CS230 Quiz 2 Guide (4/4/2023)

Topics

Topics

- Strings
  - Indexing, Slicing
  - Common String Functions and Methods
- Functions
  - Parameters and return values
  - Multiple Parameters, Default Parameters
  - Multiple Values Returned
- Lists
  - Creating
  - Iterating
    - for item in a_list
    - for i in range(len(a_list))
  - Adding/removing elements
  - Slicing, Searching, Sort or Reverse
  - Lists of Lists
  - Tuples
- Files and file system
  - OS module functions
  - Current working directory, listing the files in a folder
    - statements to open a file, opening modes
  - Reading a line / multiple lines of a file
  - Reading the whole file in one operation
  - Writing into a file
  - ~~Exception Handling using Try/Except block~~
- Dictionaries
  - Creating, Adding items
    - Accessing data by key, keys, values;
    - Iterating over dictionaries
    - Modifying values in a dictionary
- Working with combinations of data structures: dictionaries/tuples/lists

Overview

- Quiz will take place during class
- It will consist of some short programming questions and some longer programming problems.

- May use **one sheet (both sides) of notes** for reference during the exam and no other reference. You may use paper as needed to work on drafts for your answers.
- You can take the quiz **on paper** or in **lockdown browser**. The lockdown browser runs from BrightSpace, limits what you do on the computer to just entering text into BrightSpace. You need to download it from here and set it up before the quiz.
    - You **will not have access to a Python editor or environment**.
    - A sample lockdown browser **practice test will be** available for you to try the working in the lockdown browser on the course Assignments page ahead of time.

- Partial credit will be given

Practice problems

Note – a py file with this content is provided as a starting point for you on our course schedule.

#1

What is generated by the following print statements? - **answer without executing this code.**

```
a = { "tom":["cs150", "cs230"],
      "joe":["FT370"],
      "jill":["FT370","ma360"]
      }
b = {"t":34, "c":117, "d" : 20, "o":45 }
c = [  'dinner'.count(ch)  for ch in  'dinner']
d = "university"
e = ["Norway", "Brazil", "USA" ]


print ('1 -->', len(a["jill"]))
print ('2 -->', sorted(b.keys()))
print ('3 -->', "joe" in a, 20 in b )
print ('4 -->', c)
print ('5 -->', set(c) )
print ('5 -->',  d.find('uni'), d.find("city") )
print ('6 -->',  e.index("Brazil"), "country" in e )
# 2
```

Add code to update code above to do the following:

- add cs370 to tom's courses
- calculate sum of all values in dictionary b
- replace letters i and e in d with capitals
- list all classes listed in dictioinary a
- find the maximum of all values in b
- output the content of the dictionary into a file "registrations.txt" in a form

```
tom:cs150 cs230
joe:FT370
jill:FT370 ma360
```

#3

Given the two parallel lists below (one listing `animals`, another, a number of `days` each of these animals was in the clinic), ask the user to enter an animal name, then, if that animal name is inside the list `animals`, increase the number of `days` associated with that animal by 1

```
animals = ['Hippopotamus','Horse','Kangaroo','Chipmunk']
days = [225, 330, 42, 31]
```

#4

Define and test a function `isValidPassw`() that is passed a string as a parameter and returns True, if and only if the string satisfies all of the following conditions:

- contains between 6 and 15 characters
- contains no spaces
- contains at least one letter, one number and one special symbol (non-space, non-number, non-letter)

Hint: go through letters, computing how many of each type of character there is, then make a decision based on those numbers

#5

Define and test a function allPasswords() that will ask user to enter a string of words, and will check each word for being a valid password, as defined above, using the isValidPassw() function. Return a list of all valid passwords sorted in alphabetical order.

For example, given the following input

```
sdj45ff. fhe4 df!3 sdfdfsdf2222.... passw12?
```

the function should return

```
['passw12?', 'sdj45ff.']
```

#6

Given the list of dictionaries `data` below, print each person's name and age, and then print the name and age of the youngest person, as shown:

```
Miraj 30
Tom 24
Lelise 25
Phung 27
Tom is the youngest, at age 24
```

```
data = [ {"name": "Miraj", "age": 30, "salary": 80000},
         {"name": "Tom", "age": 24, "salary": 90000},
         {"name": "Lelise", "age": 25, "salary": 65000},
         {"name": "Phung", "age": 27, "salary": 74000}]
```