1 hour and 45 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. Multiple Choice	21	
2. Alias vs. Copy	6	
3. Recursion Fun	5	
4. DooWaa	6	
5. Parrot	9	
6. Replace A/E	10	
7. fileMin	10	
Total:	67	

1. (21 points)

For each of the following multiple choice questions, indicate any and all correct answers! Each question will have at least one correct answer. Indicate your selected answer(s) by circling it.

(a) Given the following code, what is printed?

```
print( list( range(1, 5)) )
```

- A. [1, 2, 3, 4, 5]
- B. [1, 2, 3, 4]
- C. [1, 1, 1, 1, 1]
- D. [2, 3, 4, 5]
- E. None of the above

(b) Assume the x, y and z variables are already defined as follows:

```
x = [1,2,3,4,5]
```

```
y = (0,1,2,3,4)
```

```
z = {"one" : "data", 2 : "data2", 3 : "data3"}
```

Which of the following code fragments cause an exception (if ran)?

- A.

```
for mystery in range(0,6):  
    print(z[mystery])
```
- B.

```
for mystery in y:  
    print(x[mystery])
```
- C.

```
for mystery in z:  
    print(z[mystery])
```
- D.

```
for mystery in x:  
    print(y[mystery])
```
- E.

```
print(z[0])
```
- F.

```
print(y[x[4]])
```

- (c) Which name(s) below will give an error message because it is **not** a valid Python variable name?
- A. picture-1
 - B. picture_1
 - C. picture1
 - D. picture
 - E. None of the above
- (d) Assume the following code has been executed by the python interpreter:

```
def mysteryFunc():
    x = 17
    return print(x)

def mysteryFunc2():
    for x in range(5,15):
        print(x)
    return x

x = [1,2,3]
x = x.append(4)
a = mysteryFunc()
b = mysteryFunc2()
```

Which of the following is/are correct?

- A. **The value in x is None**
 - B. The value in x is [1, 2, 3, 4]
 - C. The value in a is 17
 - D. **The value in a is None**
 - E. the value in b is 14
 - F. the value in b is 15
 - G. **The value in b is 5**
 - H. The value in b is None
- (e) In Python, compound data types include:
- A. Integers
 - B. Floats
 - C. **Lists**
 - D. Booleans
 - E. **Strings**

(f) Immutable compound data types include:

- A. Lists
- B. Tuples**
- C. Strings**
- D. Dictionaries
- E. None of the above

(g) What is the result of attempting to run the following code?

```
g = [1,2,3,4]
g.append("the")
print(g)
g = g.append("quick brown fox")
print(g)
x = ("a", "b", "c")
x.append("d")
print(x)
```

Select any valid options:

- A. [1, 2, 3, 4] is printed
- B. "quick brown fox" is printed
- C. None is printed**
- D. An exception occurs**
- E. ("a", "b", "c", "d") is printed
- F. [1,2,3,4,"the", "quick brown fox"] is printed

(h) Which of these python statements evaluates to True?

- A. not True and not False
- B. True or False and False**
- C. True and False or False
- D. True and True and False
- E. not True or False

- (i) Read the following segments of code.

```
a = 10
b = 20
c = 30
if a == 10:
    if b > 20:
        print ("first")
elif b == 20:
    print("second")
if c > 35:
    print("third")
elif c <=35:
    if a < 50:
        print("fourth")
    if b > 0:
        print("fifth")
    else:
        print("sixth")
else:
    print("seventh")
```

Select all pieces of text that are printed when it is executed:

- A. **first** is printed
 - B. **second** is printed
 - C. **third** is printed
 - D. fourth is printed**
 - E. fifth is printed**
 - F. **sixth** is printed
 - G. **seventh** is printed
- (j) The reserved word **return** in Python serves which of the following purpose(s)?
- A. Do something again
 - B. Give a value back to the calling function**
 - C. Go back to the beginning of a loop
 - D. Retrieve a previously stored value
 - E. Interrupt the flow of execution**
 - F. None of the above
- (k) Which of these statements is a legal expression?
- A. "A" + "B"**
 - B. "A" - "B"
 - C. "A" * "B"
 - D. "A" / "B"

