

## ICT1002 Programming Fundamentals

### Lab 2

#### Topics:

1. Advanced data structures in python: list, tuple, dictionary
2. For/while loops
3. File I/O

#### Warmup exercises:

The following lab assignment requires the use of all topics discussed so far in the module. You may wish to practice some of the concepts with simple exercises before attempting the lab assignment. You are not required to include these exercises in your submission, though you may wish to do so, to help you in the lab test.

1. Evaluate the following expression

List1=['abc', 'bcd', ['123', 567], 789]	Tuple1[1:]
List1[1]	Tuple1[:2]
List1[2]	Tuple1[2]=23
List1[2][2]	Dic={'A01':'xiaoming', 'A02':['mie', 3]}
List1[4]	Dic['A01']
List1[-3]	Dic['A02']
List1[2:]	Dic['A02'][0]
Tuple1=('23',15,8,100)	range(10)
Tuple1[0]	range(2:10:2)

2. Create one tuple and one list to store five of your favourite fruits (e.g. pear, apple, strawberry, banana, orange ) separately

---



---

3. Print all the items that you have created from the list and tuple

---



---

4. Update one of the items from the list or tuple, e.g. change the 'apple' into 'papaya'

---



---

5. Delete one of the items from the list or tuple, e.g. delete pear

---



---

6. Sort all the remained fruits and print the ordered fruits out using for and while loops

---

---

---

---

7. Create one dictionary to store 12 months and its corresponding number of days. Use for and while loops to print out all the months and all the number of days.

---

---

---

---

8. You are going to design one program to check the popular words in a given document. Please download the lab2\_testdata.txt from LMS. Note that the given data has a fixed scheme where each line is one long string and each string contains multiple keywords that are separated by “,”. You need to write one program to read this file and calculate the top 5 most frequent keywords and write out these 5 keywords in the end of the file. The following are some hints which may help you design this program.

- String has a cool function that you can use to split a string separated by a ‘,’. For instance, given one string `str1 = “apple, pear, peach”`, to get all the keywords in `str1`, `str1.split(',')` will return a list of keywords. You can use `list1=str1.split(',')` to obtain all the keywords and put them into one list.
- To get the top 5 most frequent keywords, you need to extract all the keywords first and figure out one way to calculate the frequency of each keyword. Then select the top 5 keywords.

**Input:** The given text file “testdata.txt”

**Output:** Print out the top 5 keywords both in the screen and write them to a new txt file “top\_5.txt”.

**\*\*\*\*\* You must try exercise 8 to learn how to do the Files I/O!\*\*\*\*\***

## Lab Assignment:

To help you better practice, you need to perform a set of tasks in one auto-grading system, CodeDr. CodeDr will provide you immediate feedback of your program, such that you will know the issue of your program. Below is the link for you to access the system:

<http://172.27.54.87/codeDr/public>

PS. The codeDR system is only accessible via the ICT network. You can access it from outside via the ICT VPN. The ICT VPN account confidential should have been sent to your sit email. If you have not or

have any issue using the ICT VPN, please contact ICT program professional officer: Arthur Loo at [weeeyeong.loo@singaporetech.edu.sg](mailto:weeeyeong.loo@singaporetech.edu.sg) and Thiru at [Thiru@singaporetech.edu.sg](mailto:Thiru@singaporetech.edu.sg).

For your CodeDr account, please get it from your lab instructor in the lab. Please remember your user name and password for the future usage of the system. And, you are not allowed to change the password.

In this lab, you need to finish three tasks in CodeDr system, including the [Lab2\\_Task1](#), [Lab2\\_Task2](#) and [Lab1\\_Task3](#).

P.S. CodeDr system will provide you immediate feedback about the correctness of your program. You can view the feedback of your system by clicking the [view detail](#) beside your grade. CodeDr adopts a test cases-based approach to check your program. Your grade is depending on the number of test cases that you program can pass. Note that to train you to have a good programming practice, you have to write your program strictly according to the requirement of the tasks, including your input and output format. If there is any difference (even one more or less space), your program will fail on the test cases. SO TRAIN YOURSELF TO BE AN EXACT THINKER!

To help you practice, you are allowed to do multiple attempts for each task. Enjoy your learning!

You can use either the Python packaged IDLE (Python GUI) or PyCharm to create the Python program. To create one new program, you can create a new file by clicking the [NEW FILE](#) under the [FILE](#) menu. See below figure.

