# **In-Lab (Lab2)**

## **Task 1**

Execute a simple python program to check the python installation and environment setup.

phonebook ={}

phonebook["Ali"] = {"Phone" : "03056663224",

"Email" : "ali@gmail.com" }

phonebook["Zain"] = {"Phone" : "0307666555",

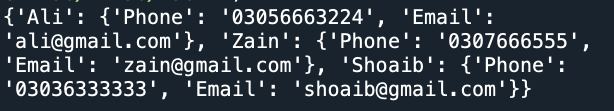
"Email" : "zain@gmail.com" }

phonebook["Shoaib"] = {"Phone" : "03036333333",

"Email" : "shoaib@gmail.com" }

print(phonebook)

After executing the program, you should see an output similar to the following image.



## **Task 2**

Execute a simple python program to check the python installation and environment setup.

phonebook ={}

phonebook["Ali"] = {"Phone" : "03056663224",

"Email" : "ali@gmail.com" }

phonebook["Zain"] = {"Phone" : "0307666555",

"Email" : "zain@gmail.com" }

phonebook["Shoaib"] = {"Phone" : "03036333333",

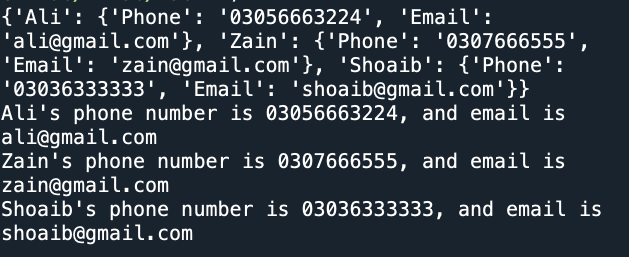
"Email" : "shoaib@gmail.com" }

print(phonebook)

for name, record in phonebook.items():

print("{}'s phone number is {}, and email is {}".format(name, record ["Phone"], record["Email"]))

After executing the program, you should see an output similar to the following image.



## **Task 3**

Execute a simple python program to check the python installation and environment setup.

phonebook = {}

phonebook [ "John"] = {"Phone": "012 794 794",

"Email":"john@email.com"}

phonebook [ "Jill"] = {"Phone": "012 345 345",

"Email": "jill@email.com"}

phonebook [ "Joss"] = {"Phone": "012 321 321", "Email": "joss@email.com"}

print (phonebook)

del phonebook [ "John"]

for name, record in phonebook.items():

print("{}'s phone number is {}, \

and their email is {}".format(name, record[ "Phone"], record[ "Email"]))

# Pop returns the record, and deletes it

jill\_record = phonebook.pop("Jill")

print(jill\_record)

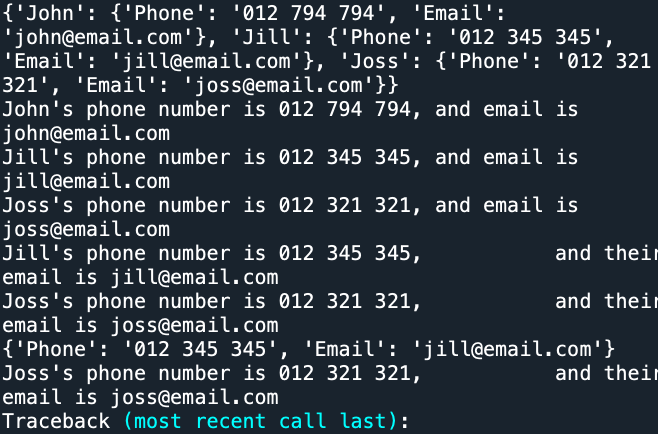
for name, record in phonebook.items():

print("{}'s phone number is {}, \

and their email is {}".format(name, record[ "Phone"], record[ "Email"]))

del phonebook [ "John"]

After executing the program, you should see an output similar to the following image.



## **Task 4**

Execute a simple python program to check the python installation and environment setup.

#Arithmatic Operators

number = 1 +2 \* 3 / 4.0

print(number)

remainder = 11 % 3

print(remainder)

# power

squared = 7 \*\* 2

print(squared)

cubed = 2 \*\* 3

print(cubed)

#%%

# List Operators

# ============================================================================================

even\_numbers = [2, 4, 6, 8]

uneven\_numbers = [1, 3, 5, 7]

all\_numbers = uneven\_numbers + even\_numbers

print(all\_numbers)

print([1, 2, 3] \* 3)

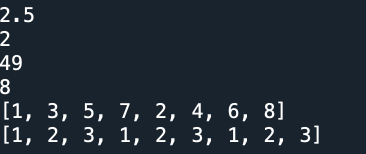
uneven\_numbers = [1, 3, 5, 7]

all\_numbers = uneven\_numbers + even\_numbers

print(all\_numbers)

print([1, 2, 3] \* 3)

After executing the program, you should see an output similar to the following image.



## **Task 5**

Execute a simple python program to check the python installation and environment setup.

# Define the two strings

greeting = "Hello, World!"

repeated\_hello = "Hello " \* 7

# Print the strings

print(greeting)

print(repeated\_hello)

After executing the program, you should see an output similar to the following image.



## **Task 6**

Execute a simple python program to check the python installation and environment setup.

x = 2

print(x == 2)

print(x == 3)

print(x < 3)

name = "John"

4

print(name == "John" and x == 2)

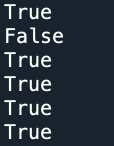
# Using `or`

print(name == "John" or name == "Jill")

# Using in on lists

print(name in ["John", "Jill", "Jess"])

After executing the program, you should see an output similar to the following image.



