**Department of Electrical Engineering and   
Computer Science**

**Faculty Member:** Dr. Hafsa Iqbal **Date:** 30/01/2024

**Semester:** 8th **Group:** GP-1

**Lab Engineer:** Munadi Ahmad Sial

**EE-381 Robotics**

Lab 1: Python Programming in Linux (Ubuntu)

**Group Members**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **PLO5 – CLO4** | | **PLO9 –**  **CLO4** |  |
| **Name** | **Reg. No** | **Analysis of Data in Lab Report** | **Modern Tool Usage** | **Individual and Team Work** | **Total Marks** |
|  |  | **10 Marks** | **5 Marks** | **5 Marks** | **20 Marks** |
| Muhammad Abdullah Sohail | 343642 |  |  |  |  |
| Muhammad Ahmed Mohsin | 333060 |  |  |  |  |
| Muhammad Umer | 345834 |  |  |  |  |
| Hassan Rizwan | 335753 |  |  |  |  |

**Table of Contents**

[2 Python Programming in Linux 3](#_Toc157868402)

[2.1 Introduction 3](#_Toc157868403)

[2.2 Objectives 3](#_Toc157868404)

[2.3 Software 3](#_Toc157868405)

[3 Lab Tasks 4](#_Toc157868406)

[3.1 Task 1 – Operator Precedence 4](#_Toc157868407)

[3.2 Task 2 – Input Prompt and Type Casting 4](#_Toc157868408)

[3.3 Task 3 – Selection Statements 5](#_Toc157868409)

[3.4 Task 4 – Relational Operators 6](#_Toc157868410)

[3.5 Task 5 – WHILE Loop 7](#_Toc157868411)

[3.6 Task 6 – Functions 7](#_Toc157868412)

[3.7 Task 7 – FOR Loops 8](#_Toc157868413)

[4 Conclusion 9](#_Toc157868414)

# Python Programming in Linux

## Introduction

This laboratory exercise is meant to introduce the fundamental aspects of the python programming language which will be very important when learning the Robot Operating System (ROS) for programming robots.

## 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
* Run the python scripts using the terminal in Linux

## Software

Python is an interpreted language which is very popular in the field of robotics. It has an easy-to-learn syntax and is ideal for developing robot prototypes in a short time. To write python scripts (.py files), text editors with syntx highlighting like SublimeText application can be used. Once the code is written, the script is saved and can be executed by using the Linux terminal where the text output is displayed.

The terminal commands are given as:

**cd <directory>**  change directory

**cd..**  go back to previous directory

**pwd**  print the current directory

**ls** list the contents of the current directory

**python <script.py>** execute python script

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

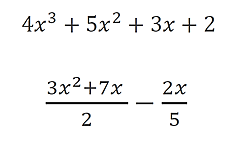
**elif** equivalent to else if

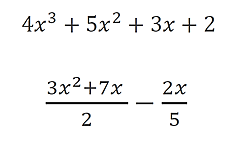
**while** loops code as long as a condition is true

# Lab Tasks

## Task 1 – Operator Precedence

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*):

    return 4 \* x\*\*3 + 5 \* x\*\*2 + 3 \* x + 2

for x in range(1, 5 + 1):

    print("f({}) = {}".format(x, f(x)))

### Expression 1 Code Ends Here ###

### Expression 2 Code Starts Here ###

*def* g(*x*):

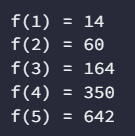
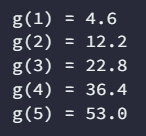
    return ((3 \* x\*\*2 + 7 \* x) / 2) - (2 \* x / 5)

for x in range(1, 5 + 1):

    print("g({}) = {}".format(x, g(x)))

### Expression 2 Code Ends Here ###

### TASK 1 SCREENSHOT STARTS HERE ###

### TASK 1 SCREENSHOT ENDS HERE ###

## Task 2 – Input Prompt and Type Casting

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 ###

*# Input two integers*

a = *int*(input("Enter first integer: "))

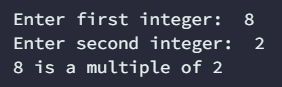
b = *int*(input("Enter second integer: "))

if a % b == 0:

    print("{} is a multiple of {}".format(a, b))

### Task 2 Code Ends Here ###

### Task 2 Screenshot Starts Here ###



### Task 2 Screenshot Ends Here ###

## Task 3 – Selection Statements

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 ###

*# Input two numbers*

a = *int*(input("Enter first number: "))

b = *int*(input("Enter second number: "))

*# Logic*

if a == b:

    print("Both numbers are equal")

else:

    print("The numbers are not equal")

    if a > b:

