

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 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
1. Vocabulary	9	
2. List Dictionary	4	
3. Multiple Choice	2	
4. Reading Functionals	6	
5. List Question	4	
6. Erica's Menu	10	
7. copyRect	10	
Total:	45	

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] argument

Solution: argument - A value provided to a function when the function is called. This value is assigned to the corresponding parameter in the function.

(b) [3 pts] iteration

Solution: Repeated execution of a set of statements.

(c) [3 pts] modulus operator

Solution: An operator, denoted with a percent sign (%), that works on integers and yields the remainder when one number is divided by another.

2. (4 points)

Write down exactly what the following code prints. If it generates an error, say which line generated the error and why.

```
aList = [12, 13, 89]
aDict = { print("Olivia Benson") : "test"}
aList.append(aDict)
stuff = aList[3] [aList.append("Meredith")]
print(stuff)
result = aList[-1]
print(result)
```

Solution:

Olivia Benson
test
Meredith

Grading:

+1 for Olivia Benson (only if the first printout) +1 for 'test' somewhere in the middle of anything else. +1 for Meredith (only if the last printout) +1 if all three lines are exactly correct.

-1 penalty for more than 3 lines.

3. (2 points)

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

(a) [1 pt] Examine the following code:

```
def main():
    counter = Counter()
    num = 0
    for x in range(0,100):
        incrementor(counter,num)
    return(counter.count,num)

def incrementor(c,num):
    c.count = c.count + 1
    num = num + 1

class Counter:
    def __init__(self):
        self.count = 0
```

aTuple = main()

What is stored in aTuple after the above code is run?

A. (0,100) **B. (100,0)** C. (100,100) D. (0,0) E. An error is raised

(b) [1 pt] Assume a class named FinalExam exists and the variable "aVar" points to an instance/object of type FinalExam. Which of the following lines of code will cause Python to throw an error?

A. FinalExam.color = "gold"
B. aVar.color = "navy"
C. aVar = FinalExam

D. `FinalExam.aVar.color = "white"`

4. (6 points)

Examine the following python code that uses functional programming.

```
a = [0,1,2,3,4,5,6]
b = map(lambda x: x*2, a)
c = filter(lambda x: x/2>3, b)
d = reduce(lambda x,y: x-y, c)
```

Tell us what each of the above variables refers to after the code is executed:

a: _____
b: _____
c: _____
d: _____

Solution:

answer:

```
a: [0, 1, 2, 3, 4, 5, 6]
b: [0, 2, 4, 6, 8, 10, 12]
c: [8, 10, 12]
d: -14
```

Grading: +1 point for a and b if fully correct.
+2 points for c and d (if fully correct.)

5. (4 points)

Beside the following code, write down exactly what would be printed after it is executed by the python interpreter.

```
oList = []
iList = [40]
for index in range(4):
    oList.append(iList)
    iList[0] = iList[0] + 20
print(oList)
```

Solution:

```
[[120], [120], [120], [120]]
```

+1 prints a list (of anything)

+1 List has proper number of items (4)

+1 Items in list are nested lists

+1 if integers inside nested lists are correct (120)

6. (10 points)

Oh No! I'm ordering food for Erica's party and the restaurant just sent a copy of their menu as a text file. It all sounds really good, except for the fact that she doesn't eat any vegetables. Write a function called `removeVegetables` to remove all the lines with vegetables from the menu. Your function will be given a string that represents the file name, and a list of strings (each of the vegetables to remove if found). Look for lines that do not contain any vegetables and write them out to another file called 'EricasMenu.txt'. You may assume that the vegetable will be separated from other words by a space.

Example: If the `menu.txt` file contains the following...

```
Tomato Soup, $4.56
Sushi, $15.89
Eggs with Onions, $1.23
Cheese Grits, $5.67
Beef Stew, $3.25
Tacos, $3.27
Spinach Ravioli, $9.75
```

After running `removeVegetables("menu.txt", ['Onions', 'Tomato', 'Spinach'])` the `EricasMenu.txt` file should contain:

Sushi, \$15.89
Cheese Grits, \$5.67
Beef Stew, \$3.25
Tacos, \$3.27

Solution:

```
def removeVegetables(fileName, listOfVeg):
    fIn = open(fileName, 'r')
    fOut= open('EricasMenu.txt', 'w')
    lines = fIn.readlines()
    fIn.close()
    for line in lines:
        keepLine = True
        for word in listOfVeg:
            if word in line:
                keepLine = False

        if keepLine == True:
            fOut.write(line)

    fOut.close()
```

Grading:

(10 points)

- +1 correct header
- +1 opens file using input parameter for reading
- +1 opens output file named "EricasMenu.txt" for writing
- +1 reads all the data from input file
- +2 correctly checks the line against the list of veggies.
(+1 of 2 if they only check one word, such as the first correctly)
- +2 writes correct lines to "EricasMenu.txt"
- +1 closes reading file
- +1 closes writing file

This page intentionally left blank so that you may answer the question on the previous page here.

7. (10 points)

Write a function named `copyRect` that accepts two pictures (references to picture objects, NOT string filenames) and four integers as parameters. The first picture is a source, and the second picture is the destination. The four integers are the X,Y position of a starting point (top left) of a rectangle and the second set of X,Y numbers represent the ending point (bottom right) of a rectangle.

Your function should copy a (solid) rectangle of pixels from the first (source) picture and draw them over the second picture (destination) at the location specified. For example, if the function is called as:

`copyRect(src,dest, 10,15, 35,40)`

You should copy a rectangle from the point (10,15) to (and including!) the point (35,40). You may assume that the picture objects that are passed into your function are large enough to completely contain your rectangle, and that the rectangle coordinates are valid. (All coordinates are inside the picture, and the top left corner is actually above and to the left of the bottom right corner.)

Your function should return `None`.

Some functions that might be helpful for this question:

- `getPixels(picture)`
- `getPixel(picture, x, y)`
- `getX(pixel)`
- `getY(pixel)`
- `getRed(pixel)`
- `setRed(pixel,value)`

Solution:

```
def copyRect(src,dest,x1,y1, x2,y2):
    for x in range(x1,x2+1):
        for y in range(y1,y2+1):
            inpix = getPixel(src, x,y)
            outPix = getPixel(dest, x,y)
            setRed(outPix, getRed(inPix) )
            setBlue(outPix,getBlue(inPix) )
            setGreen(outPix, getGreen(inPix))
```

Grading:

- +1 for correct header
- +2 for having code that iterates over (any) pixels
(either using `getPixel` with two for loops,
or by going over all pixels with `getPixels`)
- +1 for copying any pixels (regardless of where they are)

+3 for copying gthe (mostly) CORRECT pixels
(give +3 even if they missed the last row/column, see next)
+2 for correctly including the last row/column -- x2 and y2 coordinate
(x2+1, y2+1 if using range)
+1 if they correctly return None (deliberately, or by default)