Python Projects Report

Submitted by:

Farhim Aftab

Roll No: CS23BCAGN038

Semester 4th , BCA, SCS

Submitted to:

Mr. Nawab Wasim Rahman

The Assam Kaziranga University

Date: 18th May 2025

Acknowledgment

I would like to sincerely thank my respected faculty member, Mr. Nawab Wasim Rahman , for his valuable guidance, continuous support, and constructive feedback throughout the preparation of this Python Projects Report. His assistance greatly contributed to enhancing my understanding and successful completion of these assignments.

Declaration

I hereby declare that this Python Projects Report is a result of my own efforts and work. It has not been submitted, either in part or in full, to any other institution or for any other purpose. All sources and references used have been properly acknowledged.

Name: Farhim Aftab
Roll no: CS23BCAGN038

Semester 4th , BCA, SCS

Table of Contents

Abstract

This report presents a collection of five foundational Python mini-projects that demonstrate the practical application of core programming concepts. Each project is focused on enhancing specific skills: arithmetic and quadratic operations, solving linear equations, graph plotting using mathematical functions, function creation, and graphical user interface (GUI) development using Tkinter. These programs were developed using Python 3.x, leveraging built-in libraries like math, matplotlib, and tkinter. The goal of this report is to bridge theoretical knowledge with practical implementation, enabling a deeper understanding of Python's capabilities in mathematical computation, data visualization, and user interaction design. This hands-on experience has laid a solid groundwork for more advanced Python development in future academic and real-world projects.

1. Introduction:

This report consists of five Python-based mini projects. These projects are designed to improve understanding of mathematical logic, Python syntax, function implementation, graphical visualization using math functions, and GUI development using Tkinter. The programs are simple yet foundational for anyone learning Python.

2. Objective:

The main objective of this report is to :
- Perform arithmetic and quadratic operations
- Solve linear equations
- Plot mathematical graphs
- Implement Python functions
- Design a basic GUI using Tkinter

3. Software & Tools Used:

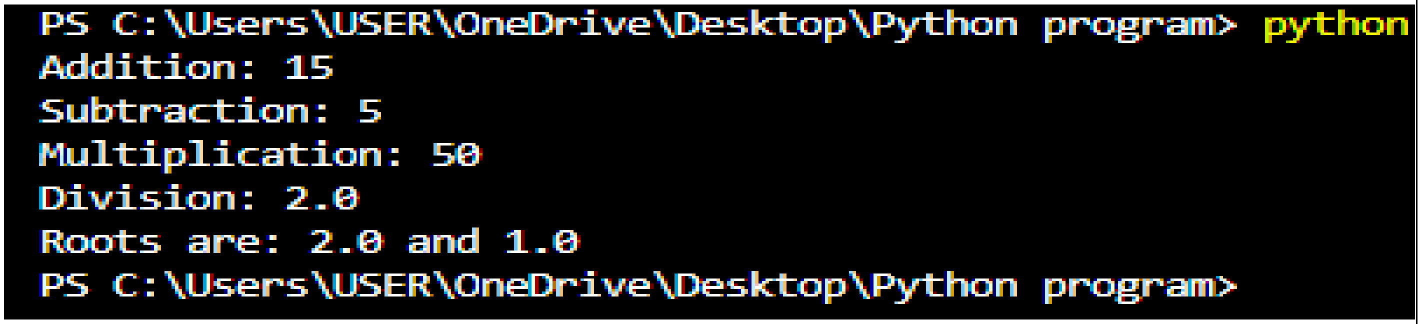
- Programming Language: Python 3.x
- Libraries: math, matplotlib.pyplot, tkinter
- IDE: VS Code / IDLE / Jupyter Notebook
- Operating System: Windows/Linux

4. Program 1: Arithmetic and Quadratic Operations

Code:

```python
#Arithmetic Operations:

a = 10
b = 5
print("Addition:", a + b)   # 10 + 5 = 15
print("Subtraction:", a - b) # 10 - 5 = 5
print("Multiplication:", a * b) # 10 * 5 = 50
print("Division:", a / b) # 10 / 5 = 2.0

#Quadratic Operation:

import math
# Step 1: Set the values for a, b, and c
a = 1
b = -3
c = 2
#Step 2: Calculate the discriminant
d = b**2 - 4*a*c

# Step 3: Check if the discriminant is greater than or equal to 0
if d >= 0:
# if d is non-negative, calculate the real roots
    root1 = (-b + math.sqrt(d)) / (2*a)
    root2 = (-b - math.sqrt(d)) / (2*a)
    print("Roots are:", root1, "and", root2)
else:
# If d is negative, the roots are imaginary
    print("Roots are imaginary (complex numbers)")
```

Output:



```
PS C:\Users\USER\OneDrive\Desktop\Python program> python
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
Roots are: 2.0 and 1.0
PS C:\Users\USER\OneDrive\Desktop\Python program>
```