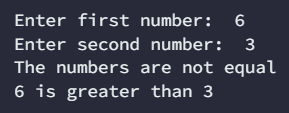
        print("{} is greater than {}".format(a, b))

    else:

        print("{} is lesser than {}".format(a, b))

### Task 3 Code Ends Here ###

### Task 3 Screenshot Starts Here ###



### Task 3 Screenshot Ends Here ###

## Task 4 – Relational Operators

Write a program that prompts the user for two numbers as input. Then, the program must compare the two numbers and print one of 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 ###

*# Input two numbers*

a = *int*(input("Enter first number: "))

b = *int*(input("Enter second number: "))

if a > 0 and b > 0:

    print("Both numbers are positive")

elif a < 0 and b < 0:

    print("Both numbers are negative")

elif a == 0 and b == 0:

    print("Both numbers are zero")

elif a == 0 or b == 0:

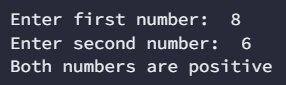
    print("At least one number is zero")

else:

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

### Task 4 Code Ends Here ###

### Task 4 Screenshot Starts Here ###

****

### Task 4 Screenshot Ends Here ###

## Task 5 – WHILE Loop

Write a program that calculates the factorial of a number. The program must prompt the user for the input number. To calculate the factorial, you need to make use of a WHILE loop. The syntax of the WHILE loop is given as:

**while** condition:

statement\_1

statement\_2

### Task 5 Code Starts Here ###

*# Input a number*

a = *int*(input("Enter a number: "))

b = a

*# Factorial through while loop*

factorial = 1

while a > 1:

    factorial \*= a

    a -= 1

print("Factorial of {} is {}".format(b, factorial))

### Task 5 Code Ends Here ###

### Task 5 Screenshot Starts Here ###



### Task 5 Screenshot Ends Here ###

## Task 6 – Functions

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(*a*, *b*):

    product = 0

    while b > 0:

        product += a

        b -= 1

    return product

a = 2

b = 4

print("Product of {} and {} is {}".format(a, b, product(a, b)))

### Task 6 Code Ends Here ###

### Task 6 Screenshot Starts Here ###

****

### Task 6 Screenshot Ends Here ###

## Task 7 – FOR Loops

Write a program that prompts the user for 3 strings variables. The user will input the 3 strings separately at the prompt. The 3 strings must be from the names of your group members. The strings will then be passed to a function as arguments. The function must use a FOR loop to iterate through all of the characters and print them. Note that each individual character is to be printed on a new line. Provide the code and a screenshot showcasing your work. You need to show both the user inputs and resulting output in the same terminal. The FOR-loop syntax is given as:

**for** index in iterable:

statement\_1

statement\_2

### Task 7 Code Starts Here ###

s1 = input("Enter first string: ")

s2 = input("Enter second string: ")

s3 = input("Enter third string: ")

*def* print\_string(*s1*, *s2*, *s3*):

    s = s1 + s2 + s3

    for c in s:

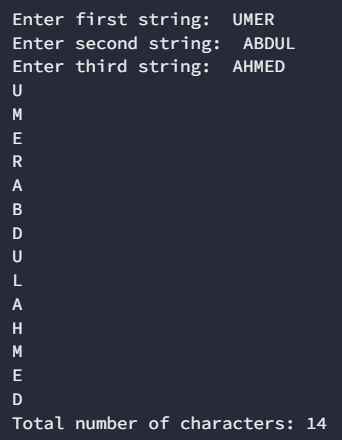
        print(c)

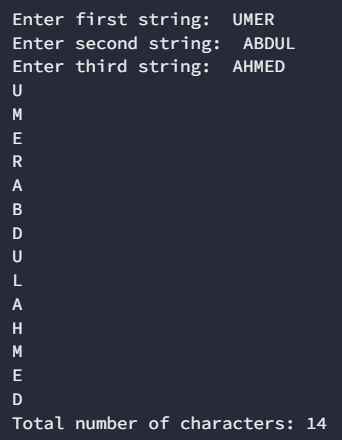
    print("Total number of characters: {}".format(len(s)))

print\_string(s1, s2, s3)

### Task 7 Code Ends Here ###

### Task 7 Screenshot Starts Here ###





### Task 7 Screenshot Ends Here ###

# Conclusion

In this lab, we delved into the foundational elements of the Python programming language, recognizing their pivotal role in mastering the Robot Operating System (ROS) for programming robots. Our primary objectives encompassed the creation of variables with diverse data types, the application of arithmetic and logical operations, the implementation of conditional statements and loops, the crafting of functions, and the execution of Python scripts through the Linux terminal.