Your Name: _____ Your TA's Name: _____

CS 1301 CS1 with Robots Spring 2009 – Exam 2
You have 50 minutes for this exam.

Problem	Points Earned	Possible Points
1. Vocabulary Matching		15
2. N_Lines		3
3. Robot Directions		10
4. Fill in the Blank		5
5. Expression Evaluation		20
6. Change Letter		10
7. Change Value		5
8. Return Smallest		6
9. My Length		15
10. Reverse List		15
Extra Credit		(4 possible)
Total Score		104

Your Name: _____ Your TA's Name: _____

1. Vocabulary Matching: (15 points) – 1 point each

Write the number from the correct definition in the blank next to each term on the left:

_14__compound data type _5__aliases _7__sequence _3__recursion _4__iteration _1__slice _6__traverse _8__nested list _13__clone _12__immutable type _2__mutable _9_increment _15__semantic error _10__decrement _11_element	1. A part of a string specified by a range of indices. 2. A compound data type whose elements can be assigned new values. 3. The process of calling the function that is currently executing. 4. Repeated execution of a set of statements using either a recursive function call or a loop. 5. Multiple variables that contain references to the same object. 6. To move through the elements of a sequence, such as a list, performing a similar operation on each. 7. Any of the data types that consist of an ordered set of elements, with each element identified by an index. 8. A list that is an element of another list. 9. To increase the value of a variable by one. 10. To decrease the value of a variable by one. 11. One of the values in a list (or other sequence). The bracket operator selects an _____ of a list. 12. A type in which the elements cannot be modified. Assignments to elements or slices of these types cause an error. 13. To create a new object that has the same value as an existing object. 14. A data type in which the values are made up of components, or elements, that are themselves values. 15. Does not display red text to the console or interrupt the program from running.

2. Program Comprehension - N_Lines (3 points)

```
def n_lines( n ):
    print "Line!"
    if ( n >= 0) :
        n_lines( n – 1)
```

How many times will the string "Line!" be printed when n_lines is called with n=4?
Number_____6_____   - 3 points, all or nothing

3. Robot Directions (10 points)
The following code makes the robot drive the trajectory drawn in the box to the right.

def turn90degrees():
    turnRight(1, 1)
 	
def nudge(x):
    forward(1, x)
 	
nudge(1)
turn90degrees()
nudge(1)
nudge(2)	

Draw the robot's trajectory when the following code is executed. Start the robot in the middle of the box and use arrow heads (as above) to indicate each movement.

def turn90degrees():
    turnRight(1, 1)
 	
def nudge(x):
    forward(1, x)
 	
nums = [4,3,2,1]
 	
for I in nums:
    if (I % 2 == 0) :
        turn90degrees()
    nudge(I)
	5 pts – Turns before 4 & 2
	5 pts – Nudges each time, but less (corresponding to I)
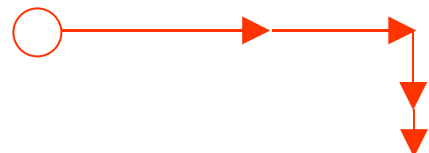
Your Name: _____ Your TA's Name: _____

4. Fill in the blank (5 points) - 1 point each

In python, the = operator performs _____assignment_____ while the == operator performs

____equality_____.

Python has several compound data types that we have learned about. A __string_____ can be used to

store a sequence of characters, while a __tuple_____ can store a sequence of any type of data (but is

immutable). A __list_____ can also store any type of data, and allows you to change elements within it.

5. Python Expression Evaluation (20 points) – 2 pts each (small errors -1 – such as lower case Booleans, an extra element in the list, or no quotes around strings)

For this question, assume the following statements have already been entered and interpreted:

```
a = [ 5, 10, 15, True, ["Cherry", "Apple","Plum"], 56, [4, 5, 6], 84 ]
b = a
c = a[0:4]
d = a[4]
d[2] = "Peach"
```

Pretend that you are the Python Interpreter (IDLE window). What do you print or return when each of the following statements are entered?

