

**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.**

## **CS 1301 Exam 1 Summer 2009**

<b>Problem</b>	<b>Earned Points</b>	<b>Possible Points</b>
1. Vocabulary		31
2. Python Expressions		19
3. Fill in the Blank		5
4. Multiple Choice		6
5. Get Number		5
6. Return Smallest		6
7. aFunc1		5
8. myInt		5
9. Run Robot Run		10
10. Robot Photographer		8
<b>Total:</b>		<b>100</b>

## 1. Vocabulary Matching (31 points)

Write the number before the definition on the right on the line before the matching vocabulary word.

___ algorithm	1. A programming language that is designed to be easy for the computer to execute that focuses on efficiency.
___ block	2. A programming language that hides details about computer hardware and focuses on human readability.
___ boolean expression	3. A general process for solving a category of problems; a finite series of steps that solve a concrete goal.
___ conditional statement	4. An error that does not occur until the program has started to execute but that prevents the program from continuing.
___ encapsulate	5. An error in a program that makes it do something other than what the programmer intended.
___ evaluate	6. An error in a program that makes it impossible to parse (and therefore impossible to interpret).
___ float	7. A Python data type that represents positive and negative whole numbers.
___ flow of execution	8. A Python data type that represents a sequence of characters.
___ function	9. A Python data type that represents a number with a fractional component.
___ high level language	10. A name that refers to a value.
___ immutable	11. A reserved word used by the compiler to parse a program; you can not use things like if, def, and while as variable names.
___ increment	12. To simplify an expression by performing the operations in order to yield a single value.
___ int	13. A special symbol that represents a simple computation like addition, multiplication, or string concatenation.
___ iteration	14. A named sequence of statements that performs some useful operation. They may or may not take parameters and may or may not produce a result.
___ keyword	15. The order in which statements are executed during a program run.
___ local variable	16. A name used inside a function to refer to the value passed as an argument.
___ low level language	17. A variable defined inside a function. These variable can only be used inside the function they are defined in.
___ modulus	18. An operator, denoted with a percent sign (%), that works on integers and yields the remainder when one number is divided by another.
___ None	19. An expression that is either true or false.
___ operator	20. Controls the flow of execution depending on some condition. In Python the keywords if, elif, and else are used for these.
___ parameter	21. A group of consecutive statements with the same indentation level.
___ proprioception	22. An explicit statement that takes a value of one type and computes a corresponding value of another type.
___ robot	23. A special Python value returned by functions that have no return statement, or a return statement without an argument.
___ runtime error	24. To increase the value of a variable by one.
___ semantic error	25. Repeated execution of a set of programming statements.
___ slice	26. To divide a large complex program into components (like functions) and isolate the components from each other (by using local variables, for example).
___ str	27. To iterate through the elements of a set, performing a similar operation on each.
___ syntax error	28. A part of a string (substring) specified by a range of indices, e.g. MyString[5:10]
___ traverse	29. A compound data types whose elements can not be assigned new values.
___ type conversion	30. A Mechanism guided by automatic controls.
___ variable	31. Sensor systems that give information about internal state.

## 2. Python Expressions ( 19 points)

Act like the python interpreter and evaluate the following expressions. Find what value the expressions evaluate to as well as its type (integer, float, string, boolean).

Expression	Evaluated Result (1 point)	Type of the Result ( ½ point)
"Hello" + "World" + "!"	"HelloWorld!"	String
3 + 2		
"cs1301" * 3		
int( 3.9) / 2		
( 6.0 - 1) ** 2 + 3		
"Thirty" + str(34) + "Four"		
True and ( 3 != 2)		
range(3,9)		
(7.0 + 6) / 2		
range(3,9,2)		
7.0 > 5.0		
print "Pumpkin %.3f" % 3.1459		
7 + 3 / 2 > 8		
(raw_input() > 3) or True		

## 3. Fill in the Blank ( 5 points )

In Python, a = is used for \_\_\_\_\_, while a == is used for \_\_\_\_\_.

When a function calls itself, it is said to be \_\_\_\_\_.

In python, the **if** keyword is used to make a \_\_\_\_\_ statement, while the **for** and **while** keywords are used to make \_\_\_\_\_.

## 4. Multiple Choice ( 6 points)

Circle the correct answer:

4a. Which of the following function definitions is correct?