5. Program 2: Linear Equation Solver (ax + b = c)

Code:

```
# y = mx + c
m = 2
c = 3
x = 5
y = m * x + c
print("Value of y for x=5 is:", y)
```

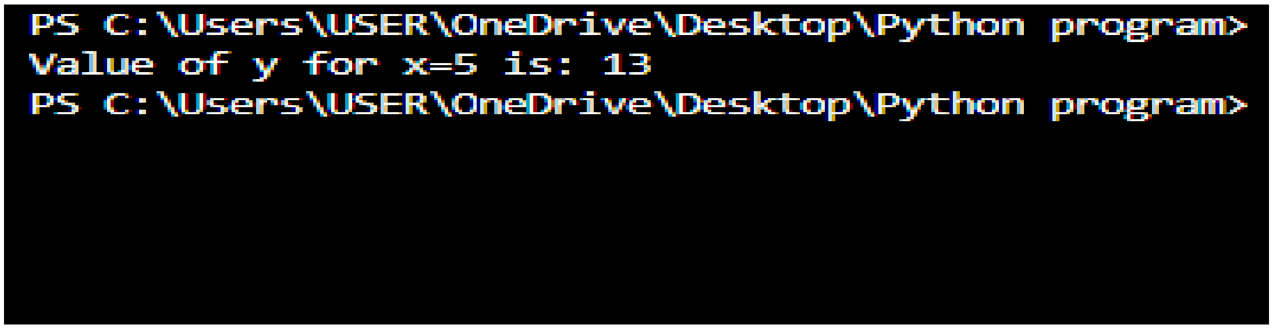
Equation: y = mx + c

This is the formula for a **straight line (linear equation)**.

What each letter means:

- **m** = the **slope** of the line
(tells how steep the line is — going up or down)
- **c** = the **y-intercept**
(the point where the line crosses the y-axis)
- **x** = the **input value**
(you choose this — it's the value you plug in)
- **y** = the **output value**
(this is the result you get after calculating)
- $y=mx+c=2\times5+3=10+3=13$

Output:



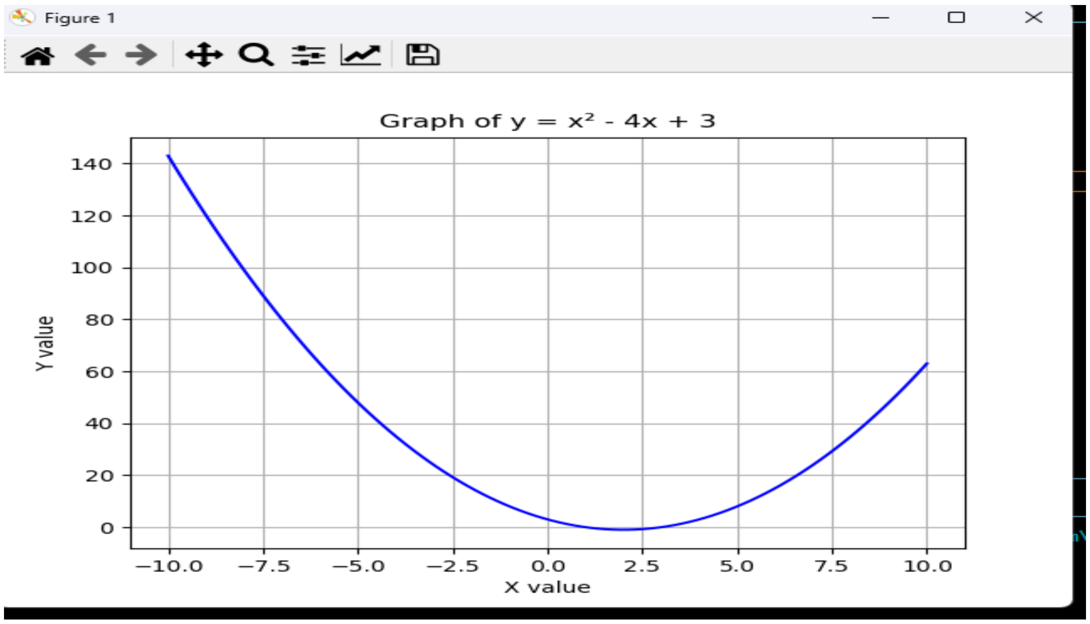
```
PS C:\Users\USER\OneDrive\Desktop\Python program>
Value of y for x=5 is: 13
PS C:\Users\USER\OneDrive\Desktop\Python program>
```

6. Program 3: Graph Plot using Math Functions

Code:

```
# Step 1: Import the libraries needed for graphing
import matplotlib.pyplot as plt  # Used for plotting the graph

import numpy as np               # Used for math and creating x values

# Step 2: Create x values (from -10 to 10, with 100 points)
x = np.linspace(-10, 10, 100)

# Step 3: Calculate y values using the quadratic equation
# Equation: y = x² - 4x + 3
y = x**2 - 4*x + 3

# Step 4: Plot the graph
plt.plot(x, y, color='blue')  # Plot x vs y in blue color

# Step 5: Add a title and labels to the graph
plt.title("Graph of y = x² - 4x + 3") # Title of the graph
plt.xlabel("X value")           # Label for x-axis
plt.ylabel("Y value")           # Label for y-axis

# Step 6: Turn on the grid (makes graph easier to read)
plt.grid(True)
# Step 7: Show the graph
plt.show()
```

