Navrachana University
School of Engineering and Technology
BSc-Data Science
Course: DSC 302 Fundamentals of Data Science
Programming Assignment

CORE PYTHON :

1. What is 7 to the power of 4?

2.Split this string:

    s = "Hi there Sam!"

into a list.

3. Given the variables:

planet = "Earth"

diameter = 12742

Use .format() to print the following string:

The diameter of Earth is 12742 kilometers.

4.Given this nested list, use indexing to grab the word "hello”

lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]

5. Given this nested dictionary grab the word "hello".

d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}

6.What is the main difference between a tuple and a list?

7.Create a function that grabs the email website domain from a string in the form: **

user@domain.com

So for example, passing "user@domain.com" would return: domain.com

8.Create a basic function that returns True if the word 'dog' is contained in the input string. Don't worry about edge cases like a punctuation being attached to the word dog, but do account for capitalization. :

findDog('Is there a dog here?')

9.Use lambda expressions and the filter() function to filter out words from a list that don't start with the letter 's'. For example:**

seq = ['soup','dog','salad','cat','great']
should be filtered down to:
['soup','salad']

10.You are driving a little too fast, and a police officer stops you. Write a function to return one of 3 possible results: "No ticket", "Small ticket", or "Big Ticket". If your speed is 60 or less, the result is "No Ticket". If speed is between 61 and 80 inclusive, the result is "Small Ticket". If speed is 81 or more, the result is "Big Ticket". Unless it is your birthday (encoded as a boolean value in the parameters of the function) -- on your birthday, your speed can be 5 higher in all cases.

11 Write a program to input roll no, student name, marks of physics, chemistry and maths out of 100. (0-100). Calculate total, percentage, calculate STATUS (pass, fail) if students scores above 40 in all the 3 subjects the STATUS should be pass otherwise fail. Calculate GRADE if STATUS is pass.

Grade must be based on percentage value
if percentage is above 70, then grade must be DISTINCTION if percentage is above 60, then grade must be FIRST CLASS
if percentage is above 50, then grade must be SECOND CLASS if percentage is above 40, then grade must be PASS CLASS

LIST & DICTIONARY ASSIGMENT :

1. Write a python program to access following data structure
data=[(12345,"Diksha ","Diksha r@nuv.ac.in"),(45667,"Jiva","Jiva@nuv.ac.in"),(16789,"Ronit","Ronitd@nuv.ac.in"),(69433,"Heena","Heena@nuv.ac.in")]

2. Write a python program to access following dictionary
Mydict =
{"Enrollment Id" : [12345,45667,16789,69433],

"Student Name : ["Diksha ","Jiva","Ronit","Heena"],
"Email Id ": ["Diksha r@nuv.ac.in","Jiva@nuv.ac.in","Ronitd@nuv.ac.in",
"Heena@nuv.ac.in"] }
print(f"{enrolid:<7}{sname:12}{emailid:25}")

3. Write a python program to convert data in nested list structure to dictionary.

eg. data=[(12345,"Diksha ","Diksha r@nuv.ac.in"),(45667,"Jiva","Jiva@nuv.ac.in"),(16789,"Ronit","Ronit@nuv.ac.in"),(69433,"Heena","Heena@nuv.ac.in")]

Provide key such as Enrollment Id, Student Name, and Email id.
Resultant Data

mydict={"Enrollment Id" : [12345,45667,16789,69433],
"Student Name : ["Diksha ","Jiva","Ronit","Heena"], "Email Id " :["Diksha@nuv.ac.in","Jiva@nuv.ac.in","Ronitd@nuv.ac.in","Heena@nuv.ac.in"]
}
Display data in a tabular format.

4. Write a menu driven Object Oriented program to store Students details like Enrollment No, Student Name and Contact Number permanently using dictionary data structure.

Add functions to add, update, delete and display data. Save data permanently in a JSON file.
HINT:
To import json library

import json
To write any object to python fw=open("students.json","w")
jsndata = json.dumps(dictionary of students details)#dumps converts dictionary to json fw.write(jsndata)
To read any json object from file fr=open("students.json","r")
allStudents=json.load(fr) #reads whole file and returns the data.

5. A data.csv file has following data(stateid, name, population in crores, No. of Universities)

data.csv

12001,Gujarat,10,29,12002,Maharashtra,19,36,12003,Rajasthan,13,31,12004,Madhya Pradesh,14,21,12005,Punjab,12,13,12006,Karnataka,23,31,12007,Tamil nadu,25,29,1208,Keral a,21,15

Read data.csv file. Transfer data to data structure like following
States={
12001: {"name":"Gujarat","population":10,"no_of_uni":20},
12002: {"name":"Maharashtra","population":19,"no_of_uni":36}
}
Using this data, construct dictionary in following manner.

