CS 1510 : Fall 2017 : Final Exam Review

Indicate the data type of the for loop iterator based on context:

thing = "As you lead us onto fame and honor, FIGHT FIGHT FIGHT!
Will be our cry. So give us a yell (ho!) the Purple and the Gold, Victory for UNI! U-N-I Fight! U-N-I Fight!"

1.
for element in thing:

Q. What is element? ______________________

2.
thing = thing.split()

for element in thing:

Q. What is element? ______________________

3.
```
thing = thing.split()
for element in thing:
        for something in element:
```

Q. What is something? ______________________

4.
thing = "As you lead us onto fame and honor, FIGHT FIGHT FIGHT!
Will be our cry. So give us a yell (ho!) the Purple and the Gold, Victory for UNI! U-N-I Fight! U-N-I Fight!"

Use a dictionary to count the number of words. Write a function that takes in a string and prints each word followed by a count, for example:

```
        as 1
        you 1
        ...
```

5. How would you do the same as in Q4, but print out the words alphabetically?

6. How would you do the same as in Q4, but print out the results ordered by count in descending order?

7.
listOfGrades = [70.5, 81.2, 77.4, 90, 85]

Q. How would I find the average of these grades? Write a function that takes in a list and returns the average.

8.
Bob = {'France','Germany','Canada'}
Alice = {'Canada','Jamaca','Brazil'}

How would I write a program to find places in common that both Bob and Alice have traveled? Write a function that takes in two sets and returns a new set of places in common.

---

9. Previous Topics:

- Foundations (i.e., intro concepts)
- Loops (for, while, when to use, sntinel counter, infinite loops, nested, break, continue, etc.)
- Conditionals (if/elif/else, boolean logic, etc.)
- Strings (string methods, using loops/conditionals to modify/create strings, etc.)
- Reading/writing files
- Functions (definitions, calling a function, passing arguments, variable scope, etc.)

---

10. Lists:

What are they?
How do you add items to a list
How do you remove things from a list
How do YOU search a list? (writing code that does this, not using native functions)
What are the different ways to search a list that we discussed in class? Can you code them?

---

11. Dictionaries:

What are they?
How do you access an individual value based on its key?
How do you change/set a key to a particular value?
How do you get access to all of the keys of a dictionary?
How do you get access to all of the values of a dictionary?
How do you print out the items (key/value pairs) sorted by key? By value?

---

12. Sets:

What are they?
How do you create an empty set?
How do you add items to a set?
How do you check to see if an item is in a set?
How do you use the common features such as union, intersection, and set difference?

---

13. Sample list programs:

13.1 Write a function called negate(myList) which takes in a list of numbers as a parameter and negates each value in the list. Since the function changes the list provided as a parameter it ***does not need to return*** anything.

13.2 Write a function called negativeCopy(myList) which takes in a list of numbers as a parameter and ***returns*** a newly created list which contains the negative values of the original list. For example, negativeCopy([1, -2, 3.5, 4.2,]) would *return* [-1, 2, -3.5, -4.2]

13.3 Write a function called addList(myList, addend) which takes in a list of numbers and a single number as parameters. The function ***returns a newly created list*** which is each number in the original list ***plus*** the addend. For example, addList([1,2,3.5,4.2] , 3) would return [4, 5, 6.5, 7.2]

13.4 Write a function called reverseList(myList) which takes in a list of items and ***returns a newly created list*** which is the original list in reverse. For example, reverseList([1, 2, 3.5, 4.2]) would return [4.2, 3.5, 2, 1]

13.5 Write a function called removeItem(myList, item) which takes in a list and an item and ***returns a new list*** with each occurrence of the item removed from the original list. For example, removeItem([1, 2, "the", "test", 3, "is", "the", "best"], "the") would return [1, 2, "test", 3, "is", "best"]

13.6 Write a function called sumOf(myList) which takes in a list of numbers as a parameter and returns the sum of all of the values in the list.

13.7 Write a function called productOf(myList) which takes in a list of numbers as a parameter and returns the product of all of the values in the list.

13.8 Write a function called count(myList,item) which takes in a list and an item and returns an integer representing how many times the item was in the original list. For example, count([1, 2, "the", "test", 3, "is", "the", "best"], "the") would return 2, but count([1, 2, "the", "test", 3, "is", "the", "best"], "Another") would return 0. (note, I am asking you to do the work here, not use native functionality).

13.9 Write a function called mean(myList) which takes in a list of numbers and returns the mean (the statistical "average") of the numbers in the list.

13.10 Write a function called mode(myList) which takes in a list of numbers and returns the value which occurs in the list the most times.

---

14. Sample dictionary programs:

14.1 Write a function called keyCount(myDictionary) which takes in a dictionary as a parameter and returns an integer representing the number of keys in the dictionary

14.2 Write a function called valueCount(myDictionary) which takes in a dictionary as a parameter and returns an integer representing the number of ***UNIQUE*** values in the dictionary.

14.3 Write a function called valueCount(myDictionary,value) which takes in a dictionary as a parameter and returns an integer representing how many times ***THAT value*** is a value in the dictionary.

14.4 Write a function called setValue(myDictionary, key, newValue) which takes in a dictionary as a parameter, locates the key in the dictionary and replaces the existing value with the newValue parameter.

14.5 Write a function called mostCommon(myString) which takes in a string and returns the most common letter in that string.

14.6 Write a fucntion called letterCounts(mystring) which takes in a string and returns a dictionary of letter counts in that string.

---

15. Sample set programs:

15.1 Write a function called commonLetters(name1, name2) that takes in the names of two people. It then returns a list of the letters ***common*** to both names.

15.2 Write a function called boxOfCrayons(colors1, colors2) that takes in two lists of colors and returns the list of ***all the colors*** from the two lists.

15.3 Write a function called howFarAhead(student1, student2) that takes in two lists of courses completed by two students. It should return a list of all of the courses that student1 has completed that student2 ***has NOT*** completed.