**Python Projects Report**

**Submitted by: Homagni Dauka**

**Roll No: CS23BCAGN034**

**Submitted To: Mr. Nawab Wasim Rahman**

**The Assam Kaziranga University**

**Date: 19th May 2025**

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my respected faculty member, Mr. Nawab Wasim Rahman, for hid invaluable guidance, continuous support, insightful feedback throughout the preparation of this Python Projects Report. His encouragement and expertise have played a significant role in deepening my understanding and successfully completing these assignments.

DECLARATION

I hereby declare that this Python Projects Report is the outcome of my own effort and work. It has not been submitted, either partially or entirely, to any other institution or for any other purpose. All references and sources used have been duly acknowledged.

TABLE OF CONTENTS

Acknowledgment

Declaration

Abstract

1. Introduction
2. Objective
3. Software & Tools Used
4. Program 1: Arithmetic and Quadratic Operations
5. Program 2: Linear Equation Solver
6. Program 3: Graph Plotting
7. Program 4: Python Function
8. Program 5: GUI Application using Tkinter
9. Conclusion

ABSTRACT

This report presents a series of Python-based programming tasks focused on mathematical computations and graphical interface design. The lab includes arithmetic and algebraic implementations, visual representation using graphs, recursive function logic, and simple GUI development with Tkinter. The goal is to enhance problem-solving skills and understand Python’s application in various domains such as mathematics, data visualization, and user interface development.

INTRODUCTION

Python is a high-level, interpreted programming language known for its simplicity and versatility. This lab assignment is designed to apply Python programming in solving mathematical problems and developing simple applications. The tasks include arithmetic operations, solving equations, visualizing functions using graphs, recursive functions, and creating GUI applications using Tkinter.

OBJECTIVE

* To implement basic arithmetic and algebraic expressions using python.
* To solve and visualize linear and quadratic equations.
* To understand and implement mathematical functions (like factorial).
* To create simple GUI applications using the Tkinter module.
* To develop and present these solutions in a readable and organized report format.

SOFTWARE TOOLS USED:

* Python 3.x – For writing and executing code
* VS Code/ IDLE – Code editor/IDE
* Matplotlib & NumPy – For plotting graphs and handling numerical data
* Tkinter – For creating GUI applications
* GitHub – For Uploading and sharing the project

Code with Explanations and Output:

# Q1: Arithmetic Operations and Quadratic Expression

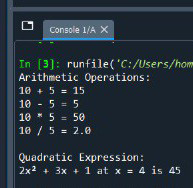
Code Explanation:

This function performs basic arithmetic operations like addition, subtraction, multiplication, and division between two numbers. The second function evaluates a quadratic expression of the form ax² + bx + c at a given x.

Code:

def arithmetic\_operations(a, b):  
 print("Arithmetic Operations:")  
 print(f"{a} + {b} = {a + b}")  
 print(f"{a} - {b} = {a - b}")  
 print(f"{a} \* {b} = {a \* b}")  
 if b != 0:  
 print(f"{a} / {b} = {a / b}")  
 else:  
 print("Division by zero is not allowed.")  
   
def evaluate\_quadratic(a, b, c, x):  
 result = a \* x\*\*2 + b \* x + c  
 print("\nQuadratic Expression:")  
 print(f"{a}x² + {b}x + {c} at x = {x} is {result}")  
arithmetic\_operations(10, 5)  
evaluate\_quadratic(2, 3, 1, 4)

Output:



# Q2: Linear Equation

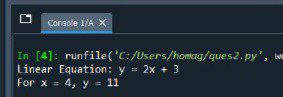
Code Explanation:

This function represents a linear equation of the form y = mx + c. It calculates the value of y for a given x, m (slope), and c (intercept).

