CS 1301

Exam 1 Fall 2011

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 10 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. Short Answer	8	
4. Speed	10	
5. Div Function	8	
6. FooBar!	4	
7. Robot Drawing	8	
8. trainCrash	5	
9. CountUp	8	
10. Inches To Feet	8	
Total:	73	

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] boolean expression

Solution: A python statement that evaluates to either True or False. Typically contains a comparison operator (<, >, <=, >=, ==, !=) and may contain boolean operators such as AND, OR, or NOT.

(c) [3 pts] parameter

Solution: parameter - A name used inside a function to refer to the value passed as an argument.

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. *(8 points)*

   (a) [5 pts] Answer the following questions about python symbols and operators:

   Which symbol tells the interpreter to ignore anything after it?

   Which symbol is used for string formatting?

   Which symbol is used to enclose a string?

   Which symbol is the modulo operator?

   Which symbol in python indicates that a new code block will begin on the next line?

   Solution: Answer:
   Hash mark or pound (lb) sign
   Percent symbol (with string and value(s))
   Quote or double Quotes
   percent symbol (with numbers) Colon (at the end of a header) Grading: +1 for each correct answer

   (b) [3 pts] With respect to the print statement and the return statement:

   Which one ends a function?

   Which one always shows the value on the screen?

   Which one allows the value to be used later in the program?

   Solution: Answer:
   return
   print
   return

   Grading: +1 for each correct answer. (-1 for each incorrect answer. Student may not go below 0 on this part)

4. *(10 points)*

   Look at the following function and then copy it below. For each line, determine if the code will work or not. If the line is correct, just copy it. If the line is not correct, correct the line as you re-write it so that the function will work correctly.

```
Def speed():
   userIn = raw_input("Enter current car speed:")
   userIn = float(s)
   if userIn > 65
      print "You are driving too fast!" + userIn + "is more than 65!"
   elif speed < 40:
      print "You are driving too slow!" + userIn + "is less than 40!"
   Else:
        print "Good Driving, keep it up!"
```

Solution: Solution:
"Def" should be "def" - Worth +1
float(s) should be float(userIn) - Worth +2
first if statement needs a colon - Worth + 1
" + userIn + " should be "+ str(userIn) + " or something similar. Worth +1 on each of the two lines. "speed" should be "userIn" - Worth +2
"Else" should be "else" - Worth +2

5. *(8 points)*
Assume the following code is executed by the Python interpreter:

```
def div(a,b):
   if b != 0:
      result = a/b
      return result

num1 = div(5,2)
num2 = div(5,0)
num3 = div(4,3)
num4 = div(4.0,3)
```

(a) [2 pts] What is stored in num1?

Solution: 2

(b) [2 pts] What is stored in num2?

Solution: None. (+1 point for Nothing or Empty)

(c) [2 pts] What is stored in num3?

Solution: 1. In Python 2.x 4/3 is an integer division.

(d) [2 pts] What is stored in num4?

Solution: 1.333333 (float/int = float answer)

6. *(4 points)*
What does the following program PRINT when executed?

```
def foo(a, b):
   print "foo!"
   return bar(a, b)

def bar(a, b):
   res = 0
   for i in range(b):
      res = res + a
   return res

def runProgram():
   print foo(3, 4)
   return "Done!"

print runProgram()
```

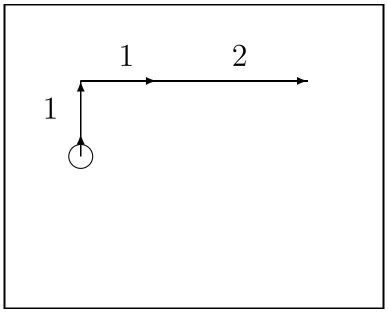
Solution: foo!
12
Done!
Grading: +1 for Foo! and Done! in correct place. +2 for 12 in correct place (only+1 if 12 not in correct place). -1 for each extra line

7. *(8 points)*
Robot Drawing - Assume `turn90degrees()` has been defined as below so the robot turns right 90° and `nudge(x)` has been defined to move the robot forward x units.

```
def turn90degrees():
    turnRight(1, 1)

def nudge(x):
    forward(1, x)
```

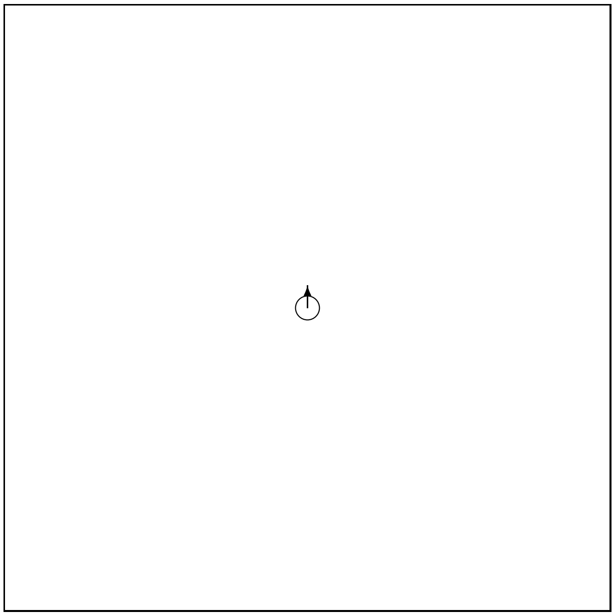
The following code makes the robot drive the trajectory drawn in the box to the right.

```
nudge(1)
turn90degrees()
nudge(1)
nudge(2)
```