Example: a[0]   *Result:___**5**____*

Example: a[1:4]   *Result: _ [10, 15, True]_*

1. **a[6][0]**   *Result:___4_____*

2. **d**   *Result:_["Cherry", "Apple", "Peach"]__*

3. **c**   *Result: _[5, 10, 15, True]_____*

4. **a[4][2]**   *Result: _"Peach"_____*

5. **b[:2]**   *Result: _[5, 10]_____*

6. **b[-2]**   *Result: _[4, 5, 6]_____*

7. **c[-2]**   *Result: __15_____*

8. **print "Pumpkin %.3f" %3.1459** *Result: __"Pumpkin 3.146"____*

9. *(5 > 10) or (5 > 3)*   *Result: _True_____*

10. *34 % 10*   *Result: _4___*

6. Write Code – Change Letter (10 points)

Write a function **changeLetter(aString, index, newLetter)** that will replace the letter stored at index in aString with the contents of newLetter and *return* the new string *without modifying the original string!* For example, *changeLetter("Python is great!", 10, "G")* will return the string "Python is Great!"

```
def changeLetter(aString, index, newLetter):
    newString = aString[:index] + newLetter + aString[index + 1:]
    return newString
```

2 pts for correct header
6 pts for correct modification
    - 2 indexing before
    - 2 adding newLetter in right place
    - 2 indexing after
2 pts for returning the string

7. Write Code – Change Value (5 points)

Write a function **changeValue(aList, index, newValue)** that will replace the element stored at index in aList with the contents of newValue. It should NOT return the list. For example after the following commands:

a = [5, True,"Test",10]

changeValue(a, 3, "Hi!")

The list **a** will be [5, True,"Test","Hi!"]

```
def changeValue(aList, index, newValue):
    aList[index] = newValue
```

1 pt – header
1 pt – no return
3 pt – change right value

8. Reading Code – Return Smallest (6 points)

```
# This function accepts 3 parameters (x,y,z) and is supposed
# to return the smallest of the 3.
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 )
```

a. What is wrong with the above code?

No checks for equality

2 pts

b. Give an example input that would produce an error.

(2, 2, 15)

2pts

c. Tell how to fix the problem.

change the < to <=

2pts

9. Write Code - myLength (15 points)

Write a function myLength (sequence) that **returns** the length of the sequence. (In essence, you are re-implementing the system's len(x) function.) Obviously, you **may not** use the len(x) or any other system functions to do the work for you.

```
myLength("test" )     #evaluates to 4

myLength([1,2,3])     #evaluates to 3

myLength( (-2,4,8,2) )    #evaluates to 4
```

```
def myLength(seq):
      count = 0
      for i in seq:
            count += 1
      return count
```

3 pts – def line
3 pts – using a counter
4 pts – iterate correctly
3 pts – increment counter
2 pts - return

Your Name: _____ Your TA's Name: _____

10. Write Code – Reverse List (15 points)

Write a function **reverseList(aList)** that will return a reversed copy of aList. For example, after the following:
a = [5, 10, True, "Hi!"]

b = reverseList(a)

The list b = ["Hi!", True, 10, 5], while a = [5, 10, True, "Hi!"].

```
def reverseList(aList):
    return aList[::-1]
```

5 pts – def line
10 pts – returns correct thing

OR

```
def reverseList(aList):
    bList = []
    for x in aList:
        bList = [x] + bList
    return( bList)
```

3 pts – def line
3 pts – making new list
3 pts – correct iteration
3 pts – adds elements to new list
3 pts – returns

Your Name: _____ Your TA's Name: _____

Extra Credit (1 point each)

What function do you use to get the Battery voltage? _____getBattery()_____

What is the decimal representation of the binary number $\{100101\}_2$? _____37_____

What is the hexadecimal representation of the decimal number $\{62\}_{10}$? _____3E_____

What does CSS stand for? Cascading_____ Style_____ Sheet_____