



Department of Electrical Engineering

Faculty Member: Sir Hassan Khaliq

Date: 15-Feb-2023

Semester:6th

EE357 Computer and Communication Networks

Lab 2: Programming in Python

Name	Reg. No	PLO5/ CLO3		PLO5/ CLO3	PLO5/ CLO3	PLO5/ CLO3
		Viva / Quiz / Lab Performance	Analysis of data in Lab Report	Modern Tool Usage	Ethics and Safety	Individual and Team Work
		5 Marks	5 Marks	5 Marks	5 Marks	5 Marks
Imran Haider	332569					
Muhammad Ahmed Mohsin	333060					
Amina Bashir	343489					



1 CONTENTS

2	Introduction	45
2.1	Objectives.....	45
3	Lab Conduct	45
4	Theory	45
5	Lab Task 1.....	46
5.1	Code	46
5.2	Output.....	47
6	Lab Task 2.....	47
6.1	Code	47
6.2	output	48
7	Lab Task 3.....	48
7.1	Code	48
8	Lab Task 4.....	50
8.1	code.....	50
8.2	output	51
9	Lab Task 5.....	51
9.1	Code:	51
9.2	Output:.....	52
10	Conclusion.....	52



1 INTRODUCTION

This laboratory exercise is meant to introduce the fundamental aspects of the python programming language which will be very important in the later labs of the course.

1.1 OBJECTIVES

The following are the main objectives of this lab:

- Create variables of different data types in python
- Use arithmetic and logical operations in python
- Implement conditional statements and loops in python
- Create functions and call them in python
- Implement lists and dictionaries in python
- Read and write to files in python

2 LAB CONDUCT

- Respect faculty and peers through speech and actions
- The lab faculty will be available to assist the students. In case some aspect of the lab experiment is not understood, the students are advised to seek help from the faculty.
- In the tasks, there are commented lines such as `#YOUR CODE STARTS HERE#` where you have to provide the code. You must put the code between the `#START` and `#END` parts of these commented lines. Do NOT remove the commented lines.
- Use the tab key to provide the indentation in python.
- When you provide the code in the report, keep the font size at 12

3 THEORY

Python is an interpreted language which is commonly used in many fields including networks, databases, robotics and artificial intelligence etc. It has an easy-to-learn syntax and is ideal for developing prototypes in a short amount of time. Python scripts are written in .py file which is then interpreted and executed by the python interpreter.

A brief summary of the relevant keywords and functions in python is provided below. (For more details, check the slides for this lab)



print()	output text on console
input()	get input from user on console
range()	create a sequence of numbers
len()	gives the number of characters in a string or list
if	contains code that executes depending on a logical condition
else	connects with if and elif , executes when conditions are not met
elif	equivalent to else if
while	loops code as long as a condition is true
for	loops code through a sequence of items in an iterable object
break	exit loop immediately
continue	jump to the next iteration of the loop
def	used to define a function

4 LAB TASK 1

Write a program that prompts the user for two numbers as input. Then, the program must compare the two numbers and print if they are equal or not. If the numbers are not equal, it must also print which number is greater (or lesser) than the other. The syntax for conditional statements is given below:

4.1 CODE

```
# Prompt the user for two numbers
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Compare the two numbers and print the result
if num1 == num2:
    print("The two numbers are equal.")
else:
    if num1 > num2:
        print("The first number is greater than the second number.")
    else:
        print("The second number is greater than the first number.")
```



4.2 OUTPUT

```
[1] # Prompt the user for two numbers
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Compare the two numbers and print the result
if num1 == num2:
    print("The two numbers are equal.")
else:
    if num1 > num2:
        print("The first number is greater than the second number.")
    else:
        print("The second number is greater than the first number.")
```

Enter the first number: 90
Enter the second number: 99
The second number is greater than the first number.

5 LAB TASK 2

Create a list with the sequence 1, 2, 3... 20. Then using the slice operation (:) on this list, print the following sub-lists:

5, 6, 7... 20

1, 2, 3... 12

7, 8, 9 ... 16

4, 5

11, 12, 13, 14

5.1 CODE

```
# Create a list with the sequence 1, 2, 3...20
my_list = list(range(1, 21))
print(my_list)

# Use slicing to print the requested sub-lists
print(my_list[4:])

print(my_list[:12])

print(my_list[6:16])
```



```
print(my_list[3:5])  
  
print(my_list[10:14])
```

5.2 OUTPUT

```
# Create a list with the sequence 1, 2, 3...20  
my_list = list(range(1, 21))  
print(my_list)  
  
# Use slicing to print the requested sub-lists  
print(my_list[4:])  
  
print(my_list[:12])  
  
print(my_list[6:16])  
  
print(my_list[3:5])  
  
print(my_list[10:14])  
  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]  
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]  
[7, 8, 9, 10, 11, 12, 13, 14, 15, 16]  
[4, 5]  
[11, 12, 13, 14]
```

6 LAB TASK 3

Write a function that takes 2 lists as arguments. Both the lists must be of the same length. The function should calculate the product of the corresponding items and place them in a third list. You must NOT use the product operator (*). You need to provide the function definition and the function call in the code. (Hint: You need to make use of loops in your function.) The function definition syntax is given as follows:

