

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 12 questions on 14 pages including the title page. Please check to make sure all pages are included. You will have 100 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. Types of Errors	9	
2. Expressions	26	
3. Iffy Letters	4	
4. Find Price	8	
5. Three Foos	6	
6. Fix Recursion	3	
7. Print vs. Return	5	
8. Range and Modulo	5	
9. Body Mass Index	9	
10. Triangle	10	
11. CountQs	8	
12. Base Conversions	5	
Total:	98	

1. (9 points)

List the three types of errors that a python program may contain, and give a concise 1-2 sentence definition of each. Be brief, and to the point.

Solution: Syntax Error - An error in a program that makes it impossible to parse – and therefore impossible to interpret.

Semantic Error - An error (in code) that leads to unexpected behavior. The program functions correctly (does what the code says) but the code does not actually perform the action that the programmer intended.

Runtime Error - An error raised by the python runtime while the program is executing if something goes wrong. For example, a divide by zero error.

Grading: +1 point for each correct type of error, +2 points for each correct definition.

2. (26 points)

Pretend you are the python interpreter. Evaluate each of the expressions below. Write down the value that they evaluate to, and the type of that value in the provided columns. If the expression is not valid python syntax, or will throw an exception, simply write "Error". The first line has been provided as an example.

Expression	Result	Type
6+5.5	11.5	float
12+3		
"3*5"		
3+4==2		
True and (3 < 2)		
4+6//5		
int(4.5) / 2		
"{0:.2f} percent".format(6.312)		
True or (4==3)		
"End" + "Program"		
"Hello" * 3		
(6-3)**2-4		
2+6/3.0		
3%2		

Solution:	Expression	Result	Type
	6+5.5	11.5	float
	12+3	15	int
	"3*5"	'3*5'	str
	3+4==2	False	bool
	True and (3 < 2)	False	bool
	4+6//5	5	int
	int(4.5) / 2	2.0	float
	"0:.2f percent".format(6.312)	'6.31 percent'	str
	True or (4==3)	True	bool
	"End" + "Program"	'EndProgram'	str
	"Hello" * 3	'HelloHelloHello'	str
	(6-3)**2-4	5	int
	2+6/3.0	4.0	float
3%2	1	int	

Grading: +1 point for each correct answer.

3. (4 points)

What does the python interpreter print when the following code is executed?

```
def mathFunc(x):
    if x%2 == 1:
        print("a")
    else:
        print("b")
    if x/2 <= 5.0:
        print("c")
    if x*1.5 <= 25:
        print("d")
    elif x*1.5 <= 20:
        print("e")
    if x%4 > 2.0:
        print("f")
    elif x%4 >= 2:
        print("g")
    elif x%4 > 1:
        print("h")
```

```
mathFunc(10)
```

Solution: b c d g

+1 point for each correct letter. -1 point for each incorrect letter.

-1 point if they don't draw them vertically.

4. (8 points)

Pretend you are the Python interpreter and the following code has been entered and executed. Write down exactly what would be printed in the shell!

```
def findPrice(item, aNum):
    if item == "grapes" or item=="Grapes":
        print( "Grapes=$2.99. Number={}".format(aNum) )
        return 2.99
        print( "Grapes are Awesome!")
    elif item == "peaches" or item == "PEACHES" and aNum != 0:
        if aNum > 10:
            return "{}! too Much!".format(aNum)
            print( "Peaches are $199" )
            return 1.99
        elif item == "apples" or item=="Apples":
            print("Apples are $1.00")
            return 1.00
        elif item == "cherries" and aNum >= 0:
            print("Cherries are $3.00")

            if aNum < 2:
                print( "2 bags")
            elif aNum > 2 and aNum < 10:
                print( "Good Number")
            else:
                print("Lots of bags")

    else:
        print("No Such Product")

d = findPrice("Apples", 1)
print( findPrice("cherries", d) )
findPrice("Grapes", 2)
print( findPrice("peaches", 14) )
x = findPrice("durian", 1)
print(x)
```

Solution:

Apples are \$1.00
Cherries are \$3.00

2 bags

None

Grapes=\$2.99. Number=2

14! too Much!

No Such Product

None

Grading:

+1 for each correct line.

-1 for each extra/incorrect item, or each correct item out of order.

-1 (only once) if they put quotes around any output

5. (6 points)

Given the following functions, write what they print to the screen when they are executed. If the function produces an error, write ERROR and draw an arrow to the line that caused the error.

```
(a) def foo1(a,b,c):
    b=a++
    if b>a:
        print(True)
    else:
        return b<a
foo1(1,2,3)
```

Solution:

ERROR - Line 1 (a++)

Grading: 1 point for Error, 1 point for knowing where.

```
(b) def foo2(a,b,c):
    if b>c:
        if c>0:
            print("Yes!")
        elif b>0:
            b=c-b
        elif a<0:
            print("Uhm")
        if b!=c:
            print (b)
        else:
            print("No.")
    else:
        print("Finish!")
foo2(-1,8,3)
```

Solution:

Yes!

