

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. Write all answers in the solution boxes unless directed otherwise. 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. Multiple Choice | 16 | |
| 2. removeDuplicates | 3 | |
| 3. Big O Mystery | 5 | |
| 4. BigO Graph | 10 | |
| 5. Searching | 8 | |
| 6. Objective Cars | 5 | |
| 7. Point Objects | 5 | |
| 8. Weather | 10 | |
| 9. Functional Practice | 7 | |
| 10. Functional: Sum It Up | 4 | |
| 11. Functional Temps | 10 | |
| 12. Functional Blanks | 7 | |
| Total: | 90 | |

1. (16 points)

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

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

```
def foo(aList):
    for i in range( len(aList) // 2 - 1 ):
        for j in range(len(aList) - 2):
            aList[i-j] = 0
```

Let N be the length of `aList`. If your "unit of work" is assigning a zero to a list element `aList[i-j]`, consider how the number of units of work increase as N increases. What is the Big O complexity class of the function `foo`?

A. $O(N)$ B. $O(N^2)$ C. $O(2^N)$ D. $O(\log N)$ E. $O(N \log N)$

(b) [1 pt] Which of the following has the smallest Big O complexity?

A. Binary Search B. Dictionary Lookup C. Merge Sort D. Insertion Sort

(c) [1 pt] Which of the following will return `[2,4,6,8,10]`?

- A. `map(lambda x: x+1, range(0,10,2))`
 B. `filter(lambda x: x%2 == 1, range(2,11))`
 C. `reduce(lambda x,y: x**y, range(100))`
 D. `filter(lambda x: x%2 == 0, range(1,11,2))`
 E. `reduce (lambda x,y: x+y, [2,4,6,8,10])`
 F. `map(lambda x: x+1, range(1,11,2))`
 G. `map(lambda x: x%2 == 0, [2,4,6,8,10])`
- (d) [1 pt] What would `list(map(lambda x: not x, [True, True, False, True, False]))` return?
 A. `[False, False, True, False, True]`
 B. `[None, None, None, None, None]`
 C. `[True, True, False, True, False]`
 D. `[False, False, False, False, False]`
- (e) [1 pt] What would `list(map(lambda x: x<10,[5, 10, 3, 20, 15]))` return?
 A. `[5, 3]`
 B. `[5, 10, 3]`
 C. `[True, True, True, False, False]`
 D. `[True, False, True, False, False]`
- (f) [1 pt] Which of the following expressions returns `[1,2,3,4,1]`?
 A. `list(map(lambda x: x + 1, [0, 1, 2, 3, 4, 0]))`
 B. `list(map(lambda x: x % 4 + 1, [12, 13, 6, 19, 0]))`
 C. `list(map(lambda x: x // 2, [2, 4, 7, 9.5, 1]))`
 D. `list(map(lambda x: x + x, [0.5, 1.5, 2.5, 3.5, 4.5]))`
- (g) [1 pt] What is printed when the following code is run?

```
def f1(n):
    return n%2==0
print(filter(f1, [1,2,3,4,5]))
```

 A. `[2, 4]`
 B. `[False,True,False,True,False]`
 C. `[1, 3, 5]`
 D. `[True,False,True,False,True]`
 E. `False`
- (h) [1 pt] Which of the following would evaluate to `[3,6,9]`
 A. `map(lambda x:x, [3,6,9])`
 B. `filter(lambda x:(x+1)%3==0, [2,5,8])`
 C. `filter(lambda x:x==x-1, [4,7,10])`
 D. A and C above

- E. B and C above
- F. A, B, and C above

Examine the following code which defines a class and then creates an instance of the object, then answer the following questions about it:

(i) [2 pts]

```
class Costume:
    numberOfCostumes = 0

    def __init__(self, type):
        self.type = type

    def printType(self):
        print("I am wearing a {} costume".format(self.type))

ghost = Costume("Ghost")
```

Look at the following pieces of code and the statement that describes what they do. Select the one that is true.

- A. `ghost.numberOfCostumes = 1` updates the **class** variable `numberOfCostumes`
- B. `ghost.type = 1` updates the **instance** variable `type`
- C. `Costume.numberOfCostumes = 1` updates the **instance** variable `numberOfCostumes`
- D. instance and class variables are the same thing
- E. All of the above
- F. None of the above

Which line of code below correctly calls the `printType` method?

- A. `ghost.printType(ghost)`
- B. `ghost.printType()`
- C. `self.printType(ghost)`
- D. `ghost.printType(self)`
- E. `Costume.printType(self.ghost)`

(j) [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"`

(k) [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

- (l) [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

- (m) [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!)

2. (3 points)

You are an intern at a company and they have asked you and a friend to write a function that removes duplicate strings from a list. (You do not need to preserve the order of the entries in the list.) You each write a function to remove duplicates.

Function A looks like this:

```
def removeDuplicatesA(aList):
    newDict = {}
    for item in aList:
        if item not in newDict:
            newDict[item] = 0

    return newDict.keys()
```

Function B looks like this:

```
def removeDuplicatesB(aList):
    newList = []
    for item in aList:
        if item not in newList:
            newList.append(item)

    return newList
```

Which function will work faster on a list with a very large number of strings? Why?

