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 8 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. Base Conversions	5	
3. Expressions	16	
4. Multiple Choice	9	
5. Recursive Function	9	
6. Three Foos	6	
7. isNegative	5	
8. Square	10	
Total:	69	

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] block

   Solution: One or more program statements that share the same level of indentation.

   (b) [3 pts] evaluate

   Solution: To simplify an expression by performing the operations in order to yield a single value.

   (c) [3 pts] semantic error

   Solution: 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.

2. *(5 points)*
   For each of the following base conversion questions, give a brief answer:

   (a) [1 pt] Convert the decimal number 9 into binary:

   Solution: 1001

   (b) [1 pt] Convert the decimal number 172 into binary:

Solution: 10101100

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

Solution: 42

(d) [1 pt] Convert the binary number 110101 into decimal:

Solution: 53

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

Solution: 9

3. *(16 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**
1 + 2 * 3	7	int
7 % 4		
`"cat" + "dog" * 2`		
len(range(4,12,2))		
`print("Hi")`		
9 + 5 / 2		
2**(10//3)		
2 + 5 = 7		
True and 3 < 2		

Solution:

Expression	**Result**	**Type**
1 + 2 * 3	7	int
7 % 4	3	int
”cat” + ”dog” * 2	’catdogdog’	str
len(range(4,12,2))	4	int
print(”Hi”)	None	NoneType
9 + 5 / 2	11.5	float
2**(10//3)	8	int
2+5 = 7	Error	Error
True and 3 < 2	False	bool(ean)

Grading: +1 point for each correct answer.

4. *(9 points)*
For each of the following multiple choice questions, indicate the best answer by circling it.

(a) [1 pt] Given the following code, what is printed?

```
pi = 3.14159
a = 3
b = 10

if b // a > pi :
    print('a')
elif b / a > pi :
    print('d')
else:
```

```
print("Nope")
```

A. `a`

B. `d`

C. `Nope`

D. A and B

E. None of the above

(b) [1 pt] `x = input("Please enter a number.")` What is `type(x)`?

A. None B. bool C. float D. int **E. str** F. list

(c) [1 pt] What is the type of the `a` variable after this line of code is executed?

```
a = print("7")
```

A. String B. Integer C. Float D. Boolean **E. NoneType** F. a is not assigned due to an error.

(d) [1 pt] Which of these python statements evaluates to `True`?

A. `not True and not False`

B. `True or False and False`

C. `True and False or False`

D. `True and True and False`

E. `not True or False`

(e) [1 pt] In 1964, Douglas Engelbart demonstrated the worlds first:

A. Transistorized Computer **B. Mouse** C. Tape Drive D. Vacuum Tube E. Solid State Memory

(f) [1 pt] What is/are the difference(s) between the / and the // operators in Python?

A. The // operator will perform string division.

B. The / operator will perform integer division while the // operator performs floating-point division.

C. The / operator will perform floating-point division while the // operator performs integer division.

D. The // operator works for only floats while the / operator works for any numeric type.

E. The / operator takes two key-presses to type, while the // operator takes one key-press to type.

F. The / and // operators do exactly the same thing, one is left over from Python 2 and is kept for compatibility reasons only.

(g) [1 pt] Which of these is a legal expression?

A. “A” + 5

B. “A” - 5

C. “A” * 5

D. “A” / 5

(h) [1 pt] Examine this code. What numbers does it print when executed?

```
x = 1
while x < 5:
   print( x+1 )
   x = x + 1
```

A. `1,2,3,4,5` **B.** `2,3,4,5` C. `1,2,3,4` D. `2,4,6` E. `1,3,5`

(i) [1 pt] Which of these would you use to generate string contaiing pi to three decimal places?

A. `"{}".format(3.14159)`

B. `"{.3f}".format(3.14159)`

C. `"{:.3f}".format(3.14159)`

D. `"{:f.3}".format(3.14159)`

5. *(9 points)*
Pretend you are the Python interpreter and the following code has been entered and executed. Write what is printed to the screen beside the code:

```
def foo(num):
    print("start recursion!")
    print(num)
    if num <= 0:
        print("stop")
        return 0
    elif num == 1:
        print("close")
        return 1
    final = foo(num-1)*2
    print("test")
    return final

def funct(num):

    go = True
    while go:
        print("loop running")
        if num%3 == 0:
            go = False
            print("change go")
        num = num - 1
        print("fun")
    return num
    print("finished")
print( foo(3) )
print( funct(11) )
```

Solution:

```
start recursion!
3
start recursion!
2
start recursion!
1
close
test
test
4
loop running
fun
loop running
fun
loop running
change go
fun
8
```

Grading: +3 for the ”start, 3, start, 2, start, 1, close” sequence (+1 all three starts) (+1 all three numbers) (+1 for close)
+1 for each ”test”
+1 for the ”4”
+2 for ”loop running, fun, loop running, fun, loop running, fun” sequence (+1 all three loop runnings) (+1 for all three funs)
+1 for ”change go”
+1 for the ”8”
-1 for extra stuff (max -3)

6. *(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)

7. *(5 points)*
Write a function named `isNegative` that takes in an integer parameter and returns a boolean. If the parameter is negative, the function should return the boolean value True. If the parameter is zero or positive, it should return the boolean value False.

Example test cases:

```
>>> print isNegative(4)
False
>>> print isNegative(-7)
True
>>>
```

Solution:

```
def isNegative( aNum):
   if aNum < 0:
return True
   else:
  return False
```

Grading: 1 point for a correct header.
2 points for testing if the number is positive/negative correctly. 1 point for returning False if the number is positive or zero.
1 point for returning True if the number is negative

8. *(10 points)*
Write a function named `square`. This function will cause your robot to move in a square. The function will accept one integer parameter, representing the length of one side of the square in inches.

You may assume that your robot moves one inch per second (forward or backwards), and turns 360 degrees in a time of 2 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 square( numInches):
  for i in range(4):
     forward(1,numInches)
     turnLeft(1,0.5)   #Turn 90 degrees!
  beep(1,800)
  return 4*numInches
```

Grading: 2 points for a correct header with parameter
2 points for iterting 4 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 90 degree corner exactly 90 degrees
(0.5 seconds 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.