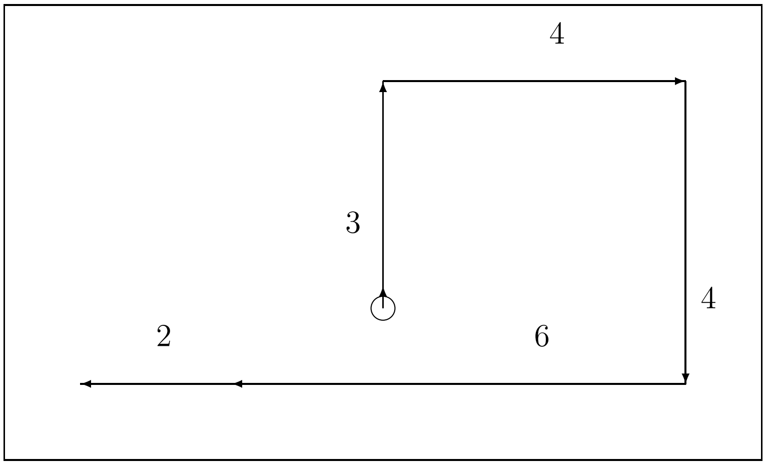


Draw the robot's trajectory when the following code is executed. Label the length of each move (nudge) using numbers as in the example above.

```
def turn90degrees():
    turnRight(1, 1)

def nudge(x):
    forward(1, x)

turns = [3,5]

for idx in [2,3,3,5,1]:
    if idx in turns:
       turn90degrees()
    nudge(idx + 1)
```



Solution:



Grading: 1 point for each correct line segment and length. 1 point for each correct turn.

8. *(5 points)*
Examine the following code:

```
def trainCrash (x):
   while x <= 10:
       if x % 5 == 0:
           return "oh no, Crash!"
       if x % 3 == 0:
           print "I'm a train..."
       x = x +1
       print "Choo-choo!"
   return "I'm too tired to go on"
```

If this code is called from the IDLE window as follows:

```
y = trainCrash(8)
```

(a) [3 pts] What is displayed on the screen?

Solution:

```
Choo-choo!
I'm a train...
Choo-choo!
```

Grading: 1 point for each correct line, -1 point for each extra line

(b) [2 pts] What will be stored in the y variable from the example function call above?

Solution:

```
'oh no, Crash!'
```

Grading: 2 points for the string stored in y.

9. *(8 points)*
Write a function called **countUp** that accepts two integer parameters. The function will print out all integers between the two parameters (excluding both parameters!) in ascending order *using a while loop*. You may assume that the two parameters are valid integers, and that the first parameter will always be smaller than the second parameter. Note that you **may not use a for loop** in your solution!

Example test case:

```
>>>countUp(1,5)
2
3
4
>>>
```

Solution:

```
def countUp(a,b):
   a = a + 1
   while a < b:
      print a
      a = a+1
```

Grading: 2 points for a correct header.
2 points for starting the printout at a+1
2 points for printing multiple numbers with the while loop.
2 points for stopping correctly just before b.

10. *(8 points)*
Write a function named `inchesToFeet` that takes in a float parameter which represents a number of inches. It should return a string representing the calculated number of feet and inches in the input parameter.
The string must be formated as follows: ”X feet, Y.Y inches”, where X and Y should be replaced by the number of feet and inches you calculate. The string you return should be formatted such that the remainder of inches always has one decimal place. Note: 1 foot = 12 inches.

Example test cases:

```
>>> ans1 = inchesToFeet(44.0)
>>> print ans1
3 feet, 8.0 inches
>>> ans2 = print inchesToFeet(65.5)
>>> print ans2
5 feet, 5.5 inches
>>>
```

Solution:

```
def inchesToFeet( aNum):
   feet = int(aNum) / 12
   #or feet = int(aNum / 12)
   inches = aNum % 12

   answer = "%i feet, %.1f inches" % (feet, inches)
   return answer
```

Grading: 1 point for a correct header.
1 point for using ”int()” to make the feet come out right
1 point for dividing by 12 to get the number of feet
2 points for using MOD correctly to find the number of left-over inches.
1 point for formatting the inches answer with one decimal place.
1 point for returning a string.
1 point for correctly inserting the number of inches and feet into the string.