3. (5 points)

You are an intern at BigProgrammingCompany Inc, and your first job is to figure out the code left for you by a previous intern. Your boss gives you the following code and wants to know what it does and how efficient it is:

```
def function(key, aList):
    a = -1
    for i in aList:
        a = a + 1
        if i==key:
            return a
    return None
```

Describe using a sentence or two what this function does. Be sure to clearly specify what it returns in all cases.

If the number of items in the aList is N, what is the BigO time complexity class of this function (average case)? Use the comparison operator as your unit of "work".

4. (10 points)

Draw a single graph with a line for each of the four main Big O complexity classes we have learned about. Label both the X and Y axes appropriately. For each line, label it with the Big O complexity class it represents, and also give the name of an algorithm that falls in that complexity class.

5. (8 points)

(a) [2 pts] Here is a sequence of numbers: 1, 3, 5, 7, 8, 10, 15

If you were to perform a linear search on the above sequence, looking for the number '8', in what order does the computer search through the items in the sequence? Write them below.

(b) [2 pts] If you were to perform a binary search on the original sequence for the number 8, in what order does the computer search through the items in the sequence? Write each item that is examined in the order that it is examined below.

(c) [2 pts] What advantage does the binary search give us?

(d) [2 pts] Why might we use a linear search instead of a binary search?

6. (5 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 E2:

    def __init__(self, paint, make):
        self.paint = paint
        self.make = make

    def printMyCar(self):
        print("I'm driving a {0} {1}, bro.".format(self.paint,self.make))

car1 = E2('red', 'Mercedes')
car2 = E2('blue', 'Audi')

car1.paint = car2.make
car2.make = car1.make

car1.printMyCar()
car2.printMyCar()
```

7. (5 points)

Examine the following class definition of a Point and the seven lines of code that use Points:

```
class Point:
    def __init__(self,x=0,y=0):
        self.x = x
        self.y = y

    def getDistance(self):
        from math import sqrt
        dist = sqrt( self.x*self.x + self.y*self.y)
        return(dist)

    def __eq__(self,other):
        if self.x == other.x:
            return True
        else:
            return False

p1 = Point()
p2 = Point(3,4)
```

```
p3 = Point(3,3)

v1 = p1.x
v2 = p2.y
v3 = p2 == p3
v4 = p2.getDistance()
```

What are the values in the 4 variables (v1,v2,v3,v4) after this code executes?

8. (10 points)

Examine the following code:

```
class GA:
    population = 9800000
    weather = "hot"

    def __init__(self, loc):
        print("Georgia")
        self.city = loc
        if self.city == "Atlanta":
            self.Atlanta()
        elif self.city == "Athens":
            print("Bad!")
            GA.weather = "cold"

    def Atlanta(self):
        self.weather = "humid"

place = GA("Atlanta")
place2 = GA("Athens")
print("The weather in {0} is {1}.".format(place.city, place.weather))
print("The weather in {0} is {1}.".format(place2.city, GA.weather))
```

What is printed on the screen as the code above is executed?

Different identifiers above are used to name a **class**, **class variable**, **object**, and **object variable**. For each of the following identifiers, write what it was used to name:

GA -

population -

city -

place -

9. (7 points)

Examine the following python code that uses functional programming.

```
a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
b = filter(lambda x: x % 2 == 0, a)
c = map(lambda x: x**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: _____

10. (4 points)

Entirely re-write the following function using functional programming instead of iteration. You may **not** use iteration or recursion.

```
def sumItUp(aList):
    num = 0
    for i in aList:
        num = num+i

    return num
```

11. (10 points)

You are working for a meteorologist and need to use Python's functional programming to manipulate the following list of temperatures (in Fahrenheit) that represents the high temperature for each day in a particular week. You may only use map/filter/reduce and lambda functions.

```
tempF = [86, 91, 82, 88, 90, 87, 85, 72]
```

(a) [3 pts] Write a line of code that will filter out temperatures under 85 degrees F, keeping only temps 85 or higher, and store the resulting list as highTempsF.

```
highTempsF = _____
```

(b) [3 pts] Write a line of code that will convert the highTempsF into Celsius and store the resulting list as highTempsC. Celsius = $5/9 * (\text{Fahrenheit} - 32)$.

```
highTempsC = _____
```

(c) [4 pts] Write a line of code that will sum the Celsius high temperatures, and then calculate the average Celsius high temp, storing it in avgHighC. You MAY NOT use the sum() function, but may use the len() function.

```
avgHighC = _____
```

Fill in the following blanks so that the python code is correct:

```
originalList = [-4,-3,-2,-1,1,2,3,4]
```

1) all elements are multiplied together to produce the product of all values in originalList

12. (7 points)

```
576
```

```
>>> _____
```

2) all elements in originalList are doubled

```
>>> list(_____ (lambda x: 2*x, originalList))
[-8, -6, -4, -2, 2, 4, 6, 8]
```

3) only positive elements are retained

```
>>> list(_____ (lambda x: x > 0, originalList))
[1, 2, 3, 4]
```

4) only multiples of 3 and multiples of 5 are returned

```
>>> list(filter(_____, originalList))
[-3, 3]
```

5) all elements in the list are printed out (A list of None's is returned)

```
>>> list(map(_____, originalList))
-4
-3
-2
-1
1
2
3
4
[None, None, None, None, None, None, None, None]
```

6) the expression sums all negative, odd numbers (note these numbers must be both odd and negative)

```
>>> reduce(_____,
           filter(_____, originalList)
           )
-4
```