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

Department of Electrical Engineering

Faculty Member: Ma'am Neelma Naz **Date:** 9/18/2025

Semester: 7th **Group:** 02

CS471 Machine Learing

Lab 1: Introduction to Python

		PLO4	PLO5	PLO5	PLO8	PLO9
		CLO4	CLO5	CLO5	CLO6	CLO7
Student Name	**Reg. No**	**Viva / Quiz / Demo** 5 Marks	**Analysis of Data in Report** 5 Marks	**Modern Tool Usage** 5 Marks	**Ethics** 5 Marks	**Individual and Teamwork** 5 Marks
Hanzla Sajjad	403214					
Irfa Farooq	412564					

Introduction

This laboratory exercise will introduce the fundamental aspects of the Python programming language which is a very popular programming language and is used extensively in the area of Machine Learning.

Objectives

The following are the main objectives of this lab:

- Write and execute python code in Google Colaboratory (Colab)
- Create and use variables of different data types in python
- Use arithmetic and logical operations in python
- Implement conditional statements in python
- Implement WHILE and FOR loops in python
- Define and call functions 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/screenshot/plot 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 open source, interpreted language which is widely used for machine learning tasks in research, academia and industry. It has an easy-to-learn syntax and is ideal for writing programs in a short duration. The python interpreter can be downloaded from the website and installed on the system. By default, the IDLE program is installed. For machine learning, it is recommended to switch to a more powerful IDE such as PyCharm, Spyder and Jupyter etc. For this lab, we will use Google Colab for writing python code. Google Colab is a cloud-based platform that allows you to write python code in your web browser and provides free access to computing resources such as GPUs.

A brief summary of the relevant keywords and functions in python is provided below:

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
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 _____ [2]

Write a program which evaluates the following three expressions for when x = 4, 5, 6, 7 and 8.

$$4x^3 + 5x^2 + 3x + 2$$

$$\frac{3x^2+7x}{2} - \frac{2x}{5}$$

(a) Fill the following table with the answers:

	x = 4	x = 5	x = 6	x = 7	x = 8
Expression 1	350	642	1064	1640	**2394**
Expression 2	36.4	53.0	72.6	95.2	**120.8**

(b) Provide the code for both expressions in the indicated regions.

Code

```
x = 8
y = 4*x**(3) + 5*x**(2) + 3*x + 2
print(y)
```

```
2394
```

Code

```
x = 4
y = ((3*x**2 + 7*x)/2) - ((2*x)/5)
print(y)
```

```
36.4
```

Lab Task 2 _____ [1]

Write a program that reads in two integer inputs, then determines and prints if the first is a multiple of the second. To input a variable, use the following syntax:

```
variable = input("prompt_message")
```

Remember that the above function returns a string which is stored in the variable. You need to explicitly convert the string variable to an integer type using the int() casting. Provide the code and screenshot of the result.

Code

```
variable1 = input("Input variable 1: ")
variable2 = input("Input variable 2: ")

if (int(variable1) % int(variable2) == 0):
  print(variable1, "is a multiple of", variable2)
else:
  print(variable1, "is not a multiple of", variable2)
```

```
Input variable 1: 12
Input variable 2: 2
12 is a multiple of 2
```

Lab Task 3 _____ [1]

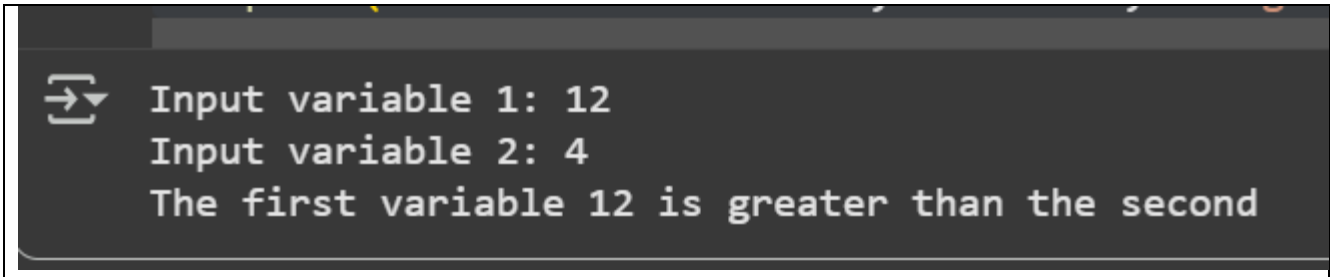
Write a program that prompts the user for two numbers as input. Then, the program must compare the two numbers and print if they are equal or not. If the numbers are not equal, it must also print which number is greater (or lesser) than the other. The syntax for conditional statements is given as follows:

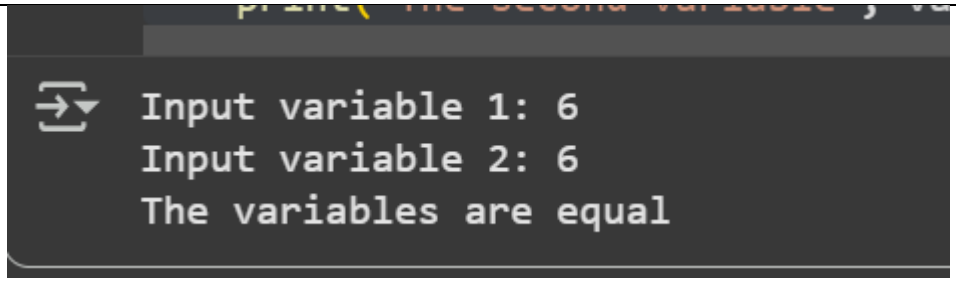
```
if condition:
    statement_1
else:
    statement_2
```

Code

```
variable1 = input("Input variable 1: ")
variable2 = input("Input variable 2: ")

if (int(variable1) == int(variable2)):
  print("The variables are equal")
else:
  if (int(variable1) > int(variable2)):
    print("The first variable", variable1, "is greater than the second")
  else:
    print("The second variable", variable2, "is greater than the first")
```


Input variable 1: 12
Input variable 2: 4
The first variable 12 is greater than the second


Input variable 1: 6
Input variable 2: 6
The variables are equal

Lab Task 4 _____ [1]

Write a program that takes two numbers as inputs. Then, the program must compare the two numbers and print appropriately from among the following lines:

- Both numbers are positive
- Both numbers are negative
- Both numbers are zero
- At least one number is zero
- One number is positive and the other number is negative

Code

```
variable1 = input("Input variable 1: ")
variable2 = input("Input variable 2: ")

if (int(variable1) > 0 and int(variable2) > 0):
  print("Both numbers are positive")
elif (int(variable1) < 0 and int(variable2) < 0):
  print("Both numbers are negative")
elif (int(variable1) == 0 and int(variable2) == 0):
  print("Both numbers are zero")
elif (int(variable1) == 0 or int(variable2) == 0):
  print("At least one number is zero")
elif ((int(variable1) > 0 and int(variable2) < 0) or (int(variable2) > 0 and
int(variable1) < 0)):
  print("One number is positive and the other is negative")
```

```
Input variable 1: 2
Input variable 2: 1
Both numbers are positive
```

```
Input variable 1: -1
Input variable 2: -12
Both numbers are negative
```

```
Input variable 1: 0
Input variable 2: 0
Both numbers are zero
```

```
Input variable 1: 0
Input variable 2: -6
At least one number is zero
```

```
Input variable 1: 6
Input variable 2: -9
One number is positive and the other is negative
```

Lab Task 5 _____ [1]

Write a program that calculates the factorial of a number. To calculate the factorial, you will need to make use of a *while* loop. The syntax of the while loop is given as follows:

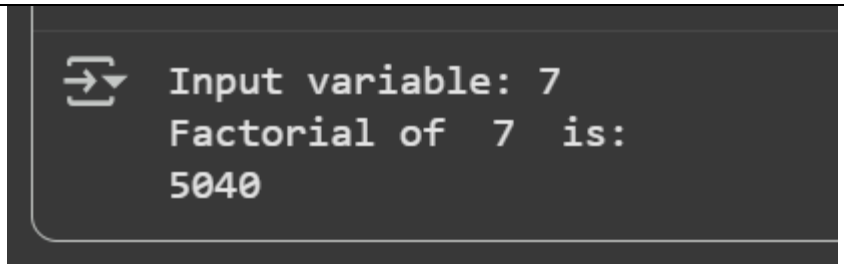
```
while condition:
    statement_1
    statement_2
```

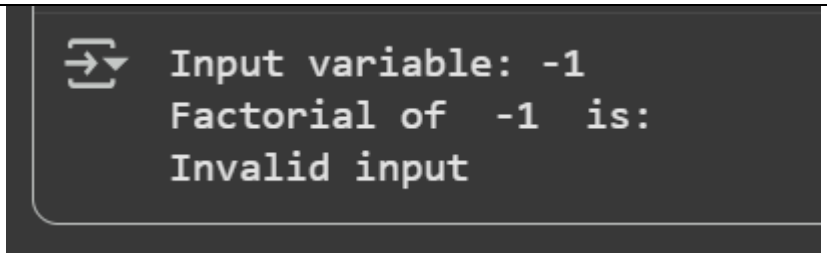
Code

```
variable = int(input("Input variable: "))
print("Factorial of ", variable, " is: ")
y = 1

if(variable == 0):
  print("1")
elif (variable > 0):
  while(variable != 0):
    y = variable*y
    variable = variable - 1
  print(y)
elif (variable < 0):
  print("Invalid input")
```

```
Input variable: 0
Factorial of  0  is:
1
```