6.1 CODE

```
def product_lists(list1, list2):  
    """  
    Takes two lists of the same length as arguments,  
    calculates the product of the corresponding items,  
    and returns a third list containing the products.  
    """
```



```
# Initialize an empty list to hold the products
product_list = []

# Iterate over the lists and calculate the products
for i in range(len(list1)):
    product = 0
    for j in range(list1[i]):
        product += list2[i]
    product_list.append(product)

# Return the list of products
return product_list
```

```
list1=[10,5,6,90]
list2=[12,17,19,14]
result_list=product_lists(list1,list2)

#Now we print the list
print(result_list)
```

6.2 OUTPUT

```
[5] def product_lists(list1, list2):
    """
    Takes two lists of the same length as arguments,
    calculates the product of the corresponding items,
    and returns a third list containing the products.

    """
    # Initialize an empty list to hold the products
    product_list = []

    # Iterate over the lists and calculate the products
    for i in range(len(list1)):
        product = 0
        for j in range(list1[i]):
            product += list2[i]
        product_list.append(product)

    # Return the list of products
    return product_list
```

```
list1=[10,5,6,90]
list2=[12,17,19,14]
result_list=product_lists(list1,list2)

#Now we print the list
print(result_list)

[120, 85, 114, 1260]
```



7 LAB TASK 4

In this task, you will make use of dictionaries. Write a program that first prompts the user to input five strings which will be the keys of the dictionary. Then, the program must prompt the user to input the values of the respective keys. When entering the values, the user must be shown the key whose value is being input. Once all values are entered, display the dictionary.

7.1 CODE

```
# Initialize an empty dictionary
my_dict = {}

# Prompt the user to input five keys
for i in range(5):
    key = input("Enter key " + str(i+1) + ": ")
    my_dict[key] = None

# Prompt the user to input the values for each key
for key in my_dict:
    value = input("Enter the value for " + key + ": ")
    my_dict[key] = value

# Display the resulting dictionary
print("Resulting dictionary:")
print(my_dict)
```




7.2 OUTPUT

```
# Initialize an empty dictionary
my_dict = {}

# Prompt the user to input five keys
for i in range(5):
    key = input("Enter key " + str(i+1) + ": ")
    my_dict[key] = None

# Prompt the user to input the values for each key
for key in my_dict:
    value = input("Enter the value for " + key + ": ")
    my_dict[key] = value

# Display the resulting dictionary
print("Resulting dictionary:")
print(my_dict)
```

Enter key 1: 1
Enter key 2: 2
Enter key 3: 3
Enter key 4: 4
Enter key 5: 5
Enter the value for 1 : Ahmed
Enter the value for 2: Ali
Enter the value for 3: Umer
Enter the value for 4: Tariq
Enter the value for 5: Bilal
Resulting dictionary:
{'1': 'Ahmed', '2': 'Ali', '3': 'Umer', '4': 'Tariq', '5': 'Bilal'}

8 LAB TASK 5

In this task, you will focus on file handling. Write code that first creates a text file “lab2.txt” with the message “My name is <your_name>”. Then, your code must open the file in *read* mode and display the contents of the text file. Next, the file must be opened in *append* mode and the message “My registration number is <reg_number>” must be added to the text file. Finally, the file is read again to display the modified contents.

8.1 CODE:

```
# Create a text file with a message
filename = "lab2.txt"
name = "Muhammad Ahmed Mohsin"
reg_number = "333060"
with open(filename, "w") as file:
    file.write("My name is " + name)

# Read the contents of the file
with open(filename, "r") as file:
    content = file.read()
```



```
print("File contents (before appending):")
print(content)

# Append additional text to the file
with open(filename, "a") as file:
    file.write("\nMy registration number is " + reg_number)

# Read the modified contents of the file
with open(filename, "r") as file:
    modified_content = file.read()
    print("\nFile contents (after appending):")
    print(modified_content)
```

8.2 OUTPUT:

```
# Create a text file with a message
filename = "lab2.txt"
name = "Muhammad Ahmed Mohsin"
reg_number = "333060"
with open(filename, "w") as file:
    file.write("My name is " + name)

# Read the contents of the file
with open(filename, "r") as file:
    content = file.read()
    print("File contents (before appending):")
    print(content)

# Append additional text to the file
with open(filename, "a") as file:
    file.write("\nMy registration number is " + reg_number)

# Read the modified contents of the file
with open(filename, "r") as file:
    modified_content = file.read()
    print("\nFile contents (after appending):")
    print(modified_content)
```

File contents (before appending):
My name is Muhammad Ahmed Mohsin

File contents (after appending):
My name is Muhammad Ahmed Mohsin
My registration number is 333060

9 CONCLUSION

In this lab we learnt about the basics of python and how to use lists, dictionaries, and file handling. In this lab we learnt how to implement python code using Google Collab. We learnt the basic syntax of python and then we understood how to implement small codes in programming language. Later



National University of Sciences and Technology (NUST) **School of Electrical Engineering and Computer Science**

we will be using this to implement in wire shark and other platforms to implement different communication things.