

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 11 questions on 11 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. Expressions	10	
3. Print v. Return	3	
4. Find the Error	3	
5. Hulk Strings	4	
6. Iffy Letters	4	
7. In the Cups	5	
8. For2While	6	
9. Leaky Pipes	6	
10. Waffles and Pancakes	5	
11. FizzBuzz	6	
Total:	61	

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] algorithm

**Solution:** A finite set of unambiguous instructions performed in a prescribed sequence to achieve a goal.

(b) [3 pts] int

**Solution:** A Python data type which stores positive and negative whole numbers.

(c) [3 pts] operator

**Solution:** Operators are special tokens that represent computations like addition, multiplication and division.

## 2. (10 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
<code>1 + 2 * 3</code>	7	int
<code>6 % 4 / 3</code>		
<code>"a" + "b" * 2</code>		
<code>len( range(4,5) )</code>		
<code>3.5 * 2</code>		
<code>4**(3.0/2)</code>		

**Solution:**

Expression	Result	Type
<code>1 + 2 * 3</code>	7	int
<code>6 % 4 / 3</code>	0.66666666	float
<code>"a" + "b" * 2</code>	"abb"	str
<code>len( range(4,5) )</code>	1	int
<code>3.5 * 2</code>	7.0	float
<code>4**(3.0/2)</code>	8.0	float

Grading: +1 point for each correct answer.

## 3. (3 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 return1():
    print(1)
    return 1

def someFunc():

    if return1() == 1:
        print('hi')

    if return1() == 0:
        print('bye')

someFunc()
```

**Solution:**

```
1  
hi  
1
```

**Grading:**

+1 for each correct line.  
-1 for each extra/incorrect item.  
-1 if they put quotes around the "hi"

## 4. (3 points)

Find the Error: The following code contains a statement that will cause a runtime error. Circle the line and explain what's wrong.

```
e = "2.718"  
pi = 3.14  
pie = str(pi) + e  
print( int(e) )  
print( int(pi) )  
print( pie )
```

**Solution:** Line 4 ( `print( int(e))` ) contains the error. You cannot convert a string into an integer if it's a floating point number. 1 point for identifying the line, 2 points for explaining what is wrong.

## 5. (4 points)

Examine the following code and write what it would print beside it:

```
phrase = "HULK MAD"  
phrase.lower()  
phrase.capitalize()  
cutPhrase = phrase[:5]  
newPhrase = ""  
  
for char in phrase[5:]:  
    if char == "m":  
        newPhrase = newPhrase + cutPhrase + "Hungry"  
    elif char == "d":
```

```
        newPhrase = newPhrase + cutPhrase + "Angry"
    elif char == "A":
        newPhrase = newPhrase + cutPhrase + "Smash"

print(newPhrase)
```

**Solution:**

HULK Smash

Grading:

- +2 for "HULK" in uppercase (+1 only if lowercase)
- +2 for "Smash" (+1 only if case incorrect)
- 1 for each other word included
- 1 if HULK Smash is out of order (Smash HULK)

## 6. (4 points)

Write what a python interpreter would print when the following code is executed to the right of the code.

```
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.

7. (5 points)

Beside the code, write down what the following code would print:

```
for x in range(1,6):
    if x < 4:
        print("cup")
    elif x > 4 and x <= 6:
        print("cake")
    else:
        print("cuppy")
```

**Solution:**

cup  
cup  
cup  
cuppy  
cake

Grading: +1 for each correct line  
(-1 for each incorrect addition or if they are not on different lines)

8. (6 points)

Examine the following code:

```
aStr = "This is only a test!"
for char in aStr:
    print(char*2)
```

Now, the "f" key on your keyboard has failed, so that you can not use the "for" keyword. Re-write all of the code above using a while loop instead of a for loop. You must use a while loop, and may not use the "for" keyword. (Your solution may not use the letter 'f'.)

**Solution:**

```
aStr = "This is only a test!"
index = 0
while index < len(aStr):
    char = aStr[index]
    print(char*2)
    index = index + 1
```

**Grading:**

- +1 for using an index variable
- +1 for starting index variable at zero
- +1 for ending when index is no longer less than the length of aStr
- +1 for incrementing the index variable
- +1 for correctly indexing into aStr to get char.
- +1 for printing the character twice (\*2)

9. (6 points)

Leaky Pipes - What is printed by the following function if it is called with an input of 12?

```
def leakyPipes(n):
    if (n > 0):
        if (n % 4 == 0):
            print "drip %d" % n
            leakyPipes(n-3)
        if (n % 3 == 0):
            print "drop %d" % n

leakyPipes( 12 )
```

**Solution:**

```
drip 12
drop 9
drop 12
```

2 points for each line.

10. (5 points)

Execute the following piece of code as if you were the python interpreter. Next to the code, write down exactly what would be printed.

```
def mysteryFun1(x):
    print("I love food" + "!" * x)
    return "waffles"
    print("pancakes")

mystery = mysteryFun1(5)
print("I ate" )
print(mystery*3)
print("I'm so full.")
```

**Solution:**



```
I love food!!!!  
I ate  
waffleswaffleswaffles  
I'm so full.
```

Grading Key:

```
+1 for printing I love food  
+1 for correct number of exclamation points: 5  
+1 for printing "I ate" AND "I'm so Full"  
+1 for printing correct number of waffles:3  
+1 for printing any waffles  
-1 for any extra lines.
```

11. (6 points)

Write a function named `fizzbuzz` that takes in a single integer parameter. Your function should print out the numbers from 1 to the value of the parameter you passed in, inclusive. Exceptions: If the number is a multiple of three, you should print out the word “Fizz” **instead** of the number. If the number is a multiple of five, you should print out “Buzz” **instead** of the number. If the number is a multiple of **both** three and five, you should print out “FizzBuzz” **instead** of the number. You may assume the integer parameter is non-negative. (If it is zero, your function should do nothing.)

**Example run:**

```
>>> fizzbuzz(5)
1
2
Fizz
4
Buzz
>>>
```

**Solution:**

```
def fizzbuzz(num):
    for i in range(1,num+1):
        if i%3==0 and i%5==0:
            print("FizzBuzz")
        elif i%3 == 0:
            print("Fizz")
        elif i%5 == 0:
            print("Buzz")
        else:
            print(i)
```

Grading:

- +1: Correct function header
- +1: Correct loop (Loops from 1 to num+1)
- +1: Prints FizzBuzz for multiples of 3 AND 5 (and no others!)
- +1: Prints Buzz for multiples of 5 (and no others!)
- +1: Prints Fizz for multiples of 3 (and no others!)
- +1: Prints out num for numbers that are not multiples of 3 and 5.

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!