8

Grading: 1 point for each correct line, -1 point for incorrect lines.

```
(c) def foo3(a,b):
    if b==0:
        return a
    else:
        return foo3(b,a%b)
print( foo3(10,6) )
```

Solution:

2

Grading: 2 points for the correct number (2)

6. (3 points)

The following recursive function is supposed to print "Cute Puppy! X" where X is a number that starts at N and moves down to zero decrementing by one each time it prints the string. This code has a problem that you have to fix. Indicate what the problem is and what code you would remove, add, or change to fix it.

```
def loopOfDogs(N):  
    puppies = N  
    print("Cute Puppy! {}".format(N) )  
    loopOfDogs(puppies-1)
```

Solution: Answer: This function suffers from infinite recursion, and will keep counting down past zero and never stop!

Adding the following lines at the top would fix this:

```
if N < 0:  
    return
```

Grading:

3 points if they understand what the problem is, and give a valid fix. 2 points if they describe the problem, but don't quite get the fix right. 1 point if they mention infinite recursion, but don't give the correct answer. 0 points for anything else/less.

7. (5 points)

Complete each statement below by filling in the blank with the appropriate letter from these options:

- A. return
- B. print
- C. both print and return
- D. neither print nor return

1. _____ can be used while defining the code for a function.
2. _____ can be used outside a function definition.
3. _____ can be used to terminate execution inside a function.
4. _____ will always show the result on screen.
5. _____ will assure you an A in your CS class.

Solution: Grading: +1 for each correct answer.

C

B

A

B

The 'correct' answer is D, but we give them a free point for anything.

8. (5 points)

What does the following code PRINT when executed?

```
for n in range(1,10):
    if (n%3 == 0) or (n%5 == 0):
        print("oak")
    elif (n%4 == 0):
        print("elm")
    if (n==5):
        print("birch")
```

Solution:

oak

elm

oak

birch

oak

elm

oak

Grading: +5 for all correct. Otherwise, -1 for each missing or extra item. (zero minimum)

9. (9 points)

Write a function named `bodyMassIndex`. Prompt the user to enter their weight in kilograms and height in meters. Make sure to use a descriptive prompt for each value. You may assume that the user will enter valid floating point numbers (e.g. 44.8). Your function should calculate the Body Mass Index of that person using the following formula: $BMI = Weight/Height^2$

Then print the result. The result should be formatted to one significant digit after the decimal point. For example, your function should print something like: "Your BMI is 18.5".

Example run:

```
>>> bodyMassIndex()
Please enter your weight in kg: 79.3
Please enter your height in meters: 1.82
Your BMI is 23.9
>>>
```

Solution:

```
def bodyMassIndex():
    weightStr = input(Please enter your weight in kilograms: )
    heightStr = input(Please inters your height in meters: )
    weight = float(weightStr)
    height = float(heightStr)
    bmi = weight/(height*height) #or weight / (height**2)
    print( 'Your BMI is {0:.1f}'.format( bmi) )
```

Grading: 1 point for a correct header.

1 point for getting weight

1 point for getting height

2 points for correctly converting weight/height to float

2 points for correct math.

2 points for correct output formatting.

10. (10 points)

Write a function named `triangle`. This function will cause your robot to move/draw an equilateral triangle. The function will accept one integer parameter, representing the length of each side of the triangle in inches.

You may assume that your robot moves one inch per second (forward or backwards), and turns 360 degrees in a time of 3 seconds (when traveling at full speed). When the function finishes running, it should beep in celebration and return the total distance travelled by the robot, in inches (excluding turns, as the robot does not “travel” while rotating).

You may assume that `from Myro import *` has already been executed, and that the correct `init(...)` call has already been made, so your robot is ready to execute myro robot movement functions.

Solution:

```
def triangle( numInches):  
    for i in range(3):  
        forward(1,numInches)  
        turnLeft(1,1)    #Turn 120 degrees  
    beep(1,800)  
    return 3*numInches
```

Grading: 2 points for a correct header with parameter

2 points for iterating 3 times (for, while, recursion, or just repeating the code!)

2 points for traveling one/each side length correctly.

(-1 if they get it right once but mess up the others)

2 points for turning one/each 120 degree corner exactly 120 degrees

(1 second at full speed)

(-1 if they get it right once but mess up the others)

1 point for a beep at the end.

1 point for returning the correct answer.

11. (8 points)

Write a function called **countQs** that accepts a string as a parameter. Count the total number of Q's in the string (both capital and lower case q's!) and return the count as an integer. If the string has no Q's (for example, if it is empty, or just doesn't have any) return a zero.

Example test case:

```
>>>ans = countQs("The Quick brown quack")
>>>print(ans)
2
>>>
```

Solution:

```
def countQs(aString):
    counter = 0
    for letter in aString:
        if letter == "Q" or letter=="q":
            counter = counter + 1
    return counter
```

Grading: 1 point for a correct header.

1 point for starting the count at zero.

2 points for traversing through all letters in the string.

2 points for checking for both Q and q.

1 point for returning an integer.

1 point for returning the correct answer.

12. (5 points)

For each of the following base conversion questions, give a brief answer:

(a) [1 pt] Convert the decimal number 17 into binary:

Solution: 10001

(b) [1 pt] Convert the decimal number 218 into binary:

Solution: 11011010

(c) [1 pt] Convert the hexadecimal number 2F into decimal:

Solution: $2 * 16 + F (15) * 1 = 32 + 15 = 47$

(d) [1 pt] Convert the binary number 10011111 into decimal:

Solution: 159

(e) [1 pt] Convert the decimal number 9 into hexadecimal:

Solution: 9

The rest of 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!