Output:



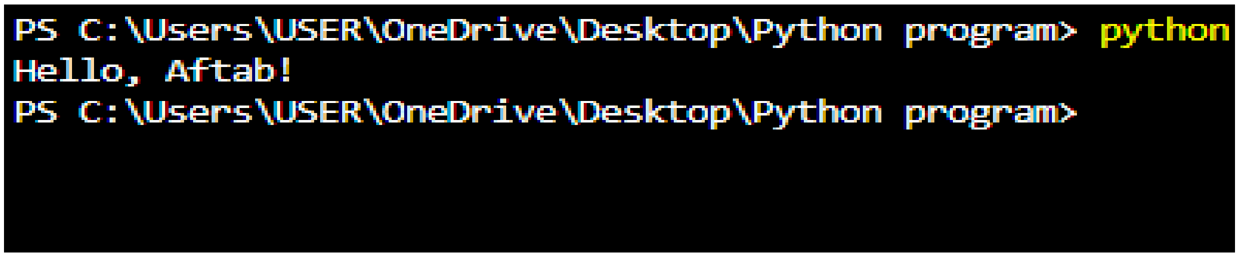
7. Program 4: Python Function

Code:

```python
def greet(name):
    return f"Hello, {name}!"
print(greet("Aftab"))
```

This program defines a function called greet() that takes a name as input and returns a greeting message. It uses Python's **f-string formatting** to insert the name into the message. When called with "Aftab", it outputs: Hello, Aftab!

Output:



```
PS C:\Users\USER\OneDrive\Desktop\Python program> python
Hello, Aftab!
PS C:\Users\USER\OneDrive\Desktop\Python program>
```

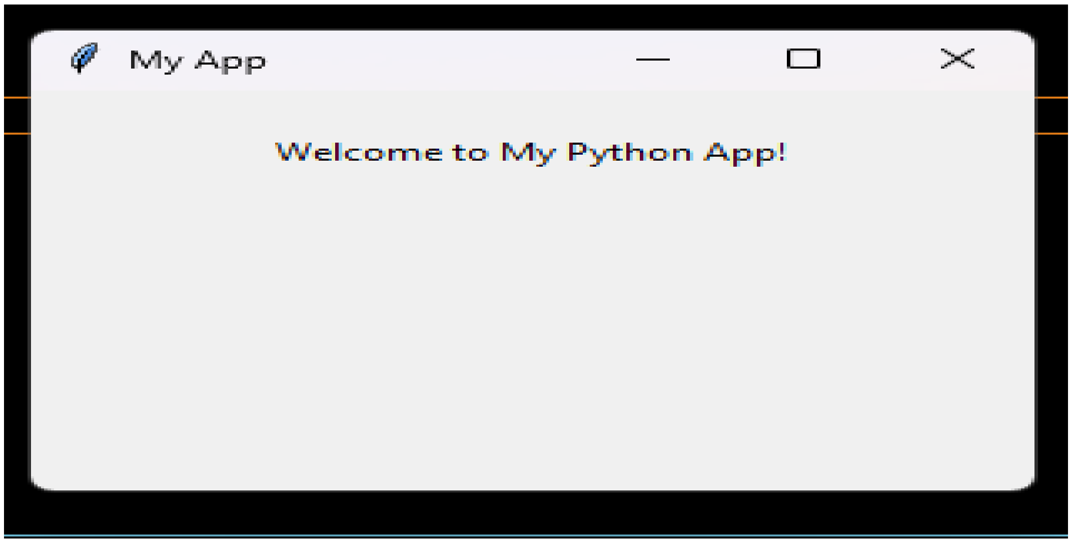
8. Program 5: Tkinter App Example (Basic Window)

Code:

```python
import tkinter as tk

root = tk.Tk()
root.title("My App")
root.geometry("300x200")

label = tk.Label(root, text="Welcome to My Python App!")
label.pack(pady=20)

root.mainloop()
```

This program creates a simple GUI application using Python's **Tkinter** library. It initializes a window titled *"My App"* with a fixed size of 300x200 pixels. A label displaying the text *"Welcome to My Python App!"* is added to the window. The mainloop() function keeps the window open and responsive.

Output:



Welcome to My Python App!

9. Conclusion

Through these five Python programs, I gained hands-on experience in:

- Performing basic and advanced mathematical operations
- Defining and using functions
- Visualizing data using graphical plots
- Creating graphical user interfaces with Tkinter

These projects significantly improved my confidence in writing clean, functional Python code and strengthened my understanding of core programming concepts.