

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
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. Flip	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] conditional statement

Solution: A statement that controls the flow of execution depending on some condition. In Python the keywords `if`, `elif`, and `else` are used for conditional statements.

- (b) [3 pts] integer division

Solution: integer division - In python 3.x, the `//` operator that divides two integers (int data type) the division will return an integer data type (int). Any remainder will be lost (truncated).

- (c) [3 pts] variable

Solution: A name (more specifically, an identifier) that refers to a value.

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("Test") : "Meredith"}
aList.append(aDict)
stuff = aList[3] [aList.append("Olivia Benson")]
print(stuff)
```

```
result = aList[-1]
print(result)
```

Solution:

```
Test
Meredith
Olivia Benson
```

Grading:

+1 for Test (only if the first printout) +1 for Meredith somewhere in the middle of anything else. +1 for Olivia Benson (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] Given the following code:

```
class Puppy:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def birthday(self, age):
        age = age + 1
        dogYears = self.age * 7
        print(self.name, "is", age, "years old! (Or", dogYears, "dog years!)"
```

```
spike = Puppy("Spike", 2)
spike.age = 4
spike.birthday(2)
```

What will be printed when the code is executed?

- A. Spike is 2 years old! (Or 14 dog years!)
- B. Spike is 3 years old! (Or 21 dog years!)
- C. Spike is 4 years old! (Or 21 dog years!)
- D. Spike is 3 years old! (Or 28 dog years!)**
- E. Spike is 4 years old! (Or 28 dog years!)

(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]
b = map(lambda x: x-2, a)
c = filter(lambda x: x>1, b)
d = reduce(lambda x,y: y, c)
```

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

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

Solution:

Answers:

[0, 1, 2, 3, 4, 5]

[-2, -1, 0, 1, 2, 3]

[2, 3]

3

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 = [50]
for index in range(5):
    oList.append(iList)
    iList[0] = iList[0] + 10
print(oList)
```

Solution:

```
[ [100], [100], [100], [100], [100]]
```

+1 prints a list (of anything)

+1 List has proper number of items (5)

+1 Items in list are nested lists

+1 if integers inside nested lists are correct (100)

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 flip that takes a picture object as a parameter. It should return a new picture that is an upside down version of its input. Another way of thinking about this is that the bottom row of the output picture should look exactly like the top row of the input picture. Your function should not modify the input picture.

Solution:

```
def flip(p):
    newP = makePicture( getWidth(p), getHeight(p)) # or copyPicture(p)
    for x in range( getWidth(p)):
        for y in range (getHeight(p)):
            pix = getPixel(p,x,y)
            newPix = getPixel(newP, x, getHeight(p) -1 -y )
            oldColor = getRGB(pix) #Could also do red/green/blue separately
            setRGB(newPix, oldColor)
    return newP
```

Grading:

- +1 correct header
- +2 making new picture same size as old picture
- +1 iterating through all of the pixels
- +1 getting the color(s) from the old pixel
- +1 putting the same color(s) into (any) new pixel
- +3 calculating CORRECT new pixel to put the color into.
- +1 returning the new picture.