Sindh Madressatul Islam University
Office of Research, Innovation & Commercialization (ORIC)

NAME : DIL ZAIB

ID : BCB-23S-007

SECTION : A

DEPARTMENT : CYBER SECURITY

SEMESTER : 3rd

INSTRUCTOR NAME : MISS AQSA UMER

QUESTION : 01 PYTHON BASIC (STRINGS,DATA TYPE,LOOPS,IF CONDITIONS,FUNCTION,CLASSES)

STRINGS :

Strings in Python are sequences of characters, typically used to represent text. They can include letters, numbers, symbols, and whitespace.

Strings are enclosed within either single (' '), double (" "), or triple (''' ''' or """ """) quotes.

1) **str1 = "Hello" str2 = "World" result =**

str1 + " " + str2 print(result) # Output:

Hello

World

2)

 # Take input from the user user_input = input("Enter

a string: ")

 # Print the input string print("You entered:",

user_input)

DATA TYPE :

A data type in programming defines the type of data that a variable can hold. It specifies the type of values that can be assigned to the variable and the operations that can be performed on those values. Here's a simple definition: 1) num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: "))

 # Calculate the sum sum = num1

+ num2

 # Print the result print(num1 + num2)

2) **frist_number =input("ENTER YOUR**

NUMBER: ") operator =input("ENTER YOUR OPERATOR:(+,-,*,/,%) ") second_number

=input("ENTER YOUR SEC_NUMBER: ")

frist_number=int(frist_number)

```
second_number=int(second_number)


if  operator == "+":
   print(frist_number + second_number) elif
operator == "-":
   print(frist_number - second_number) elif  operator
== "*":
   print(frist_number * second_number) elif
operator == "/":
   print(frist_number / second_number)
elif  operator == "%":
print(frist_number % second_number)
print("INVALID OPERATION")
```

LOOPS:

A loop in programming is like a repetitive task that a computer performs over and over again until a certain condition is met. Imagine you're folding shirts: you pick up a shirt, fold it, put it down, and repeat until all shirts are folded. In a loop, you tell the computer to repeat a certain block of code multiple times, and it keeps doing that until a specific condition is no longer true.

1)

```
# CONNECTION OF NODES
class Node:       def __init__(self,
data):        self.data = data
self.next = None

node1 = Node(1) node2 =
Node(2) node3 = Node(3) node4
= Node(4)

node1.next = node2 node2.next =
node3 node3.next = node4
```

```
current = node1

while current is not None:
print(current.data, end=" -> ")     current = current.next

print("None")
```

2)

```
# Using a for loop to print numbers from 1 to 5
print("Using a for loop:") for i in range(1, 6):
   print(i)

# Using a while loop to print numbers from 1 to 5
print("\nUsing a while loop:") num = 1 while num <=
5:     print(num)     num += 1
```

IF CONDITIONS:

Sure! Let's break down if-else conditions in simple terms:Imagine you have a decision to make based on a condition. If the condition is true, you'll do one thing, and if it's false, you'll do something else. That's where if-else statements come in handy.

1)

```
num = int(input("Enter number to check even or odd : "))

if num % 2 == 0:      print("The
number is even.") else:
   print("The number is odd.")
```

2)

```
number = 9


if number > 10:      print("Number is
greater than 10") else:

   print("Number is not greater than 10")
```

FUNCTIONS :

Imagine you have a set of instructions that you need to perform repeatedly. Instead of writing the same instructions over and over again, you can put them together and give them a name. Then, whenever you need to perform those instructions, you can simply call that name.

1)

```
# Define a function called "greet" def greet():
   print("Hello, welcome!")

# Call the function "greet" greet()
```

2)

```
def add_numbers(a, b):
```

```
    sum = a + b

    return sum


result = add_numbers(3, 5)

print("The sum of the two numbers is:", result)
```

CLASSES:

In programming, a class is like a blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) that objects of that class will have.

1)

class Dog:

```
    def __init__(self, name, age):
     self.name = name        self.age =
age         def bark(self):
print(self.name + " says ACHA!")
my_dog = Dog("Buddy", 3)
print(my_dog.name)        print(my_dog.age)
my_dog.bark()
```

QUESTION 02:

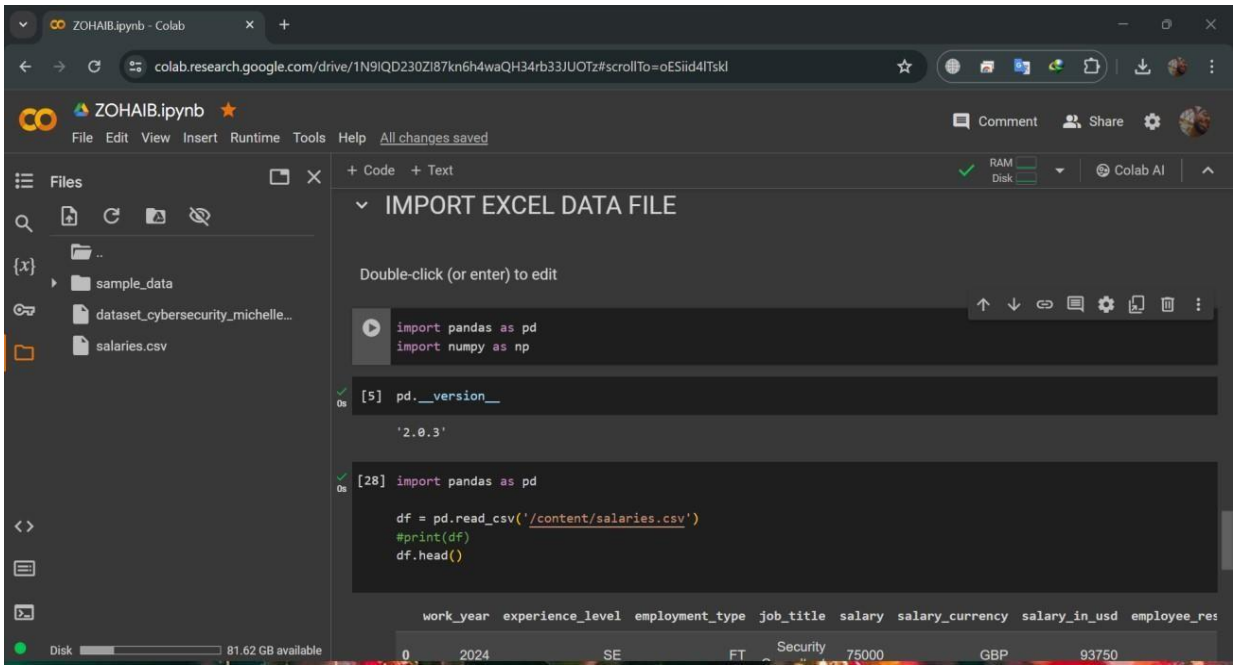
NUMPY AND PANDAS (THESE ARE LIBRARIES WE NEED TO CHECK ITS DOCUMENTATON)

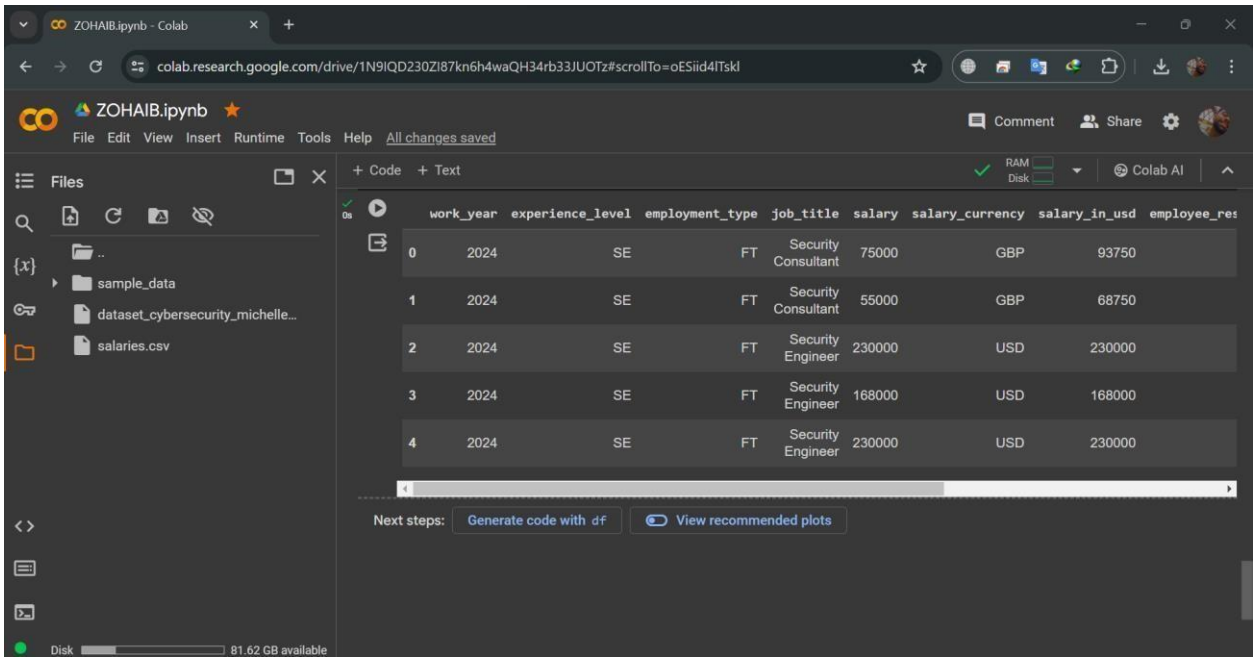
```
import pandas as pd
import numpy as np

[5]  pd.__version__
```

QUESTION 03:

DATASET DOWNLOAD (MAX TWO DATASETS OF THE TOPIC SOME HOW RELATED TO THE CYBER SECURITY)

(SOURCE: KAGGLE.COM)




ZOHAIB.ipynb - Colab
colab.research.google.com/drive/1N9IQD230ZI87kn6h4waQH34rb33JUOTz#scrollTo=oESiid4ITskI
ZOHAIB.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Colab AI
..
sample_data
dataset_cybersecurity_michelle...
salaries.csv
work_year experience_level employment_type job_title salary salary_currency salary_in_usd employee_res
0 2024 SE FT Security Consultant 75000 GBP 93750
1 2024 SE FT Security Consultant 55000 GBP 68750
2 2024 SE FT Security Engineer 230000 USD 230000
3 2024 SE FT Security Engineer 168000 USD 168000
4 2024 SE FT Security Engineer 230000 USD 230000
Next steps:
Generate code with df
View recommended plots
Disk
81.62 GB available