**Machine Learning**

Lab 1: Introduction to Python

Yasir khan

**Table of Contents**

[2 Introduction to Python 3](#_Toc145854617)

[2.1 Introduction 3](#_Toc145854618)

[2.2 Objectives 3](#_Toc145854619)

[2.3 Software 3](#_Toc145854620)

[3 Lab Tasks 4](#_Toc145854621)

[3.1 Task 1 4](#_Toc145854622)

[3.2 Task 2 4](#_Toc145854623)

[3.3 Task 3 5](#_Toc145854624)

[3.4 Task 4 6](#_Toc145854625)

[3.5 Task 5 7](#_Toc145854626)

[3.6 Task 6 7](#_Toc145854627)

[3.7 Task 7 8](#_Toc145854628)

[3.8 Task 8 9](#_Toc145854629)

[4 Conclusion 10](#_Toc145854630)

# Introduction to Python

## Introduction

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

## Objectives

The following are the main objectives of this lab:

* 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

## Software

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.

A 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

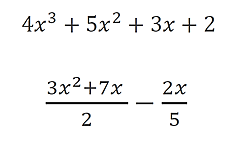
**continue** jump to the next iteration of the loop

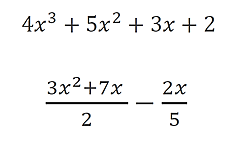
**def** used to define a function

# Lab Tasks

## Task 1

Write a program which evaluates the following three expressions for when x = 1, 2, 3, 4 and 5.





1. Fill the following table with the answers:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **x = 1** | **x = 2** | **x = 3** | **x = 4** | **x = 5** |
| **Expression 1** | 14 | 60 | 164 | 350 | 642 |
| **Expression 2** | 4.6 | 12.2 | 22.8 | 36.4 | 53.0 |

1. Provide the code for both expressions in the indicated regions:

### Expression 1 Code Starts Here ###

*def f(x):*

*result = (4\*(x\*\*3)) + (5\*(x\*\*2)) + (3\*x) + 2*

*return result*

*for i in range(1,6):*

*print(f"x={i}: {f(i)}")*

### Expression 1 Code Ends Here ###

### Expression 2 Code Starts Here ###

def g(x):

result = ((3 \* (x\*\*2) + (7\*x)) / 2) - ((2\*x) / 5)

return result

for i in range(1,6):

print(f"x={i}: {g(i)}")

### Expression 2 Code Ends Here ###

## Task 2

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

### Task 2 Code Starts Here ###

def check\_multiple(x, y):

if x % y == 0:

print(f"{x} is the multiple of {y}")

else:

print(f"{x} isn't the multiple of {y}")

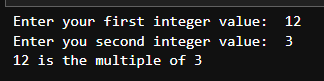
var1 = int(input("Enter your first integer value: "))

var2 = int(input("Enter you second integer value: "))

check\_multiple(var1, var2)

### Task 2 Code Ends Here ###

### Task 2 Screenshot Starts Here ###



### Task 2 Screenshot Ends Here ###

## Task 3

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 as follows:

**if** condition:

statement\_1

**else**:

statement\_2

### Task 3 Code Starts Here ###

def compare\_numbers():

var1 = int(input("Enter the first number: "))

var2 = int(input("Enter the second number: "))

if var1 == var2:

print(f"{var1} is equal to {var2}")

elif var1 > var2:

print("Both numbers aren't equal")

print("{} is greater than {}".format(var1, var2))

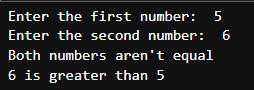
else:

print("Both numbers aren't equal")

print("{} is greater than {}".format(var2, var1))

### Task 3 Code Ends Here ###

### Task 3 Screenshot Starts Here ###



### Task 3 Screenshot Ends Here ###

## Task 4

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

### Task 4 Code Starts Here ###

def comparison():

