

## Homework\_Week1: How do we use Strings, Lists and Dictionaries?

**Q1. Write a Python program which accepts a sequence of comma-separated numbers from user and generate a list and a tuple with those numbers.**

### Pseudocode:

num= Get user input for numbers separated in ',' and split the numbers where ',' is found

a= Convert the input 'num' to list

b= Convert the list 'a' to tuple

print "Your list is 'a' and your tuple is 'b'"

### Python program:

```
#Q1. Write a Python program which accepts a sequence of comma-separated numbers from user  
#and generate a list and a tuple with those numbers.
```

```
num=input("Input numbers separated by a comma and press enter:").split(',')  
a=list(num)  
b=tuple(a)  
print (" Your list is: ", a, "\n" ,"Your tuple is:", b)
```

```
Input numbers separated by a comma and press enter: 1,2,3,4,5  
Your list is: ['1', '2', '3', '4', '5']  
Your tuple is: ('1', '2', '3', '4', '5')
```

**Q2. Write a Python program to accept a filename from the user and print the extension of that.**

### Pseudocode:

filename= Get user input for filenames separated with '.' and split the filename where '.' is found  
print the extension of the file and concatenate the represent of the filename with index -1

### Python program:

```
#Q2. Write a Python program to accept a filename from the user and print the extension of that.
```

```
filename = input("Input the Filename and press enter: ").split(".")  
print ("The extension of the file is : " + repr(filename[-1]))
```

```
Input the Filename and press enter: Maisha.pdf  
The extension of the file is : 'pdf'
```

**Q3. Write a Python program that accepts an integer (n) and computes the value of  $n+nn+nnn$ .**

**Pseudocode:**

n= Get user input for integer number

m\_series= calculate the summation of the increased power of n until the 3<sup>rd</sup> series in integer form

print m\_series which is the computation of the first three mathematical series.

**Python program:**

```
#Q3. Write a Python program that accepts an integer (n) and computes the value of n+nn+nnn.
```

```
n=input("Input an integer number and press enter:")
m_series= int(n)+int(n*2)+int(n*3)
print("The computation of the first three mathematical series is:", m_series)
```

```
Input an integer number and press enter: 5
The computation of the first three mathematical series is: 615
```

**Q4. Write a Python program to concatenate all elements in a list into a string and return it.**

**Pseudocode:**

elements= Get user input for all the elements you want to concatenate storing it to a list and split where ',' is found

print the elements by using join function

**Python program:**

```
#Q4. Write a Python program to concatenate all elements in a list into a string and return it.
```

```
elements = list(input("Write down all the elements you want to *join* separated by a *,* without spaces:").split(","))
print("".join(elements))
```

```
Write down all the elements you want to *join* separated by a *,* without spaces: 1,2,3,4,5
12345
```

**Q5. Write a Python program to sum all the items in a dictionary.**

**Pseudocode:**

Predifine dictionary car\_price equals 'Mustang':40000,'Hyundai':35000,'Honda':10000

Print out the sum of all the values in the dictionary car\_price.

### Python program:

```
#Q5. Write a Python program to sum all the items in a dictionary.  
  
car_price = {'Mustang':40000, 'Hyundai':35000, 'Honda':10000}  
print("Total price of all cars: $", sum(car_price.values()))  
  
Total price of all cars: $ 85000
```

### Automated and Extended Version:

**#Write a Python program to sum all the values in a given mixed set of dictionaries.**

#### Pseudocode:

Predefine a set of six dictionaries.

Put all the dictionaries in a new list.

Define a function (sum\_of\_dict) that takes dictionaries and adds/concatenates its values depending on whether they are int or string- inside which:

initialize variables

run through values in dictionary and use error handling (using try and except) in the for loop

Inside the for loop:

-force values to be int type and sum them

-if values cannot be forced to int type, force them to be str type and store them in a list

convert list of strings to a string with '+' as delimiter

if: sum is 0 (i.e., there were no numbers in dictionary) return string

elif: string is empty (i.e., there were no non-numbers in dictionary) return sum

else: if both are non-zero/non-empty then return both

Define the second function (value\_printer) that takes in dictionary and prints a nice string using the function (sum\_of\_dict) to get values

Define the third function (dict\_print\_auto) that takes the given list of dictionaries and prints out their values

Call the function (dict\_print\_auto) on (new\_list) to print out all the summations of the values from the mixed set of dictionaries.

## Python program:

```
#Write a Python program to sum all the values in a given mixed set of dictionaries.      #Maisha Farzana Version.
#Master list of dictionaries:
#define dictionaries
dict_1 = {'Mustang':40000,'Hyundai':35000,'Honda':10000}
dict_2 = {"Brand": "Ford", "Model": "Mustang", "Year": "New"}
dict_3 = {1:"100", 2:"200", 3:"300"}
dict_4 = {1:100, 2:200, 3:300}
dict_5 = {1:"abc", 2:"xyz"}
dict_6 = {'A':"abc", 'B':"100", 'C':"5", 'D':"xyz", 'E': "200"}

new_list = [dict_1, dict_2, dict_3, dict_4, dict_5, dict_6]
```

```
#function that takes dictionaries and adds/concatenates its values depending on whether they are int or string
def sum_of_dict(x):
    #initialize variables
    sum1 = 0
    string1 = ''
    string2 = ""
    list1 = []
    list2 = []
    #run through values in dictionary
    for value in x.values():
        #force values to be int type and sum them
        try:
            value = int(value)
            sum1 = sum1 + value
        #if values cannot be forced to int type force them to be str type and store them in a list
        except ValueError:
            value = str(value)
            list1.append(value)
    #convert list of strings to a string with + as delimiter
    string1 = "+".join(list1)
    #if sum is 0 (i.e there were no numbers in dictionary) return string
    if sum1 == 0:
        return string1
    #if string is empty (i.e. there were no non-numbers in dictionary) return sum
    elif string1 == "":
        return sum1
    #if both are non-zero/non-empty then return both sum and strings
    else:
        for value in x.values():
            value = str(value)
            list2.append(value)
        string2 = "+".join(list2)
    return sum1, string1, string2
```

```
#function that takes in dictionary and prints a nice string using sum_of_dict to get values
def value_printer(x):
    value = sum_of_dict(x)
    if type(value) == int or type(value) == str:
        string = "The sum of the values in {} is {}".format(x, value)
        return string
    string = "The sum of the values in {} are as follows:\nThe sum of the numeric values is {} \n\nThe sum of the non_numeric values is {} \nAnd the sum of all the values in the dict is {}".format(x,value[0], value[1], value[2])
    return string
```

```
#function that takes the given List of dictionaries and prints out their values
def dict_print_auto(new_list):
    for dictionary in new_list:
        print(value_printer(dictionary))
```

```
#call the function dict_print_auto on new_list
dict_print_auto(new_list)
```

```
The sum of the values in {'Mustang': 40000, 'Hyundai': 35000, 'Honda': 10000} is 85000
The sum of the values in {'Brand': 'Ford', 'Model': 'Mustang', 'Year': 'New'} is Ford+Mustang+New
The sum of the values in {1: '100', 2: '200', 3: '300'} is 600
The sum of the values in {1: 100, 2: 200, 3: 300} is 600
The sum of the values in {1: 'abc', 2: 'xyz'} is abc+xyz
The sum of the values in {'A': 'abc', 'B': '100', 'C': '5', 'D': 'xyz', 'E': '200'} are as follows:
The sum of the numeric values is 305
The sum of the non_numeric values is abc+xyz
And the sum of all the values in the dict is abc+100+5+xyz+200
```