

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

Problem	Earned Points	Possible Points
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
Total:		100

1. Vocabulary Matching (31 points)

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

_ 3 __ algorithm	1. A programming language that is designed to be easy for the computer to execute that focuses on efficiency.
_ 21 __ block	2. A programming language that hides details about computer hardware and focuses on human readability.
_ 19 __ boolean expression	3. A general process for solving a category of problems; a finite series of steps that solve a concrete goal.
_ 20 __ conditional statement	4. An error that does not occur until the program has started to execute but that prevents the program from continuing.
_ 26 __ encapsulate	5. An error in a program that makes it do something other than what the programmer intended.
_ 12 __ evaluate	6. An error in a program that makes it impossible to parse (and therefore impossible to interpret).
_ 9 __ float	7. A Python data type that represents positive and negative whole numbers.
_ 15 __ flow of execution	8. A Python data type that represents a sequence of characters.
_ 14 __ function	9. A Python data type that represents a number with a fractional component.
_ 2 __ high level language	10. A name that refers to a value.
_ 29 __ 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.
_ 24 __ increment	12. To simplify an expression by performing the operations in order to yield a single value.
_ 7 __ int	13. A special symbol that represents a simple computation like addition, multiplication, or string concatenation.
_ 25 __ 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.
_ 11 __ keyword	15. The order in which statements are executed during a program run.
_ 17 __ local variable	16. A name used inside a function to refer to the value passed as an argument.
_ 1 __ low level language	17. A variable defined inside a function. These variable can only be used inside the function they are defined in.
_ 18 __ modulus	18. An operator, denoted with a percent sign (%), that works on integers and yields the remainder when one number is divided by another.
_ 23 __ None	19. An expression that is either true or false.
_ 13 __ operator	20. Controls the flow of execution depending on some condition. In Python the keywords if, elif, and else are used for these.
_ 16 __ parameter	21. A group of consecutive statements with the same indentation level.
_ 31 __ proprioception	22. An explicit statement that takes a value of one type and computes a corresponding value of another type.
_ 30 __ robot	23. A special Python value returned by functions that have no return statement, or a return statement without an argument.
_ 4 __ runtime error	24. To increase the value of a variable by one.
_ 5 __ semantic error	25. Repeated execution of a set of programming statements.
_ 28 __ 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).
_ 8 __ str	27. To iterate through the elements of a set, performing a similar operation on each.
_ 6 __ syntax error	28. A part of a string (substring) specified by a range of indices, e.g. MyString[5:10]
_ 27 __ traverse	29. A compound data types whose elements can not be assigned new values.
_ 22 __ type conversion	30. A Mechanism guided by automatic controls.
_ 10 __ 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	5	integer
"cs1301" * 3	"cs1301cs1301cs1301"	String
int(3.9) / 2	1	integer
(6.0 - 1) ** 2 + 3	28	float
"Thirty" + str(34) + "Four"	"Thirty34Four"	String
True and (3 != 2)	TRUE	Boolean
range(3,9)	[3,4,5,6,7,8]	List
(7.0 + 6) / 2	6.5	float
range(3,9,2)	[3,5,7]	List
7.0 > 5.0	TRUE	Boolean
print "Pumpkin %.3f" % 3.1459	"Pumpkin 3.146"	String
7 + 3 / 2 > 8	FALSE	Boolean
(raw_input() > 3) or True	TRUE	Boolean

3. Fill in the Blank (5 points)

In Python, a = is used for Assignment / variable declaration, while a == is used for equality checking / comparison.

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

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

4. Multiple Choice (6 points)

Circle the correct answer:

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

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

```
def get_int():  
    userString = raw_input("Enter a number:")  
    userInt = int( userString )  
    return( userInt)
```

Grading:

+1 point for correct def header

+1 point for prompting the user to enter a number

+1 points for getting user input (`raw_input` or `input`)

+1 points for ensuring that the result is an integer

+1 point for returning the answer

-1 point for minor syntax problems.

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.

```
def return_smallest( x, y, z):  
    if( x <= y and x <= z):  
        return( x )  
    elif (y <= x and y <= z):  
        return( y )  
    else:  
        return(z)
```

Grading:

+2 points for correct header w/ three parameters.

+2 points for returning the correct value in simple cases (1,2,5)

+2 points for returning the correct value in complicated cases (3,3,3), (2,2,1)

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
```

GoGoGo

15

Grading:

+3 points for printing "GoGoGo"

+2 points for printing "15"

-2 points for printing "goodbye!"

-1 point for anything else in the answer.

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
```

4

5

16

11

28

Grading: +1 point for each correct number. -2 if they list more than 5 numbers.

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).

```
from myro import *
init("COM4")

while timeRemaining(35):
    if getIR("left") == 0:
        forward(1)
        beep(1,800)
        stop()
    if getIR("right") == 0:
        forward(1)
        beep(1,440)
        stop()
```

Grading:

+1 points for "from myro import *"

+1 point for `init("com4")` (or /dev/tty.scribbler)

+2 points for getting the "something behind me" detection (`getIR` and `if` statements) correct.

+3 points for moving forward while beeping at the same time!

+3 points for repeating for 35 seconds (+/- 1-2 seconds)

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 )
```

Answer:

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

Grading:

+2 points for keeping track of the number of pictures shown, not the number of pictures taken.

+2 points for correct light value test (< 150)

+2 points for the correct numPictures shown test (< 20)

+2 points for implementing the behavior correctly (fully working).