Week 7 Computer Seminar - Lists & Dictionaries

Lists

- Like strings the values stored in a list can be accessed using:
  - indexing ([ ])
  - slicing operations ([:])
  - indexes start at 0.

Exercise 1

Complete the following questions to check your understanding.

```
list_a = ['abc' , 100, 3.14, 'def' ]
```

Self-Check Questions		**Answer**
A.	Write the code to print the first element of **list_a**	
B.	Write the code to print the **list_a** elements starting from the 2nd element until the 3rd element	
C.	Write the code to insert a new item to the list. The item is **35** and should be inserted at index 2 to produce: `['abc', 100, 35, 3.14, 'def']`	
D.	Write the code to insert 90 to the end of the list to produce: `['abc', 100, 35, 3.14, 'def', 90]`	

Exercise 2

Write a program that, for every element in the list [’how’, ’why’, ’however’, ’where’, ’never’] prints:
• a star symbol ‘*’
• the first two letters from the element
• the whole element i.e., producing something like:

* ho how
* wh why
* ho however
* wh where
* ne never

Exercise 3

Nested lists - Write the code to access the value 20 from the following list.

```
list_b = ["Orange", [10, 20, 30], [5, 15, 25]]
```

Dictionaries

- A dictionary maps keys to values. Dictionaries are indexed by keys.
- If you store using a key that is already in use, the old value associated with that key is

forgotten/overwritten.

- Dictionary methods:
  - keys() - Returns a view of the keys in the dictionary
  - values() - Returns a view of the values in the dictionary
  - get() - Returns the value associated with key
- To check whether a key is in the dictionary, use the **in** keyword.

Exercise 4

Complete the following questions to check your understanding. Example of telephone extension numbers stored in a dictionary.

```
tel = { 'Andrea' : 4351, 'Gill' : 3506 }
```

	Self-Check Question	**Answer**
A.	Write the code to add 'Ivan' : 3483 to the dictionary. Then print the directory.	
B.	Write the code to print the extension number for Andrea.	
C.	Write the code to print all the keys used in the dictionary.	
D.	Replace the extension number for 'Ivan' with 4422. Then print the dictionary.	
E.	• What would happen if you try to access tel['Roy']? • What method could you use to return 'None' rather than produce an error?	
F.	Write the code to print all the values used in the dictionary.	
G.	Write the code to check if the key 'Andrea' is in the dictionary. Print out the result.	

Exercise 5 - Extended Exercise

A local pizzeria that specialises in thin pizza bases requests a program that will allow customers to pre-order their pizza toppings. Build a program that creates and modifies a list for the pizza toppings, uses built-in list operations, and while and for loops.

The program will handle the following operations (example of large form of menu):

- a: Add a topping
- r: Remove a topping
- o: Order the current pizza
- q: Quit the program
- s: Start the current pizza over

Pseudocode:

```
Print Welcome message and instructions.
Print large form of menu

Loop forever
    Show short form of menu
    Ask the user what they want to do
    Convert to lowercase
```

```
If "a"
    ask user what topping to add, add it
Else if "r"
    Ask user what topping to remove
    Remove that topping if found
Else if "o"
    Show current pizza toppings. Thank user
    quit
Else if "q"
    quit
Else if "s"
    Reset to starting state
Else
    Tell user we did not understand
Show the current pizza toppings
```

Output: – The output of a typical run could look like this:

```
Welcome to Cavendish Pizzeria - choose your toppings.

When prompted, enter first letter or full word of operation.
---- Operations ----
a   Add a topping
r   Remove a topping
o   Order the pizza
q   Quit
s   Start over

What would you like to do?
   add, remove, order, quit, startover: a
Type in a topping to add: mushrooms
The toppings on your pizza are:
   mushrooms
What would you like to do?
   add, remove, order, quit, startover: a
Type in a topping to add: pineapples
The toppings on your pizza are:
   Mushrooms pineapples
What would you like to do?
   add, remove, order, quit, startover: uvwxyz
I'm not sure what you said, please try again.
The toppings on your pizza are:
   mushrooms pineapples
What would you like to do?
   add, remove, order, quit, startover: a
Type in a topping to add: bacon
The toppings on your pizza are:
   Mushrooms pineapples bacon
What would you like to do?
   add, remove, order, quit, startover: r
What topping would you like to remove: pineapples
The toppings on your pizza are:
   Mushrooms bacon
 What would you like to do?
   add, remove, order, quit, startover: o
```

```
The toppings on your pizza are:
   Mushrooms bacon
Thanks for your order!
```

Additional Information:

- The program will use a list to hold the user's topping choices. This starts off as an empty list to represent a pizza with no toppings on it.

```
toppingsList = [ ]
```

- The program will manipulate the list.
  1. User enters 'a' (to **a**dd a topping).
     - Ask user what topping to add, then add it to the end of the list with the **append** operation.
  2. User enters 'r' (to **r**emove a topping).
     - Ask user what topping to remove. The program should check if that topping is **in** the list, and if so use the **remove** operation to remove it.
  3. User enters 'o' (to **o**rder the pizza).
     - Print the toppingsList and then set it back to an empty list (Hint: toppingsList = [ ]).
  4. User enters 'q' (to **q**uit).
     - Use a **break** to exit any loop.
  5. User enters 's' (to **s**tart over).
     - Reset toppingsList to an empty list (Hint: toppingsList = [ ])

Now write the Python code for the extended exercise.

Exercise 6 – New Menu item: Change a topping

Add a new item to the menu. User enters 'c' (to **c**hange a topping). Get topping to change & topping to replace it. First use the **in** operator to ensure that the topping to be changed exists in the list of toppings. If so, replace the old topping with the new topping by using the index of where the old topping was found.

Exercise 7 – Add a user-defined function

Attempt this is you are confident with user-defined functions. If not, don't worry as we will cover user-defined functions in detail next week!

The code to print the items in the toppingsList[] is repeated several times in the program. Move this code to a user-defined function and pass the toppingsList[] as a parameter. Call the function where appropriate. Write the function so that if there are no toppings, the function prints that. Otherwise, it uses a for loop to iterate through the list of the pizza toppings and prints each one on the same line. See Pseudocode sample.

```
Function To Show Pizza Toppings
    If there are no toppings, say there are none
    Else
        print each topping on the same line
```