Python Projects Report

Submitted by:

Sunena phukon.

Roll No: CS23BCAGN065

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 his valuable guidance, constant support, and helpful feedback during the preparation of this Python Projects Report. His encouragement and knowledge greatly helped me understand the subject better and complete these assignments successfully.

DECLARATION

I hereby declare that this Python Projects Report is the result of my own work, effort, and learning. I have prepared this report sincerely and have not copied it from any other student or source. This report has not been submitted, either partly or fully, to any other institution or for any other purpose.

I have tried my best to complete this report with honesty and dedication. Wherever I have used information, ideas, or help from books, websites, or any other sources, I have given proper credit and mentioned the references clearly.

This report reflects my understanding and hard work, and I take full responsibility for its content

Table of Contents

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 popular and easy-to-learn programming language. It is known for being simple to write and understand, which makes it great for beginners and professionals.

This lab assignment helps us use Python to solve math problems and build simple programs. The tasks include doing basic arithmetic, solving equations, drawing graphs to show functions, using recursive functions, and making basic GUI (Graphical User Interface) applications with 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

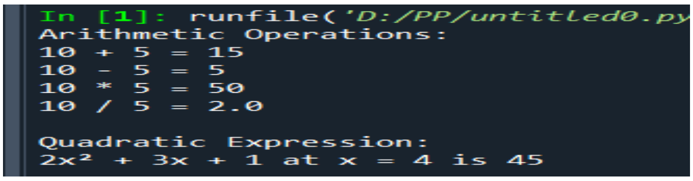
Q1: Using Python implementation of any arithmetic operation or quadratic expression?

```
# Arithmetic operations:
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.")
# Quadratic Expression:
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)
```

. **arithmetic_operations(a, b)** performs and prints:

- Addition, subtraction, multiplication, and division of two numbers.
- Handles division by zero safely.

. **evaluate_quadratic(a, b, c, x)** calculates the value of a **quadratic expression**



```
In [1]: runfile('D:/PP/untitled0.py
Arithmetic Operations:
10 + 5 = 15
10 - 5 = 5
10 * 5 = 50
10 / 5 = 2.0

Quadratic Expression:
2x² + 3x + 1 at x = 4 is 45
```

Q2: Implementation of Linear Equation?

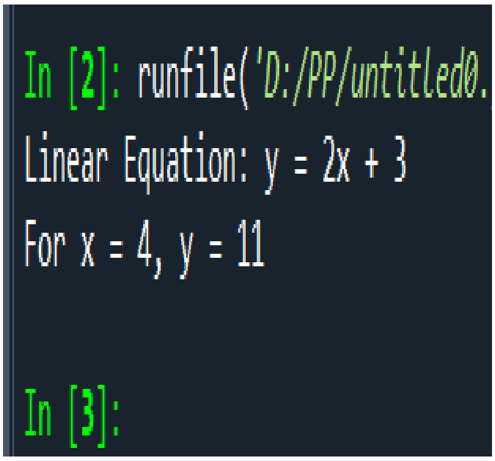
```
def linear_equation(m, x, c):
   y = m * x + c
   print(f"Linear Equation: y = {m}x + {c}")
   print(f"For x = {x}, y = {y}")
linear_equation(2, 4, 3)
```

. Defines a function to calculate the value of a linear equation:

y=mx+cy = mx + cy=mx+c

. Takes slope m, input x, and intercept c as inputs.

. Calculates y and prints the equation along with the result for the given x.



```
In [2]: runfile('D:/PP/untitled0.
Linear Equation: y = 2x + 3
For x = 4, y = 11


In [3]:
```

Q3: Using any Mathematical Functions or equation to give graphical representation like star, graph?

```
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}x^2 + {b}x + {c}')

  plt.title('Quadratic Function Graph')

  plt.xlabel('x')

  plt.ylabel('y')

  plt.grid(True)

  plt.legend()

  plt.show()

plot_quadratic(1, -2, 1)
```

