

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 6 questions on 10 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. Multiple Choice	5	
3. Ready to Add	6	
4. 2D List Question	12	
5. Dewey Dictionary Work	5	
6. Objective Car	8	
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] 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] parameter

Solution: parameter - A name used inside a function to refer to the value passed as an argument.

(c) [3 pts] instantiate

Solution: To create an instance of a class, and to run its initializer.

2. (5 points)

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

(a) [1 pt] How do you typically take notes in this class?

- A. Using pen/pencil and paper
- B. On my computer in IDLE/Python
- C. On a computer/tablet (typing text, handwriting) without executing python
- D. I don't take notes, I just watch and learn.

Examine the following code when answering the following four multiple choice questions:

```
midnightSnack={"Candy": ["Gummy Worms", "Tootsie Rolls", "Twix", "Reeses"],
               "Soda": ["Pepsi", "Tab", "Mellow Yellow"],
               "Snacks":["Cheetos", "Onion Rings", "Fritos", "Trail Mix"]  }
candyList=midnightSnack["Candy"]
goToSnack=candyList[4]
```

- (b) [1 pt] What is candyList?
- A. ["Gummy Worms"]
 - B. ["Gummy Worms", "Tootsie Rolls", "Twix", "Reeses"]**
 - C. []
 - D. An Index Error is raised
- (c) [1 pt] What is goToSnack?
- A. "Reeses"
 - B. ["Reeses"]
 - C. ["Gummy Worms", "Tootsie Rolls", "Twix", "Reeses"]
 - D. An Index Error is raised**
- (d) [1 pt] Which of the following pieces of code would successfully create a list of all the values in midnightSnack?
- A. midnightSnack.values()
 - B. midnightSnack.items()
 - C. list(midnightSnack.values())
 - D. alist=[]
alist.extend(midnightSnack.values())
 - E. both c and d are correct**
- (e) [1 pt] Which one of the following statements would create an error?
- A. midnightSnack[("Meals")]= ["Pizza", "BBQ Chicken Wings", "Pancakes"]
 - B. midnightSnack.sort()**
 - C. aName="Bojangles"
midnightSnack[aName]= aName
 - D. midnightSnack.keys()
 - E. both A and B
 - F. both C and D

3. (6 points)

Given that the files: main.py, dumbmath.py, add.py, and multiply.py are all in the same directory, what is printed to the screen when main.py is run? If a line will result in an error, print "error" in its place and continue on. Draw a box around your answer!

```
#main.py
import dumbmath
from add import *

print(dumbmath.add(a, b))
print(dumbmath.multiply(2, 3) + dumbmath.multiply(a, b))
print(multiply(2, 3))
print(add.add(5, 3))
```

```
# add.py
a = 3
b = 4
from multiply import multiply
print("Ready to add!")
def add(a, b):
    return a + b
```

```
# dumbmath.py

print("halp!")

def add(a, b):
    return "a" + "b"

def multiply(a, b):
    return "a" * b
```

```
# multiply.py
def multiply(a, b):
    return a * b
```

Solution:

```
halp!  
Ready to add!  
ab  
aaaaaaa  
6  
***ERROR***
```

Grading:

+1 for each correct line in correct position/order (6 total)
-1 for any extra lines (over 6).

4. (12 points)

(a) [6 pts] Consider the following code:

```
def mysteryFunc(x,y):
    outputList = []
    for i in range(1,x+1):
        temp = []
        for j in range(1,y+1):
            temp.append(i*j)
        outputList.append(temp)
    return outputList
```

a = mysteryFunc(4,4)

b = mysteryFunc(6,1)

Write what a and b point to after the code is executed:

Solution:

```
a = [[1, 2, 3, 4], [2, 4, 6, 8], [3, 6, 9, 12], [4, 8, 12, 16]]
b = [[1], [2], [3], [4], [5], [6]]
```

GRADING:

+2 correct number of nested lists

+2 correct number of elements within nested lists

+2 correct value per element

(b) [6 pts] Consider the following code:

```
def aFunc(aList):
    newList = []
    j = 0
    for i in range(len(aList[0])):
        temp = []
        for j in range(len(aList)):
            temp.append(aList[j][i])
        newList.append(temp)
    return newList
```

e = aFunc([["a","b","c"], [1,2,3], [4,5,6], [7,8,9], [10,11,12]])

f = aFunc([["d","e","f","g","h"], [99,98,97,96,95], [94,93,92,91,90]])

Write down what e and f point at after this code is executed:

Solution:

e = [['a', 1, 4, 7, 10], ['b', 2, 5, 8, 11], ['c', 3, 6, 9, 12]]

f = [['d', 99, 94], ['e', 98, 93], ['f', 97, 92], ['g', 96, 91], ['h', 95, 90]]

GRADING:

+2 for correct number of nested lists

+2 for correct number of elements per nested list

+2 for correct value per element

5. (5 points)

Examine this code and write down what it would print when executed:

```
def deweyDecimal(subject, country):
    dewey = {}
    dewey = {"General":000, "Philosophy":100, "Religion": 200,
             "Social Sciences":300, "Language":400}
    country = country.lower()
    try:
        dewey["Natural Sciences"]=500
        print(dewey[country])
    except:
        print("This test is easy!")
    number = dewey[subject]
    if country == "europe":
        number = str(number) + "0.5"
        dewey[subject]=number
    elif country == None:
        number = str(number) + "0.6"
        dewey[subject]=number
    numbers = []
    for item in dewey.keys():
        numbers.append(float(dewey[item]))
    numbers = numbers.sort()
    print(numbers)
    print(dewey)
```

```
deweyDecimal("Religion","Europe")
```

Solution:

This test is easy! None 'General works': 0, 'Natural Science': 500, 'Social sciences': 300, 'Language': 400, 'Philosophy and psychology': 100, 'Religion': "2000.5" **Order of items does not matter

Grading: 1 point for "This test is easy!" 1 point for "None" 1 point for any dictionary. -1 for each extra line 1 point for getting types of all the values correct (integer vs. strings) 1 point for adding Natural Sciences to the dictionary

6. (8 points)

Create a new class called Car which takes in two parameters, the number of miles the car currently has as a float (miles) and the color of the car as a string (color). In addition to the initializer, the class should have two methods:

drive - takes in the number of miles to drive the car as a parameter and updates the Car object's miles driven. Should also print "Vroom!" every time the method is called.

newPaint - takes in no parameters. Asks the user what color they would like to paint their car and updates the car's color accordingly

Solution:

```
class Car:
    def __init__(self, miles, color):
        self.miles = miles
        self.color = color

    def drive(self, milesDriven):
        print('Vroom!')
        self.miles = self.miles + milesDriven

    def newPaint(self):
        newColor = input("What color would you like to paint your car?")
        self.color = newColor
```

Grading:

```
+1 for Class Car: (or Class Car():)\
+1 for __init__ with 3 parameters
+2 for initializing self.miles and self.color correctly.
+1 for def drive(self, someParameter):
+1 for adding someParameter to self.miles.
+1 for newPaint(self)
+1 for updating self.color correctly.
```

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!