**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**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO5/**  **CLO3** | | **PLO5/**  **CLO3** | **PLO5/**  **CLO3** | **PLO5/**  **CLO3** |
| **Name** | **Reg. No** | **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** |  |  |  |  |  |
|  |  |  |  |  |  |  |

Contents

[2 Introduction 44](#_Toc127364870)

[2.1 Objectives 44](#_Toc127364871)

[3 Lab Conduct 44](#_Toc127364872)

[4 Theory 45](#_Toc127364873)

[5 Lab Task 1 45](#_Toc127364874)

[5.1 Code 46](#_Toc127364875)

[5.2 Output 46](#_Toc127364876)

[6 Lab Task 2 47](#_Toc127364877)

[6.1 Code 47](#_Toc127364878)

[6.2 output 48](#_Toc127364879)

[7 Lab Task 3 48](#_Toc127364880)

[7.1 Code 48](#_Toc127364881)

[8 Lab Task 4 50](#_Toc127364882)

[8.1 code 50](#_Toc127364883)

[8.2 output 51](#_Toc127364884)

[9 Lab Task 5 51](#_Toc127364885)

[9.1 Code: 51](#_Toc127364886)

[9.2 Output: 51](#_Toc127364887)

[10 Conclusion 51](#_Toc127364888)

# 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.

## 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

# 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

# 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

# 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:**

## 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:

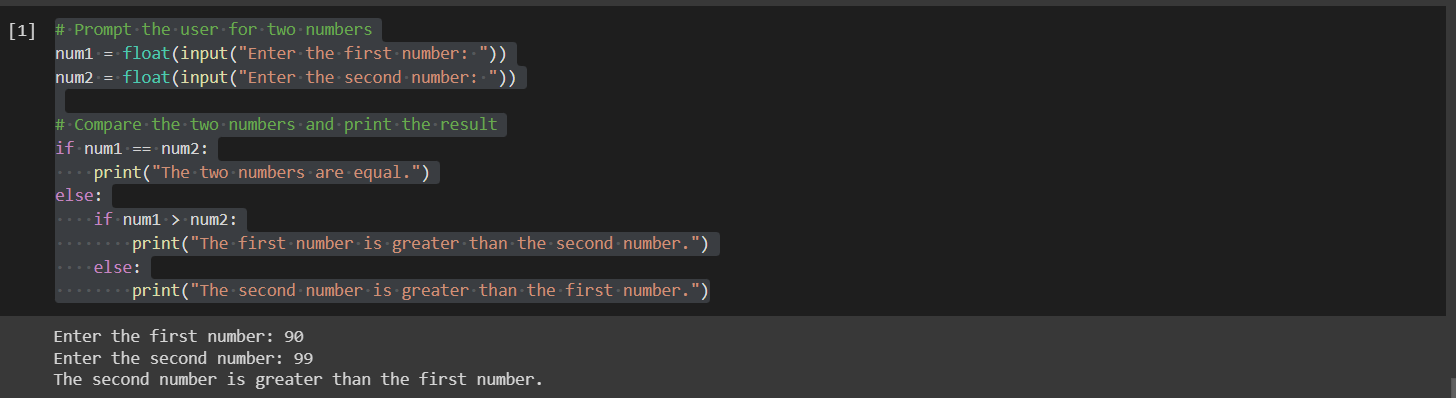
    if num1 > num2:

        print("The first number is greater than the second number.")

    else:

        print("The second number is greater than the first number.")

## Output

**

# 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**

## 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])

## Text Description automatically generatedoutput

# 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:**

## 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]

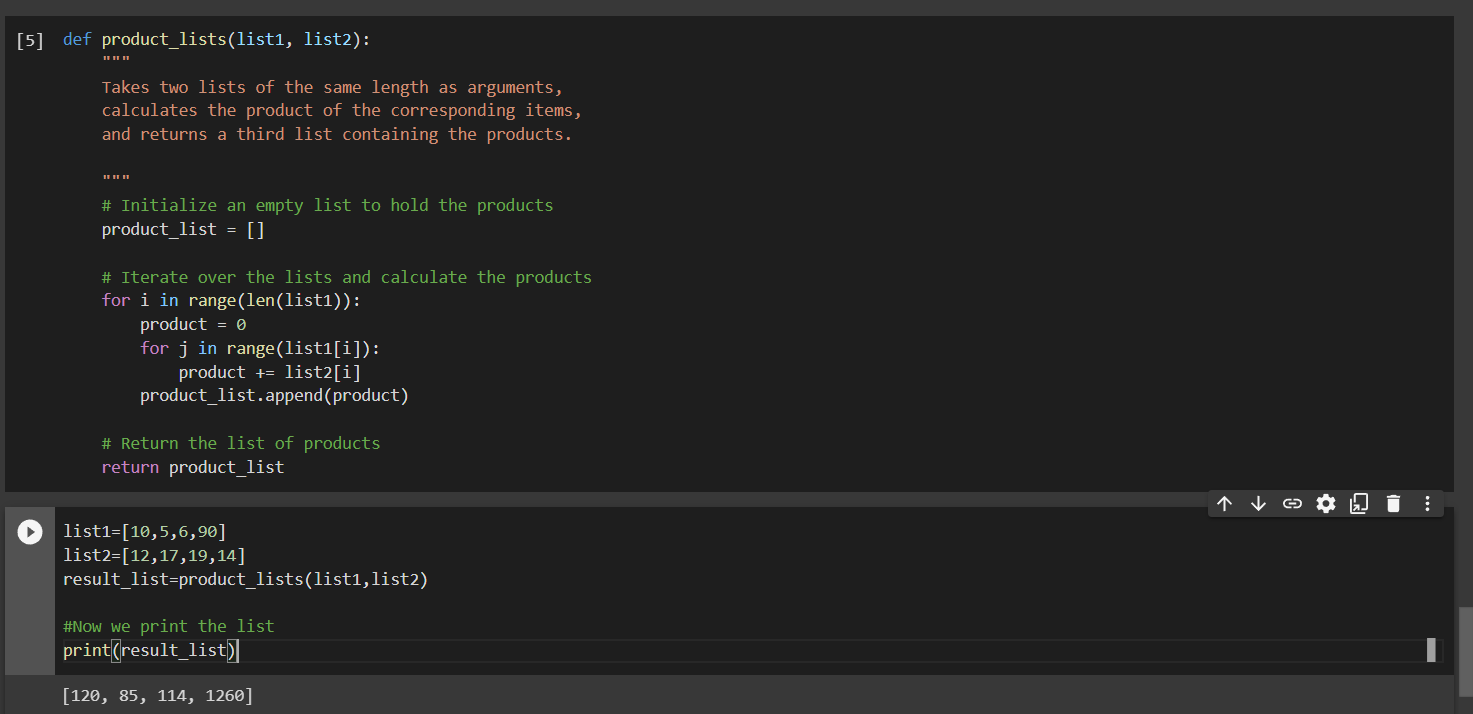
list2=[12,17,19,14]