<p><b>A.</b></p> <pre>def myFunc():     print "Hello!"</pre>	<p><b>B.</b></p> <pre>define myFunc():     print "Hello!"</pre>	<p><b>C.</b></p> <pre>def myFunc()     print "Hello!"</pre>	<p><b>D.</b></p> <pre>define myFunc():     print "Hello!"</pre>
--------------------------------------------------------------	-----------------------------------------------------------------	-------------------------------------------------------------	-----------------------------------------------------------------

**E.** None of the above

4b. Ada Lovelace is widely regarded as the first:

**A.** Computer Scientist   **B.** Programmer   **C.** Compiler   **D.** Discrete Mathematician   **E.** None of these

4c. Douglas Engelbart demonstrated the worlds first \_\_\_\_\_ in 1964 at Stanford.

**A.** Transistorized Computer   **B.** Mouse   **C.** Solid Sate Memory   **D.** Tape Drive   **E.** Transistor

4d. Grace Hopper was:

**A.** A Rear Admiral.      **B.** Awarded the “man-of-the-year” award from DPMA in 1969.  
**C.** Instrumental in the development of COBOL.      **D.** Credited for developing the first compiler.  
**E.** All of the above.

4e. The binary number  $\{ 1101111 \}_2$  is what decimal (base 10) number?

**A.** 101      **B.** 111      **C.** 102      **D.** 110      **E.** 112

4f. The decimal number  $\{ 31 \}_{10}$  is what binary (base 2) number?

**A.** 11111      **B.** 10101      **C.** 10100      **D.** 10111      **E.** None of these.

**5. Write Code - Get Number ( 5 points)**

Write a function named `get_int` that prompts the user to enter a number and returns an integer. You do NOT need to check for errors. (Assume the user always enters a valid number.)

**6. Write Code – Return smallest ( 6 points)**

Write a function named `return_smallest` that accepts 3 parameters ( $x,y,z$ ) and returns the smallest of the three. For example, `return_smallest(3,5,10)` should return 3, while `return_smallest(3,3,1.5)` should return 1.5.

## 7. Code Understanding - aFunc1 (5 points)

What does the following code print?

```
MyVar = 10
def aFunc1( MyVar):
    print MyVar * 3
    return( 5 )
    print "goodbye!"

MyVar = MyVar + aFunc1( "Go" )
print MyVar
```

## 8. Code Understanding – myInt (5 points)

What does the following code print?

```
for myInt in range(2,15,3):
    if (myInt % 2 == 0):
        print myInt * 2
    else:
        print myInt
```

## 9. Write Code – Run Robot, Run (10 points)

You have been given the following behavior to implement in python:

*Make your robot run away from things that approach it from the back. If something approaches from the left-hand side, the robot should move forward for 1 second while at the same time beeping at 880 Hz. If something approaches from the right-hand side, the robot should move forward for 1 second while at the same time beeping at 440Hz. Your robot should continue to repeat this behavior for a total of 35 seconds (irregardless of if something is behind the robot or not).*

You may find the following functions to be useful for coding up this behavior:

*getIR("left")* - Returns the left/back IR sensor value. A zero indicates something is behind the robot.

*getIR("right")* - Returns the right/back IR sensor value. A zero indicates something is behind the robot.

*timeRemaining( seconds)* – Returns True every time it is called for the specified number of seconds.

Be sure to include the proper import statements to load the myro libraries, and the correct function to initialize your robot (you may assume it is on COM4 or /dev/tty.scribbler, your choice).

## 10. Write Code – Robot Photographer (8 points)

A student in a previous class was given the following problem:

*Have your robot move forward and take pictures. Every time it takes a picture, it should turn to the right and then move forward again before taking another picture. Only show a picture if the light level reading returned by the center light sensor is **smaller than 150**. Your robot should move around and keep taking pictures until it has **shown 20** pictures.*

The student's code is below. Unfortunately, it does not work correctly. Re-write the student's code so that it works correctly as specified above. (your solution should need no more lines than the existing code, although they may be in a different order or changed.)

```
numPicturesTaken = 0
while numPicturesTaken > 20:
    forward(1,1)
    turnRight(1,0.5)
    p = takePicture()
    numPicturesTaken = numPicturesTaken + 1
    lightValue = getLight("center")
    if (lightValue > 150):
        show( p )
```