CS1301 - Exam 2 Name: ______________________ Your Grading TA: ____________

Instructions:

- Please write clearly. What I cannot read, I will not grade.
- Show all your work in detail. I give partial credit.
- This exam has 7 pages including the title page. Please check to make sure all pages are included.
- This exam is closed book, closed notes, no calculators.
- Don't get bogged down on any one question. 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.

Signature: ____

Question	Points	Score
1. Vocabulary	15	
2. Multiple Choice	5	
3. Types of Reading	6	
4. DooWaa	6	
5. Mystery Code	3	
6. trainCrash	5	
7. Breakup	7	
8. Stick Together	9	
Total:	56	

Spring 2010

1. For each of the following vocabulary terms, write a concise 1-2 sentence definition. Be brief, and to the point.

   (a) (3 points) dictionary

   Solution: A mutable compound data type that associates keys with values.

   (b) (3 points) slice

   Solution: A subsequence copied from a sequence specified by a range of indices. The slice operator is: sequence[start:stop].

   (c) (3 points) traverse

   Solution: To move through all elements of a set, performing a similar operation on each element.

   (d) (3 points) mutable type

   Solution: mutable type - A data type in which the elements can be modified. All mutable types are compound types. Lists are mutable data types; strings are not.

   (e) (3 points) decrement

   Solution: decrement - The process of decreasing a variable, typically by one.
   aVar = aVar - 1

Multiple Choice

2. For each of the following questions, select the appropriate answer by circling it.

   (a) (1 point) Order the following items from earliest (older) to latest (newer):

   1. Konrad Zuse's Z1 computer
   2. The Transistor
   3. ARPANET

   A. 1,2,3 B. 2,3,1 C. 1,3,2 D. 2,1,3 E. None of these.

   (b) (1 point) Order the following items from earliest (older) to latest (newer):

   1. The Jacquard Loom
   2. Ada Lovelace's program for the Analytical Engine
   3. Jacques De Vaucanson's Digesting Duck

   A. 1,2,3 **B. 3,1,2** C. 2,3,1 D. 2,1,3 E. None of these.

   (c) (1 point) Which of these would you use to print a number to four decimal places?
   A. "%4i" B. "0%.4i" C. "%4f" **D. "%.4f"** E. "%0.5f"

(d) (1 point) Convert 11011001_2 to decimal (base 10):
A. 217 B. 225 C. 232 D. 233 E. 234

(e) (1 point) Which data type is mutable?
A. int B. float C. str **D. list** E. tuple

Short Answer

3. Three functions for reading from a file are read(), readline(), and readlines(). Briefly explain what each of these functions returns when called. Be sure to explain how each function differs in behavior.

(a) (2 points) read() -

Solution: read() - This function returns a string that contains the entire contents of the file.

(b) (2 points) readline() -

Solution: readline() - This function contains a string that contains only the first line of the file. (all characters up to, and including, the first newline character, denoted by a \n).

(c) (2 points) readlines() -

Solution: readlines() - This function returns a list that contains each line in the file as a string.

Code Understanding

4. (6 points) Fill in the blanks so that, when run, the code below will output the followng:

```
>>> func1()
DooWaa
Diddy
Diddy
Dum
Diddy
Doo
```

```
def func1():
    print "DooWaa"
    for i in range(  ___________________ ):

        print __________________

        if i == ________________ :
            print "Dum"
      print "Doo"
```

Solution:

```
def func1():
    print "DooWaa"
    for i in range(___3___):
        print __"Diddy"__
        if i == ___1___ :
            print "Dum"
    print "Doo"
```

Grading: 2 points for each correct blank. -1 for any minor syntax errors. (leaving out quotes, etc)

5. (3 points)

```
def mysteryFunc(x,y):    # x and y are non-negative integers
   if y == 0:
      return 0
   else:
      return x + mysteryFunc(x,y-1)
```

What does this mystery function do? Also tell us the value returned as a result of calling

`mysteryFunc(5,2)`

Solution: Answer: This function multiples the two arguments by using repeated addition. The number 10 proves they know how the function works.

Grading:
3 points if they understand what it does, and give us the correct answer 2 points if they describe some of what it does, but give us the wrong number. 1 point if they mention recursion, but don't give the correct answer. 0 points for anything else/less.

6. 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 points) 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 points) 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.

Code Writing Questions

7. (7 points) Breakup - Write a function called `breakUp` that accepts a string as a parameter. It should return a list which is made up of single character strings, one per letter in the original string.

   For example:

```
>>> result = breakUp( "Yey excellence fees!" )
>>> print result
 ['Y','e','y',' ', 'e','x','c','e','l','l','e','n','c','e',' ','f','e','e','s','!']
```

Solution:

```
def breakUp(aString):
  aList = []
  for letter in aString:
     aList.append(letter)
  return(aList)
```

Grading:
1pt - Correct def statement
2pt - traverses the string
2pt - adds each letter to the list
2 pt - returns new list

8. (9 points) Stick Together - Write a function called `stickTogether` that accepts a list as a parameter. It should return a string that is the concatenation of all string elements in the list. Note that non-string elements should be skipped, including nested lists.

   For example:

```
>>> result = stickTogether( [ 4, "Hello", ["bob",3], " ", True, "There!", 7]  )
>>> print result
 "Hello There!"
```

Solution:

```
def stickTogether(aList):
  aString = ""
  for item in aList:
     if type(item) == type( ’letters’):   # or:  if type(item) == str:
        aString = aString + item
  return(aString)
```

Grading:
1pt - Correct def statement
2pt - traverses the items in the list
2pt - checks the type of each item. 2pt - concatenates each item of type string to the string. 2pt - returns the string.