 Your Name: _____

CS 1301 Summer 2009 Exam 2/2

Problem	Earned Points	Points Possible
Vocabulary Matching		22
Python Expressions		20
Comedy & Drama		8
Phonebook		6
Fill in the Blank		4
LongWords		12
Pixel Swap		14
Robot Photographer		14
Total:		100

 Your Name: _____

1. Vocabulary Matching (22 points)

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

____ aliases ____ clone ____ compound data type ____ decrement ____ dictionary ____ exception ____ file ____ format operator ____ global variables ____ immutable type ____ increment ____ iteration ____ lambda ____ local variables ____ mutable type ____ nested list ____ pixel ____ recursion ____ sequence ____ short circuit evaluation ____ slice ____ traverse	1. Smallest addressable element of a picture. 2. A variable that can only be accessed within the function that it was defined in. 3. The % operator takes a format string and a tuple of values and generates a string by inserting the data values into the format string at the appropriate locations. 4. When a boolean expression is evaluated the evaluation starts at the left hand expression and proceeds to the right, stopping when it is no longer necessary to evaluate any further to determine the final outcome. 5. A block of code which can be executed as if it were a function but without a name. 6. Can be seen through a program module, even inside of functions. 7. A named entity, usually stored on a hard drive, floppy disk, or CD-ROM, that contains a stream of characters. 8. Raised by the runtime system if something goes wrong while the program is running. 9. A data type that is itself made up of elements that are themselves values. 10. Multiple variables that contain references to the same object. 11. A data type that is made up of elements organized linearly, with each element accessed by an integer index. 12. The process of calling the currently executing function. 13. To repeat an operation on all members of a set from the start to the end. 14. A copy of part of a sequence specified by a series of indices. 15. To repeat a section of code. 16. A list that is itself contained within a list. 17. To create a new object that has the same value as an existing object. 18. A compound data type whose elements can be assigned new values. 19. A compound data type whose elements can NOT be assigned new values. 20. To add one to a variable. 21. To subtract one from a variable. 22. A collection of key/value pairs that maps from keys to values.

 Your Name: _____

2. Python Expression Evaluation (20 points)

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"
x = { 1: "one", 2 : "two"}
```

Pretend that you are the Python Interpreter (IDLE window). Watch out for the difference between aliases and clones! 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:*_____
2. **d** *Result:*_____
3. **c** *Result:* _____
4. **a[4][2]** *Result:* _____
5. **b[:2]** *Result:* _____
6. **x[2]** *Result:* _____
7. **b[-2]** *Result:* _____
8. **c[-2]** *Result:* _____
9. **x.get(0 , 5)** *Result:* _____
10. **print "Pumpkin %.3f" %3.1459** *Result:* _____

 Your Name: _____

3. Comedy & Drama (8 points)

a. Write a function called addComedy that takes a list as input, adds the string ":)" to the end of the list, and returns the modified list. This function should modify *and return* the original list.

Example:

```
>>> a = [True, 4.0, "Saturday"]
>>> addComedy(a)
[True, 4.0, "Saturday", ":)"]
>>> a
[True, 4.0, "Saturday", ":)"]
```

b. Write another function called addDrama that takes a list as input, makes a duplicate of the list, adds the string ":(" to the end of the duplicate, and returns the modified list. Note that unlike addComedy, this function should NOT modify the original list!

Example:

```
>>> a = [2.85, 98, "Othello"]
>>> addDrama(a)
[2.85, 98, "Othello", ":("]
>>> a
[2.85, 98, "Othello"]
```

4. PhoneBook (6 points)

You have a list of names and telephone numbers stored in a dictionary called phoneBook. The names are the keys, and the numbers are the values. Both the keys and values (names and numbers) are stored as strings. What ***single line of code*** would you need to execute in each of the following scenarios to update the phoneBook dictionary correctly?

a. Your old friend Steve has changed his number from "123-4567" to "987-6543". (You may assume the key "Steve" already exists in the phone book with the value "123-4567" associated with it.)

b. Steve introduces you to his younger sister, Jenny, whom you've never met before. (Her name is not in your phone book.) Her number is "867-5309", and you add it to your phonebook.

c. Steve informs you that he has been selected by the UN to be an undercover secret peace agent, keeping the world safe from megalomaniacs and mad scientists. Unfortunately, this means you won't be able to contact him by telephone any more. Remove his entry from your phone book.

5. Fill in the Blank (4 points)

Python has several compound data types that we have learned about. A _____ can be used to store a sequence of characters, while a _____ can store a sequence of any type of data (but is immutable). A _____ can also store any type of data, and allows you to change elements within it. Finally, a _____ can associate a value to a key.

6. LongWords (12 points)

The function longWords(aList) accepts a list of strings and prints out each string with more than five letters in it. You may assume that only lists containing nothing but strings will be passed into this function.

Example:

```
>>> a = [ 'a', 'to', 'two', 'reallybigstring', 'anotherlongstring']
>>> longWords(a)
reallybigstring
anotherlongstring
```

a. Write longWords using a while loop.

b. Write longWords using a for loop.

c. Write longWords using a small helper function (named printIfBig) and map.

 Your Name: _____

7. PixelSwap (14 points)
Write a function called pixelSwap() that will have your robot take a picture and then swap the red and green values of every 3rd pixel. After it swaps the red and green pixel value of every third pixel, it should return the modified picture.

 Your Name: _____

8. Robot Photographer (14 points)

Write a program that makes 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. Right after it takes a picture, it should use the getLight("center") function to sample the light value in that location. 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 (no matter how many pictures it has taken!)