result\_list=product\_lists(list1,list2)

#Now we print the list

print(result\_list)

## output

**

# 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.**

## 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)

## Graphical user interface, text, application Description automatically generatedoutput

# 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.**

## 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

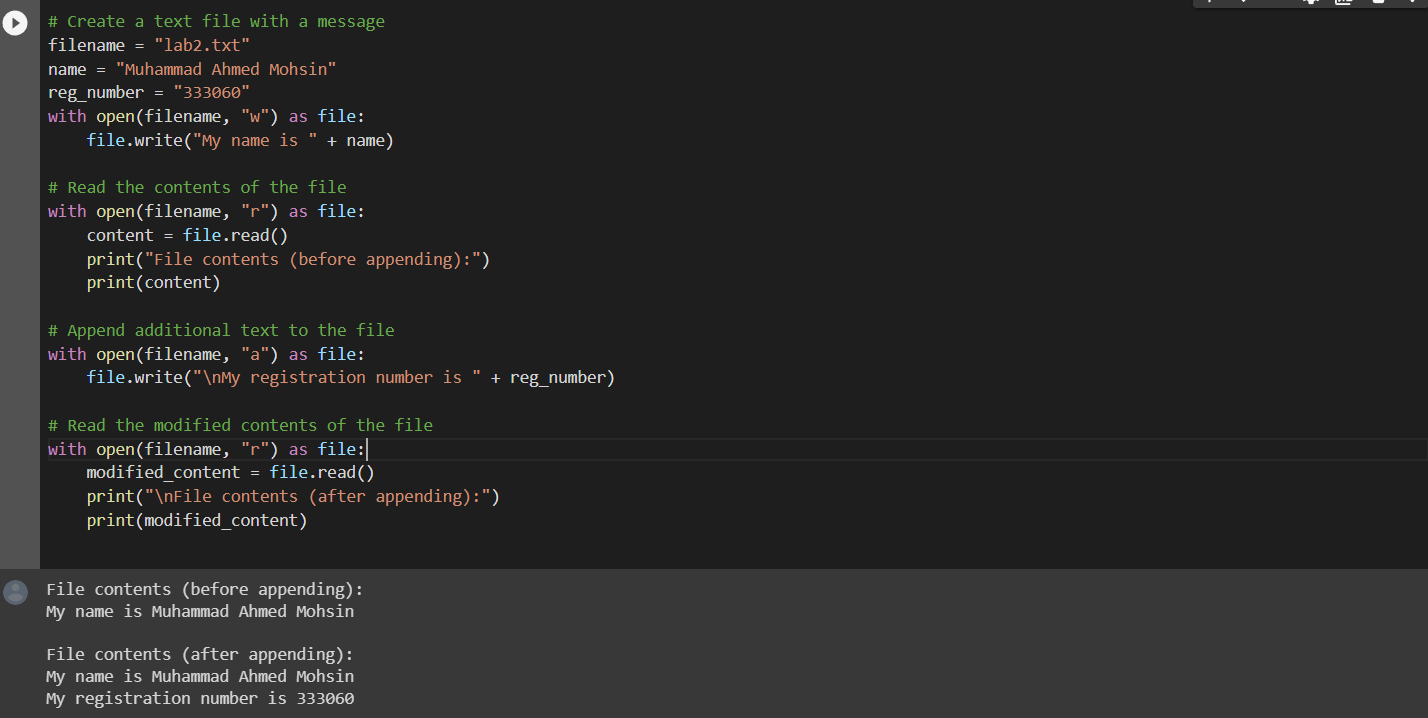
with open(filename, "r") as file:

    modified\_content = file.read()

    print("\nFile contents (after appending):")

    print(modified\_content)

## Output:



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