Input variable: 7
Factorial of 7 is:
5040


Input variable: -1
Factorial of -1 is:
Invalid input

Lab Task 6 _____ [1]

Write a function that takes 2 integer arguments and returns their product but you must **NOT** use the product operator (*). You will need to provide the function definition and the function call. (Hint: You need to make use of loops in your function.) The function definition syntax is given below:

```
def function_name:
    statement_1
    statement_2
    ...
    return output
```

Code

```
def product_self(a, b):
  c = 0

  while(a != 0):
    c = b + c
    a = a - 1

  return c

print("The product is: ", product_self(5, 4))
```

```
The product is:  20
```

Lab Task 7 _____ [1]

Write a program that prompts the user for 3 strings variables. The user will input the strings separately at the prompt, e.g. “TRI”, “GONO” and “METRY”. The strings will then be passed to a function as arguments. The function must use a *for* loop to iterate through the characters and print each character on a new line. The function must also print the total number of characters in the final string. For this, you can use the len() function. Note that the “TRIGONOMETRY” string is just an example and you need to use your own names as the string for the submission. You also need to take screenshot of this task showing the entire output. The for loop syntax is given as follows:

for index in *iterable*:
statement_1
statement_2

Code

```
def product_self(a, b):
  c = 0
name1 = input("Enter first half name: ")
name2 = input("Enter second half name: ")
name3 = input("Enter third half name: ")

def combination(a, b, c):

  for i in a:
    print(i)

  for i in b:
    print(i)

  for i in c:
    print(i)
```

```
  total_length = len(a) + len(b) + len(c)
  print("Total characters in name: ", total_length)

combination(name1, name2, name3)
```

```
Enter first half name: IRFA
Enter second half name: FAR
Enter third half name: OOQ
I
R
F
A
F
A
R
O
O
Q
Total characters in name:  10
```

Lab Task 8 _____ [1]

Write a program that generates the following number sequences and print the output. You can use the range() function for this task. Use a loop to invoke the range function iteratively.

1, 2, 3… 20
2, 4, 6… 40
3, 6, 9… 60
4, 8, 12 … 80
…
10, 20, 30… 200

Code

```
for i in range(1, 11):
  for j in range(1, 21):
    if j != 20:
      print(i * j, ", ", end= " ")
    elif j == 20:
      print(i * j, end= " ")
  print()
```

```
1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15 , 16 , 17 , 18 , 19 , 20
2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 , 20 , 22 , 24 , 26 , 28 , 30 , 32 , 34 , 36 , 38 , 40
3 , 6 , 9 , 12 , 15 , 18 , 21 , 24 , 27 , 30 , 33 , 36 , 39 , 42 , 45 , 48 , 51 , 54 , 57 , 60
4 , 8 , 12 , 16 , 20 , 24 , 28 , 32 , 36 , 40 , 44 , 48 , 52 , 56 , 60 , 64 , 68 , 72 , 76 , 80
5 , 10 , 15 , 20 , 25 , 30 , 35 , 40 , 45 , 50 , 55 , 60 , 65 , 70 , 75 , 80 , 85 , 90 , 95 , 100
6 , 12 , 18 , 24 , 30 , 36 , 42 , 48 , 54 , 60 , 66 , 72 , 78 , 84 , 90 , 96 , 102 , 108 , 114 , 120
7 , 14 , 21 , 28 , 35 , 42 , 49 , 56 , 63 , 70 , 77 , 84 , 91 , 98 , 105 , 112 , 119 , 126 , 133 , 140
8 , 16 , 24 , 32 , 40 , 48 , 56 , 64 , 72 , 80 , 88 , 96 , 104 , 112 , 120 , 128 , 136 , 144 , 152 , 160
9 , 18 , 27 , 36 , 45 , 54 , 63 , 72 , 81 , 90 , 99 , 108 , 117 , 126 , 135 , 144 , 153 , 162 , 171 , 180
10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 , 100 , 110 , 120 , 130 , 140 , 150 , 160 , 170 , 180 , 190 , 200
```

Conclusion:

This lab successfully introduced the fundamental aspects of the Python programming language, achieving all of the stated objectives. We gained hands-on experience using Google Colaboratory (Colab) to write and execute Python code in a cloud-based environment. This exercise reinforced our understanding of core programming concepts, including the creation and manipulation of variables with different data types (e.g., integers, floats, strings).

Furthermore, we applied various arithmetic and logical operations to perform calculations and make comparisons, which are essential for any programming task. The implementation of conditional statements (if, elif, else) allowed us to write programs that can make decisions based on specific conditions. We also mastered the use of loops (for and while) to automate repetitive tasks and iterate over sequences, demonstrating their efficiency in handling data. Finally, we learned how to define and call functions (def), a crucial practice for writing modular, reusable, and organized code.

Overall, this lab provided a solid foundation in Python's core syntax and functionalities, skills that are directly applicable to more advanced topics such as machine learning and data science.