- (1) Which of the following is/are valid function names to use for writing text to a file when NOT using the CSV module?
- A. writeline
 - B. writelines
 - C. writerow
 - D. writerows
 - E. write**

2. (6 points)

Pretend you are the Python interpreter and the following code has been entered and executed:

```
myList = [6, 3, (True, False), "Original"]
one = myList[:]
two = myList
three = one + ["Hello"]
one[2] = one[0] + one[1]
myList[1] = three[1:4:2]
two[3] = two[3]*2
one = one[2:]
three[0] = one[0]
```

For each of the three new lists (one, two, three), which are aliases and which are copies?

What are the values of each of the following variables:

myList =

one =

two =

three =

<p>Solution: aliases = two copies = one, three</p>

```
myList = [6, [3, 'Original'], (True, False), 'OriginalOriginal']
one = [9, 'Original']
two = [6, [3, 'Original'], (True, False), 'OriginalOriginal']
three = [9, 3, (True, False), 'Original', 'Hello']
```

Grading: +2 points for correctly identifying each alias/copy.
+1 point for each correct list.

3. (5 points)

Write down what would be **printed** when the following code is executed.

```
def rec(n):
    if n<=1:
        return 1
    print(n)
    value= rec(n-1) + rec(n-2)
    return value
```

```
result = rec(5)
```

Solution: 5

```
4
3
2
2
3
2
```

Grading:

5 points if they get all numbers correct and write them vertically. 4 points if they get all numbers correct, but don't write them vertically. 3 points if they miss only 1 number. 2 points if they get at least 4 of the numbers right (and/or add 0's and 1's) 1 point if they sort of see what is happening, but the numbers are all wrong.

4. (6 points)

Fill in the blanks so that, when run, the code below will output the following:

```
>>> 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)

5. (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". Do not print out the "stop"! Hint: A while loop or recursion may be the way to implement this.

Solution:

```
def parrot( ):
    keepGoing = True
    while keepGoing:
        userStr = 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.

6. (10 points)

Write a function named `replaceAE` that takes in a string as a parameter. It should return a string where every 'a' in the original string is replaced with an 'e' and every 'e' is replaced with an 'a'. You may assume that all letters in the string are lowercase.

Example run:

```
>>> result = replaceAE('andy dreams of flying rockets' )
>>> result
'endy draems of flying rockats'
```

Solution:

```
def replaceAE(inStr):
    outStr = ""
    for letter in inStr:
        if letter == "a":
            outStr = outStr + "e"
        elif letter == "e":
            outStr = outStr + "a"
        else:
            outStr = outStr + letter

    return outStr
```

Grading: 2 points for a correct header.

2 points for looping through each letter in the input string

2 points for correctly swapping A's and E's.

2 points for correctly building up the new string.

2 points for returning the correct new string.

-2 points if they swap any other letter accidentally. -4 if they leave out other letters.

7. (10 points)

Write a function named `fileMin` that accepts the name of a file to open as a string parameter. The function should open the file and read in each line. Each line will contain a single integer number. Your function should find the minimum (smallest) number in the file and return it (as an integer). You may assume that the file name is correct and will exist, that the file will have at least one number in it, and that all numbers are valid integers.

For example, calling the function `fileMin` on file `test.txt` containing:

```
5
32
-1
0
```

would return the integer `-1`.

Solution:

```
def fileMin(fileName):
    f = open(fileName, "r")
    data = f.readlines()
    min = int(data[0])
    for item in data[1:]:
        num = int(item)
        if num < min:
            min=num
    f.close()
    return min
```

or:

```
def fileMin( fileName):
    f = open(fileName, "r")
    aLine = f.readline()
    min = int(aLine)
    aLine = f.readline()
    while( len(aLine) > 0):
        num = int(aLine)
        if min > num:
            min = num
        aLine = f.readline()
    f.close()
    return min
```