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

if num1 > 0 and num2 > 0:

print("Both numbers are positive")

elif num1 < 0 and num2 <0:

print("Both numbers are negative")

elif num1 == 0 and num2 ==0:

print("Both numbers are zero")

elif num1 == 0 or num2 == 0:

print("At least one number is zero")

elif num1 > 0 and num2 < 0:

print("One number is positive, and the other is negative")

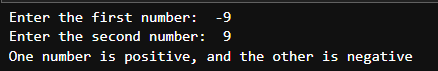
elif num1 < 0 and num2 > 0:

print("One number is positive, and the other is negative")

comparison()

### Task 4 Code Ends Here ###

### Task 4 Screenshot Starts Here ###

****

### Task 4 Screenshot Ends Here ###

## Task 5

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

### Task 5 Code Starts Here ###

def fact(number):

result = 1

while(number>0):

result = result \* number

number-=1

return result

print("Factorial of 5 is: ", fact(6))

### Task 5 Code Ends Here ###

### Task 5 Screenshot Starts Here ###

C:\Users\yasir\OneDrive\Pictures\Screenshots\Screenshot 2025-07-06 143213.png

### Task 5 Screenshot Ends Here ###

## Task 6

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

### Task 6 Code Starts Here ###

def product\_with\_loop(num1, num2):

prod = 0

for i in range(num2):

prod += num1

return prod

val1, val2 = 5,4

print(f"Product of {val1} and {val2} is: {product\_with\_loop(val1,val2)}")

### Task 6 Code Ends Here ###

### Task 6 Screenshot Starts Here ###

**C:\Users\yasir\OneDrive\Pictures\Screenshots\Screenshot 2025-07-06 144729.png**

### Task 6 Screenshot Ends Here ###

## Task 7

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 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

### Task 7 Code Starts Here ###

def count\_characters(str1, str2, str3):

concatenated = str1+str2+str3

for ch in concatenated:

print(ch)

print("The total number of characters are: ", len(concatenated))

str1 = input("Enter first string: ")

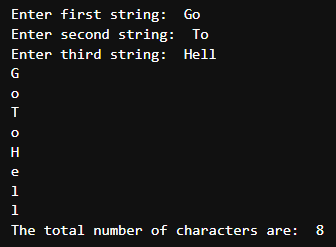
str2 = input("Enter second string: ")

str3 = input("Enter third string: ")

count\_characters(str1, str2, str3)

### Task 7 Code Ends Here ###

### Task 7 Screenshot Starts Here ###



### Task 7 Screenshot Ends Here ###

## Task 8

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

### Task 8 Code Starts Here ###

counter=21

for i in range(1,11):

for j in range(i, counter, i):

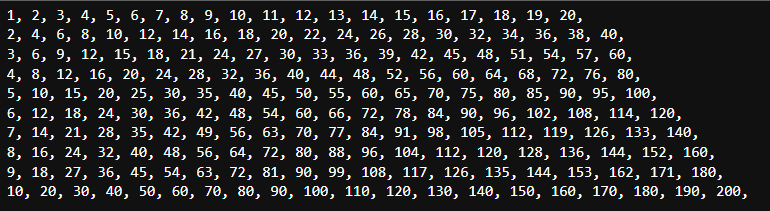
print(j, end=', ')

counter+=20

print()

### Task 8 Code Ends Here ###

### Task 8 Screenshot Starts Here ###



### Task 8 Screenshot Ends Here ###

# Conclusion

In conclusion, this laboratory exercise has provided a comprehensive introduction to the fundamental aspects of the Python programming language. Python's widespread popularity and extensive use in fields such as Machine Learning make it a valuable skill for any aspiring programmer or data scientist. Throughout this lab, we have achieved the specified objectives, which include creating and utilizing variables of various data types, mastering arithmetic, and logical operations, implementing conditional statements, and successfully working with both WHILE and FOR loops. Furthermore, we have gained proficiency in defining and calling functions in Python, a crucial skill for building modular and efficient code.