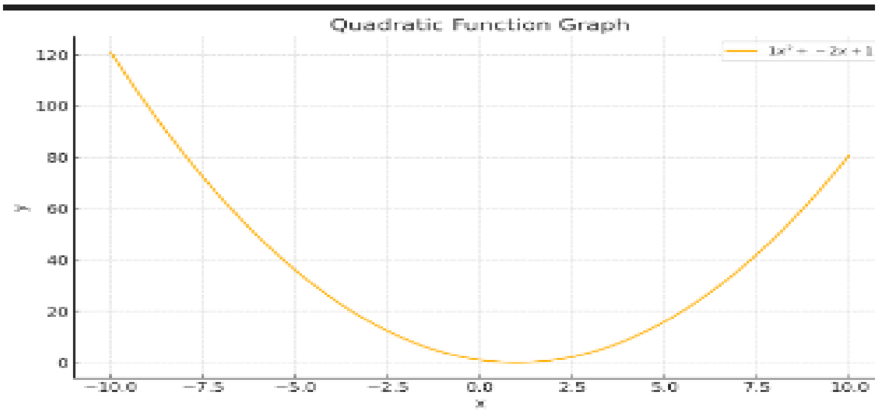
. Plots a quadratic equation: **$y = ax^2 + bx + c$**

. Uses **NumPy** to generate x-values from -10 to 10

. Calculates y-values using the quadratic formula

. Uses **Matplotlib** to display the graph with title, labels, grid, and legend

. plot_quadratic(1, -2, 1) plots: **$y = x^2 - 2x + 1$**



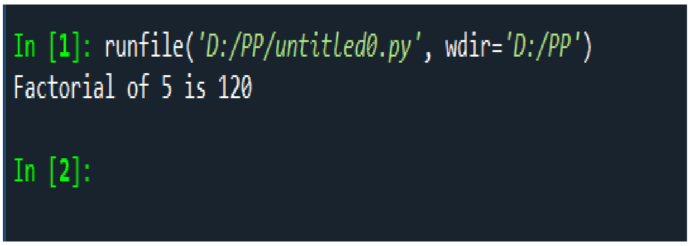
Q4. Write a Python Implementation of Function?

```
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)}")
```

. This function calculates the **factorial** of a number n using **recursion**.

. **Factorial** means multiplying a number by all positive integers less than it.

. n!=n×(n−1)×(n−2)×…×1n! = n × (n-1) × (n-2) × … × 1n!=n×(n−1)×(n−2)×…×1



```
In [1]: runfile('D:/PP/untitled0.py', wdir='D:/PP')
Factorial of 5 is 120

In [2]:
```

Q5: Using Tkinter Make any format Application according to your ideas(string,block,tetris)?

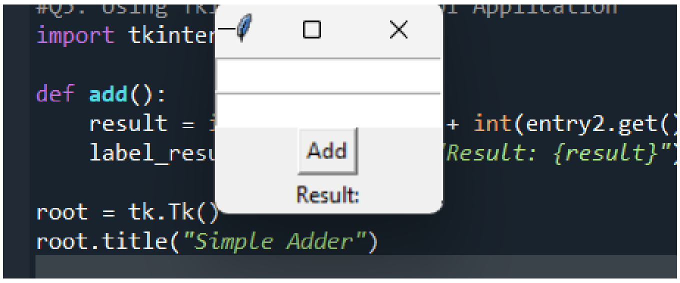
```
import tkinter as tk

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()
```

. This Python code creates a **simple GUI app** using Tkinter.

. It has **two input fields** for numbers and a **button** to add them.

. When the **"Add" button** is clicked, it reads the inputs, adds them, and displays the **result**.

. root.mainloop() keeps the window running.



```
#Q5: Using Tk                 I Application
import tkinter

def add():
    result = i                + int(entry2.get(
    label_resu                'Result: {result}"

root = tk.Tk()
root.title("Simple Adder")
```

Conclusion

This code shows basic Python skills in a simple and clear way. It includes arithmetic operations, solving quadratic and linear equations, drawing graphs using Matplotlib, a recursive function to find factorial, and a simple GUI adder made with Tkinter.

The project combines math, visuals, and user input, making it a good learning experience for beginners.

THANK YOU