OOPs:

1. Write a OOP in python to input empid, name, basic salary, no. of experience in yrs. Calculate hra(35% of basic), da (58% of basic) and pf (9.5% of basic). Also calculate bonus based on experience in years. If experience in years is >= 30, bonus must be 59% of basic,

If experience in years is >=23, bonus must be 51% of basic,
If experience in years is >=15, bonus must be 45% of basic,
If experience in years is >=7, bonus must be 33% of basic,

If experience in years is <7, bonus must be 16% of basic
Calculate netsalary as basic+da+hra-pf+bonus.
Create a class, constructor to create instance variables, getter-setter for each variable,
calculative functions for operative variables. A class methods/function should not contain
display specific and input specific code. Such code should be added in driver part of
python program.

2. Write an OOP based Python program which inputs n numbers in a list from keyboard. If 8
numbers are inputted, calculate sum of 0th and 7th element and save it in another list say
newlist[0], sum of 1st and 6th to newlist[1], sum of 2nd and 5th to newlist[2], sum of 3rd and 4th
to newlist[3] and so on. Remove duplicates from newlist if any by converting to set. Later
convert it to tuple and display.

3. Write an OOP program to perform addition, base and power, concatenation, max, min of two
numbers stored in two different objects created from same class.

```
n1=MyNumber(2)
n2=MyNumber()
n2.setNum(5)
n3=n1.add(n2)
print("Addition is ",n3.getNum()) #7
n3=n1.raisedTo(n2)
print(n1.getNum()," raised to ",n2.getNum()," is ",n3.getNum()) #32
n3=n1.concat(n2)
print("Concat answer is ",n3.getNum()) #25
```

```
    n3=n1.max(n2)
    print("Max is ",n3.getNum())
```

NUMPY EXERCISE 1 :

1. Import Numpy as np
2.Create an array of 10 zeros
3.Create an array of 10 ones
4.Create an array of the integers from 10 to 50
5.Create an array of all the even integers from 10 to 50
6.Create a 3x3 matrix with values ranging from 0 to 8
7. Create a 3x3 identity matrix
8. use NumPy to generate a random number between 0 to 1
9. Use Numpy to generate an array of 25 random numbers sampled from a standard normal distribution
10. Create a matrix of 10x10 matrix of linearly spaced elements from 0.01 to 1
11. Create an array of 20 linearly spaced points between 0 and 1
12.Create a 5x5 matrix as

```
array([[ 1,  2,  3,  4,  5],
    [ 6,  7,  8,  9, 10],
    [11, 12, 13, 14, 15],
    [16, 17, 18, 19, 20],
    [21, 22, 23, 24, 25]])
```

find following using slicing

a.
```
array([[12, 13, 14, 15],
          [17, 18, 19, 20],
        [22, 23, 24, 25]])
```

b. 20

c.array([[2],

```
                  [ 7],
                  [12]])
 d.array([21, 22, 23, 24, 25])
 e.array([[16, 17, 18, 19, 20],
              [21, 22, 23, 24, 25]])
```

13. Get sum of all values in matrix
14. Get sum of all the columns in matrix
15. Get Standard deviation , varience , minimum, maximum from matrix row wis.
16. create 2 d matrix with all border values nan and

NUMPY EXERCISE 2 :

1. Create 1-D, 2-D and 3-D numpy array
2. Print data type of array.
3. Print shape and dimensions of all the three different array.
4. Create another 1-D array of 12 elements filled with zero
5. Create another 2-D array elements filled with ones.
6. Create a 1-D array known as Maths Marks out of 100 having 20 elements.
7. Sort the above array in ascending and descending order.
8. Create a copy of sorted array to another variable by deep copying.
9. Create 2-D array of 5x5 which is an identity matrix and then convert 1D .
10. Create 2-D array of 3x3. Add one more row to it.
11. Split above array created vertically and horizontally.
12. Create 4x4 identity matrix. Add 4 to each element.
13. Create 1-D array. Initialize with floating point values. Display its ceil, floor and round (with 2 decimal’s).
14. Create 5x5 matrix. Display min, max, variance and standard deviation column-wise.
15. Create a null vector of size 10 with 6th value filled with 200.

16. Create a 1-D Array/Vector with values ranging from 100 to 150 in sequence.
17. Reverse the above vector.
18. Create a 5x5 matrix with values ranging from 0-24 using some function.
19. Find indices of nonzero elements from [1,0,2,0,0,4,0,8]
20. Create a 10x10 matrix with 1 on the borders and 0 inside.
21. Create 8x8 matrix and fill with 1 and 0 in checkerboard pattern.
22. Create two 3x3 matrix with random integer values. Perform matrix addition, multiplication and subtraction