Code:

def linear\_equation(m, x, c):  
 y = m \* x + c  
 print(f"Linear Equation: y = {m}x + {c}")  
 print(f"For x = {x}, y = {y}")  
linear\_equation(2, 4, 3)

Output:



# Q3: Graphical Representation of a Quadratic Function

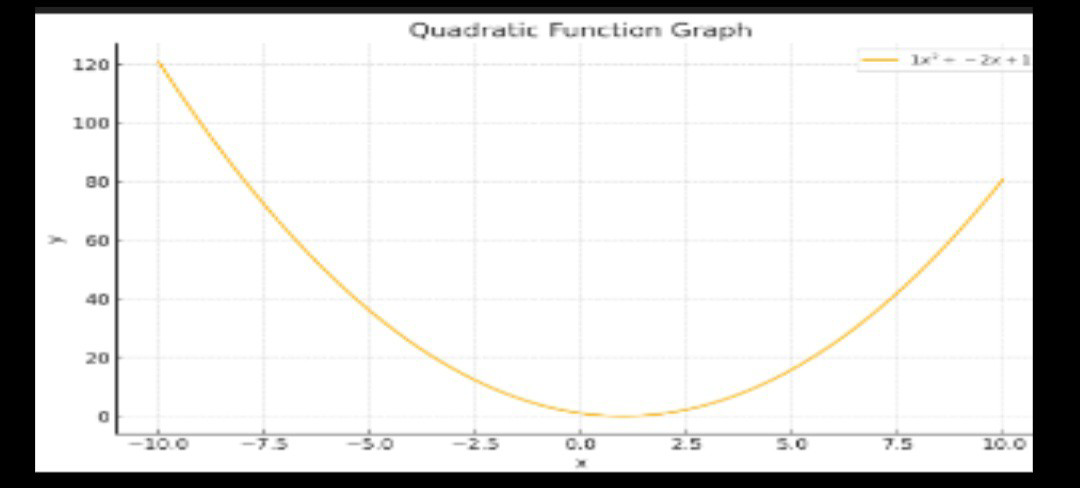
Code Explanation:

This function uses matplotlib to plot a quadratic function on a graph. The x values are generated using NumPy, and y values are calculated using the quadratic formula. It visualizes the curve of a parabola.

Code:

import matplotlib.pyplot as plt  
import numpy as np  
  
def plot\_quadratic(a, b, c):  
 x = np.linspace(-10, 10, 400)  
 y = a \* x\*\*2 + b \* x + c  
 plt.plot(x, y, label=f'{a}x² + {b}x + {c}', color='orange')  
 plt.title('Quadratic Function Graph')  
 plt.xlabel('x')  
 plt.ylabel('y')  
 plt.grid(True)  
 plt.legend()  
 plt.show()  
plot\_quadratic(1, -2, 1)

Output:



# Q4: Factorial Function

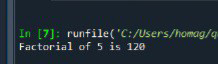
Code Explanation:

This function calculates the factorial of a number using recursion. The base case is 0! = 1 and 1! = 1.

Code:

def factorial(n):  
 if n == 0 or n == 1:  
 return 1  
 return n \* factorial(n - 1)  
num = 5  
print(f"Factorial of {num} is {factorial(num)}")

Output:



# Q5: GUI Application using Tkinter

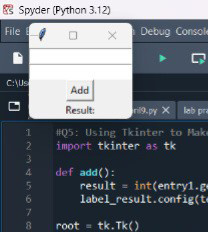
Code Explanation:

This GUI application allows users to input two numbers and displays their sum. Tkinter is used to create the interface which includes input fields, a button to trigger addition, and a label to show the result.

Code:

import tkinter as tk  
  
def add():  
 result = int(entry1.get()) + int(entry2.get())  
 label\_result.config(text=f"Result: {result}")  
  
root = tk.Tk()  
root.title("Simple Adder")  
  
entry1 = tk.Entry(root)  
entry1.pack()  
  
entry2 = tk.Entry(root)  
entry2.pack()  
  
button\_add = tk.Button(root, text="Add", command=add)  
button\_add.pack()  
  
label\_result = tk.Label(root, text="Result:")  
label\_result.pack()  
  
root.mainloop()

Output:



# Conclusion

This report demonstrates various basic to intermediate level Python programs, including arithmetic and algebraic computations, graph plotting, recursion, and GUI design. Each section is designed to show the application of Python in solving mathematical and real-world problems using both console and graphical user interfaces.

***THANK YOU***