**4CS001**

**Python Workshop 5: Data Structures**

These are instructions for your fifth Python workshop. These workshops are designed for you to be able to make a good start on in your weekly lab sessions, but you may need to take them home to finish. There are some tasks that will not have been covered in the lecture, these will require you to do some independent research!

When you have finished this workshop, please upload your work to Canvas. This way you can always check back later to see how you solved a problem.

**Select:** Start > All Programs > IDLE (Python GUI)

**Part 1**

**1.** Given the following list: nums = [23, 16, 14, 33, 19, 6, 1]

**(a)** Give the index values of all the odd numbers assuming zero-based indexing

**(b)** How many elements would be looked at when the list is traversed (from top to bottom) until the value 19 was found?

**2.** Which of the following lists are syntactically correct in Python?   
Try them out in IDLE to see if you were correct.

**(a)** [1, 2, 3, 'four']  
**(b)** [1, 2, [3, 4]]  
**(c)** [[1, 2, 3] ['four']]

**3.** What is the result of each of the following list operations performed on the following list? lst = [4, 2, 9, 1]

Try them out in IDLE to see if you were correct.

**(a)** lst[1]  
**(b)** lst.insert(2, 3)  
**(c)** del lst[3]  
**(d)** lst.append(3)

**4.** Perform a series of list operations on the following list:  
fruit = ['apple', 'banana', 'pear', 'cherry']

to produce this updated list:  
['Grapefruit', 'banana', 'Date', 'cherry', 'Orange']

**5.** Give the code to retrieve the maximum value of the second half of the following list: lst = [4, 21, 17, 7, 9, 1, 13, 45]

**6.** Which of the following are valid operations that could be used on the following tuples: t1 = (1, 7, 9) and t2 = (5, 10, 8, 3)

**(a)** len(t1)  
**(b)** t1 + t2  
**(c)** t1.append(t2)  
**(d)** t1.insert(t2)

**7.** For str1 = 'Hello World', answer the following

**a)** Give an instruction that prints the fourth character of the string.

**b)** Give an instruction that finds the location of the first occurrence of 'o'

**Part 2**

**1.** For a list of integers named nums [12, 56, 72, 33, 1, 7]

**(a)** Give a while loop that adds up all the values in nums.

**(b)** Write a for loop that adds up all the values in nums in which the loop variable is assigned each value in the list.

**(c)** Write a for loop that adds up all the elements in nums in which the loop variable is assigned to the index value of each element in the list.

**(d)** Write a for loop that displays the elements in nums backwards.

**(e)** Write a for loop that displays every other element in nums.

**2.** Give an appropriate list comprehension for each of the following:

**(a)** Producing a list of consonants that appear in a string variable

**(b)** Producing a list of numbers between 1 and 100 that are divisible by 3.

**(c)** Producing a list of temperatures in Fahrenheit when given a list of temperatures in degrees Celsius.

**3.** Define a function that can accept two strings as input and print the string with maximum length in console.

If two strings have the same length, then the function should print both strings.

**Part 3**

**1.** Write a Python program that prompts the user for a series of integers and stores in a list only the values between 1-100, and displays the resulting list.

**2.** Write a Python program that prompts the user for a list of integers and stores them in a list. For all values that are greater than 100, the string 'over' should be stored instead. The program should display the resulting list.

**3.** Write a Python program that prompts the user to enter a list of names and stores them in a list. The program should display how many times the letter 'a' appears within the list.

**4.** Write a program that accepts a comma separated sequence of words as input and prints the words in a sequence after sorting them alphabetically.

Suppose the following input is supplied to the program:

without, hello, bag, world

Then, the output should be:

bag, hello, without, world

**5.** Write a Python program that prompts the user to enter integer values to populate two lists, then print messages to determine the following:

**(a)** Whether the lists are of the same length.

**(b)** Whether the elements in each list sum to the same value.

**(c)** Whether there are any values that occur in both lists.

**Part 4 (Optional)**

**1.** Create a program that picks a winner for a contest or prize draw. The program should prompt for the names of the entrants and their contact details (address, mobile no etc.) until the user enters a blank string. Then select a winner at random and print their details. This is showcased in the demo program called *coin\_change* included in last week’s sample code.

b) Add a text based menu to the program with two initial options:  
Add Contestant or Quit with the relevant functions.

c) Add a menu option that allows users to view the current list of contests.

d) Add a menu option that allows users to delete contestants.