## **Task 7**

Execute a simple python program to check the python installation and environment setup.

x = 2

y = 10

if x > 2:

print("x > 2")

elif x == 2 and y > 50:

print("x == 2 and y > 50")

elif x < 10 or y > 50:

print("x < 10 or y > 50")

else:

print("Nothing worked.")

name\_list1 = ["John", "Jill"]

name\_list2 = ["John", "Jill"]

print (not (name\_list1 == name\_list2))

# Using `is`

name2 = "John"

print(name\_list1 == name\_list2)

print(name\_list1 is name\_list2)

print("x == 2 and y > 50")

elif x < 10 or y > 50:

print("x < 10 or y > 50")

else:

print("Nothing worked.")

name\_list1 = ["John", "Jill"]

name\_list2 = ["John", "Jill"]

print (not (name\_list1 == name\_list2))

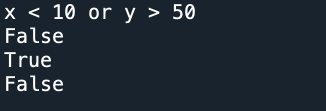
# Using `is`

name2 = "John"

print(name\_list1 == name\_list2)

print(name\_list1 is name\_list2)

After executing the program, you should see an output similar to the following image.



## **Task 8**

Execute a simple python program to check the python installation and environment setup.

numeric\_data = [10, 20, 30, 40, 50]

for number in numeric\_data:

result = number\*2 # Perform some operation (e.g., multiplication) print (result) Print the result

print(result)

#Sample string

text = "Hello, World!"

#Using a for loop to read and print each character in the string

for char in text:

print(char)

new\_text = ""

for char in text:

new\_text += char.upper() # Convert letters to uppercase else: new\_text += char #Keep non-letter characters as they are

print(new\_text)

#Writing Numeric Data

numeric\_data = []

for i in range(1,11):

numeric\_data.append(i)

print(numeric\_data)

#Sample string

text = "Hello, World!"

#Using a for loop to read and print each character in the string

for char in text:

print(char)

new\_text = ""

for char in text:

new\_text += char.upper() # Convert letters to uppercase else: new\_text += char #Keep non-letter characters as they are

print(new\_text)

#Writing Numeric Data

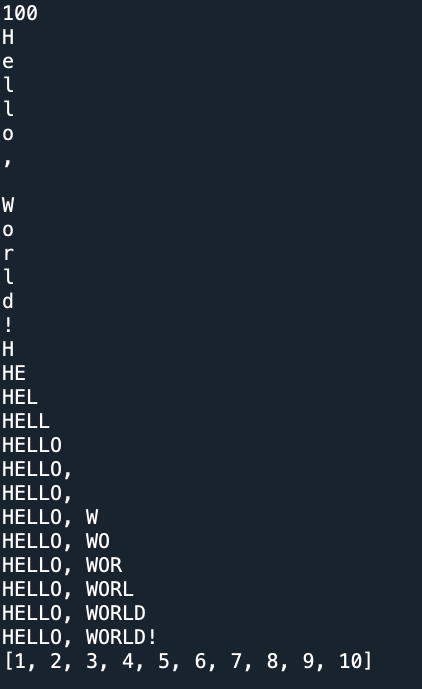
numeric\_data = []

for i in range(1,11):

numeric\_data.append(i)

print(numeric\_data)

After executing the program, you should see an output similar to the following image.



## **Task 9**

Execute a simple python program to check the python installation and environment setup.

count = 1

while count <= 5:

print(count)

count += 1

# 2. For Strings

# Using a while loop to print each character of a string text= "Hello"

text = "Hello"

index = 0

while index < len(text):

print(text[index])

index += 1

student\_grades = {"Alice": 92, "Bob": 85, "Charlie": 78}

keys = list(student\_grades.keys()) # Get the keys as a list

index = 0

while index < len (keys):

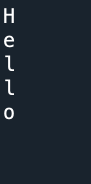
key = keys[index]

value = student\_grades[key]

print (f" (key): {value}")

index += 1

After executing the program, you should see an output similar to the following image.



# **Post-Lab (Lab2)**

## **Task**

Execute a simple python program to check the python installation and environment setup.

# Initialize a dictionary to store student names and grades

student\_grades = {}

# Define the number of students

num\_students = 7

# Input student names and grades

for \_ in range(num\_students):

name = input("Enter student name: ")

grade = float(input("Enter student grade: "))

student\_grades[name] = grade

# Calculate and display the average grade

total\_grade = sum(student\_grades.values())

average\_grade = total\_grade / num\_students

print(f"Average grade: {average\_grade:.2f}")

# Categorize each student's grade and display

for student, grade in student\_grades.items():

if grade >= 90:

category = "Excellent"

elif grade >= 80:

category = "Very Good"

elif grade >= 70:

category = "Good"

else:

category = "Needs Improvement"

print(f"{student}: {grade} ({category})")

# Search for a specific student's grade

while True:

search\_name = input("Enter student name to search for (or 'quit' to exit): ")

if search\_name.lower() == 'quit':

break

if search\_name in student\_grades:

print(f"{search\_name}'s grade: {student\_grades[search\_name]}")

else:

print("Student not found. Please enter a valid name.")

while True:

search\_name = input("Enter student name to search for (or 'quit' to exit): ")

if search\_name.lower() == 'quit':

break

if search\_name in student\_grades:

print(f"{search\_name}'s grade: {student\_grades[search\_name]}")

else:

print("Student not found. Please enter a valid name.")

After executing the program, you should see an output similar to the following image.

After executing the program, you should see an output similar to the following image.

