**TASK 1: [Use the below 2 data sources to complete the task]**

**Dataset 1**

WHO dataset to get social economic health data by country

<https://data.world/resiport/who-dataset>

**Dataset 2**

COVID dataset

<https://www.kaggle.com/imdevskp/corona-virus-report?select=country_wise_latest.csv>

- **Select** country-wise-latest.csv

**Programming Lang to be used:**

Python

**Can use one of the below libraries:**

1. pandas or
2. pyspark.

**Please note:** This is not a data-science based project. So you can avoid generating graphs or other data-science specific analysis while doing the project.

---------------------------------------------------------------------------------------------

**Answer the below 10 questions:**

1. Explain why you chose that python library (pandas or pyspark)
2. Create a simple OOPs architecture using classes and methods to do the project.
3. Document the project with docstring. Write comments wherever necessary
4. Using the asked technologies and libraries, fetch the data with API calls.
5. After fetching the data read the data:
   * 1. Read the COVID Dataset normally
     2. Assume the data from WHO is very huge with millions of records. Modify your read statement so that the file is read without memory issues.
6. Clean the datasets for nulls and other problems you think it might have
7. Select appropriate columns from each dataset
8. Merge them on their keyfields
9. Create a computed column, "Recovered\_Percent" that lists the patients recovered from confirmed cases. Additionally, please create another computed column of your choice.
10. Make sure of the below points after completing the project (say yes or no)
    * 1. Did you do docstring documentation ?
      2. Did you use OOPs architecture using classes and methods to complete the project ?

**TASK2:decorators**

class my\_cls\_info:

def \_\_init\_\_(self, f):

print("CLASS INIT")

self.func = f

self.arg = ""

def \_\_call\_\_(self, \*args):

self.arg = args

print("CLASS CALL")

print("FUNCTION NAME: ", self.func.\_\_name\_\_)

#return self.func(\*args)

def get\_no\_chars(self):

y = self.func()

if type(y)==str:

print("NO OF CHARS IN STR: ",len(y), "\nINFO:", y)

else:

print("RETURNED INFO: ", y)

def add\_bonus(self):

args = self.arg

z = self.func(args[0],args[1])

if args[0]>1000:

print("NET SALARY WITH BONUS: ", z+777)

else:

print("NET SALARY WITHOUT BONUS: ", z)

# -------------------------------------------------

def my\_func\_nod():

msg = "I like decoration. But I don't have a decorator. Sad!"

print("Call from func: ", msg)

return msg

def my\_func\_wod():

msg= "I don't like decorators. OMG!"

print("Call from func: ", msg)

return msg

@my\_cls\_info

def my\_func\_wd():

msg = "I like decoration. I have a decorator. Lucky!"

print("Call from func: ", msg)

return msg

def salary\_person(base\_salary,tax\_percent):

tax\_amt = base\_salary\*tax\_percent

net\_salary=base\_salary-tax\_amt

return net\_salary

# ------------------------------------------------------------------------------------------------

**With reference to above code, answer the below questions:**

print("\nSCENARIO 1", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

**# Explain the below code statements**

# ------------------------------------------------------------------

my\_func\_res1 = my\_cls\_info(my\_func\_nod)  **# CODEBLK 96**

my\_func\_res1.get\_no\_chars()

print("\nSCENARIO 2", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

**# Explain the functionality of my\_func\_wd in this code. Explain the output.**

# ------------------------------------------------------------------

my\_func\_wd() **# CODEBLK 98**

print("\nSCENARIO 2a", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

**# Explain the functionality of my\_func\_wd in this code. Explain the output.**

my\_func\_wd.get\_no\_chars() **# CODEBLK 100**

print("\nSCENARIO 2b", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

**# Note: This code line below is similar to CODEBLK 100. And yet it throws error ? Why ?**

try:

my\_func\_nod.get\_no\_chars() **# CODEBLK 101.**

except Exception as ex:

print("Exception raised: ", ex)

print("\nSCENARIO 3", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

**# This CODEBLK 102**  **below is similar to CODEBLK 100. And yet it throws error ? Why?**

**Please explain conceptually.**

try:

my\_func\_wd().get\_no\_chars() **# CODEBLK 102**

except Exception as ex:

print("Exception raised: ", ex)

# ------------------------------------------------------------------------------------------------

**#QUESTION 1**

**# The below CODEBLK 103** **returns a positive value of net salary after taxes.**

**# But we want to add $777 for salaries greater than 1000, as a bonus to the salary. We do not want to modify the current function.**

**# Use the concept of decorator and use the function "add\_bonus" in the class to achieve the desired result.**

salary\_person(2000,0.20) **# CODEBLK 103**

**#QUESTION 2**

**# Suppose you want to add "yrs of experience" as the third parameter to calculate bonus. If yrs\_experience > 10, then add $777**

**# Can you do it without modifying either the body or the arguments of the current function 'salary\_person' ? You can modify the "add\_bonus".**