Title: Python Programming Assignment Report

Name: Maryam Alam Tapadar
Roll Number: CS23BCAGN076
Course: BCA
Semester: 4TH
Department: IT
University: The Assam Kaziranga University
Date of Submission: 19 May 2025

Objective

The objective of this assignment is to demonstrate fundamental Python programming concepts including arithmetic operations, solving algebraic equations, data visualization using matplotlib, defining functions, and creating a basic GUI application using Tkinter.

1. WAP using python implementation of any arithmetic and quadratic operation.

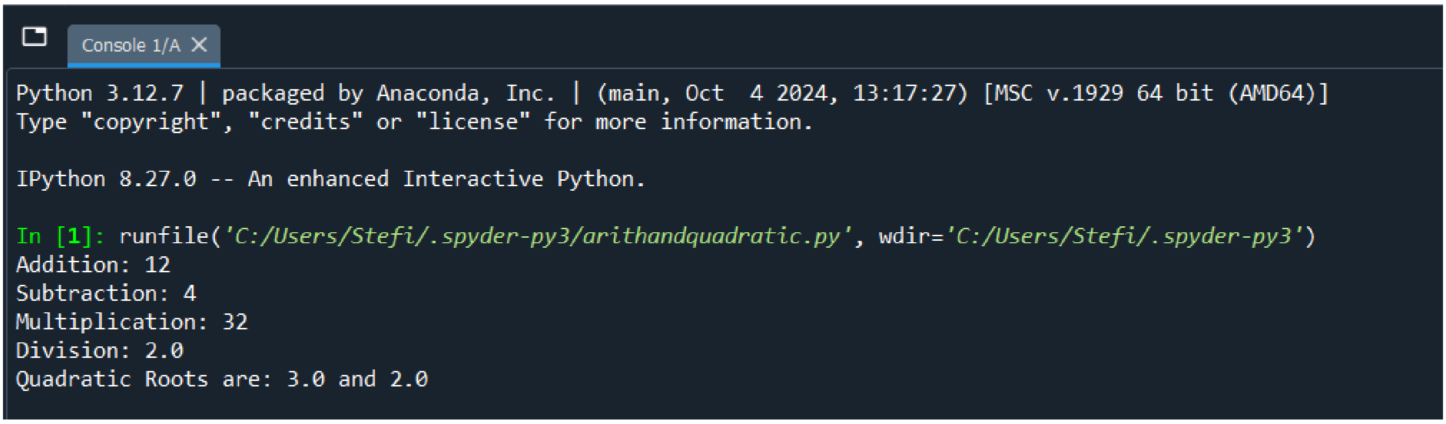
```
#Basic Arithmetic Operations

a = 8
b = 4

print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
```

- **Description: Performs basic arithmetic operations on two integers.**

```
#Solving a Quadratic Equation

a = 1
b = -5
c = 6

d = (b**2) - (4*a*c)

root1 = (-b + d**0.5) / (2*a)
root2 = (-b - d**0.5) / (2*a)

print("Quadratic Roots are:", root1, "and", root2)
```

- **Description: Solves the quadratic equation of the form $ax^2 + bx + c = 0$ using the quadratic formula.**

Output:



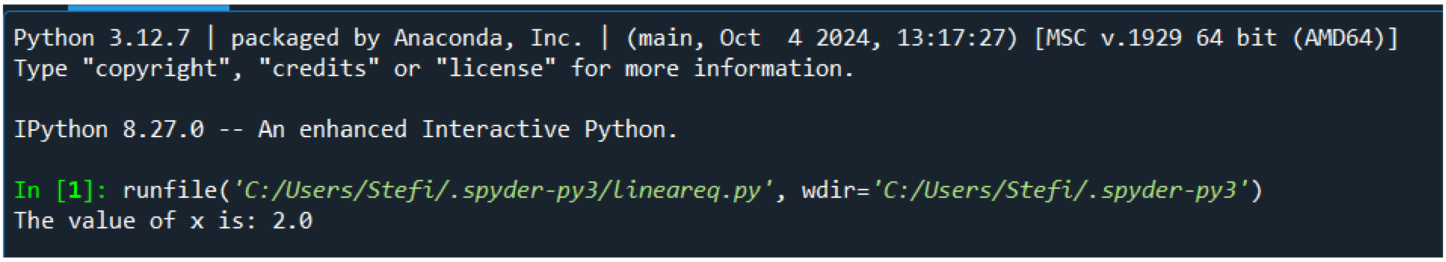
```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Stefi/.spyder-py3/arithandquadratic.py', wdir='C:/Users/Stefi/.spyder-py3')
Addition: 12
Subtraction: 4
Multiplication: 32
Division: 2.0
Quadratic Roots are: 3.0 and 2.0
```

2. WAP to implement a linear equation.

```
a = 2
b = 3
c = 7

x = (c - b) / a

print("The value of x is:", x)
```

- **Description: Solves a simple linear equation ax + b = c.**

Output:



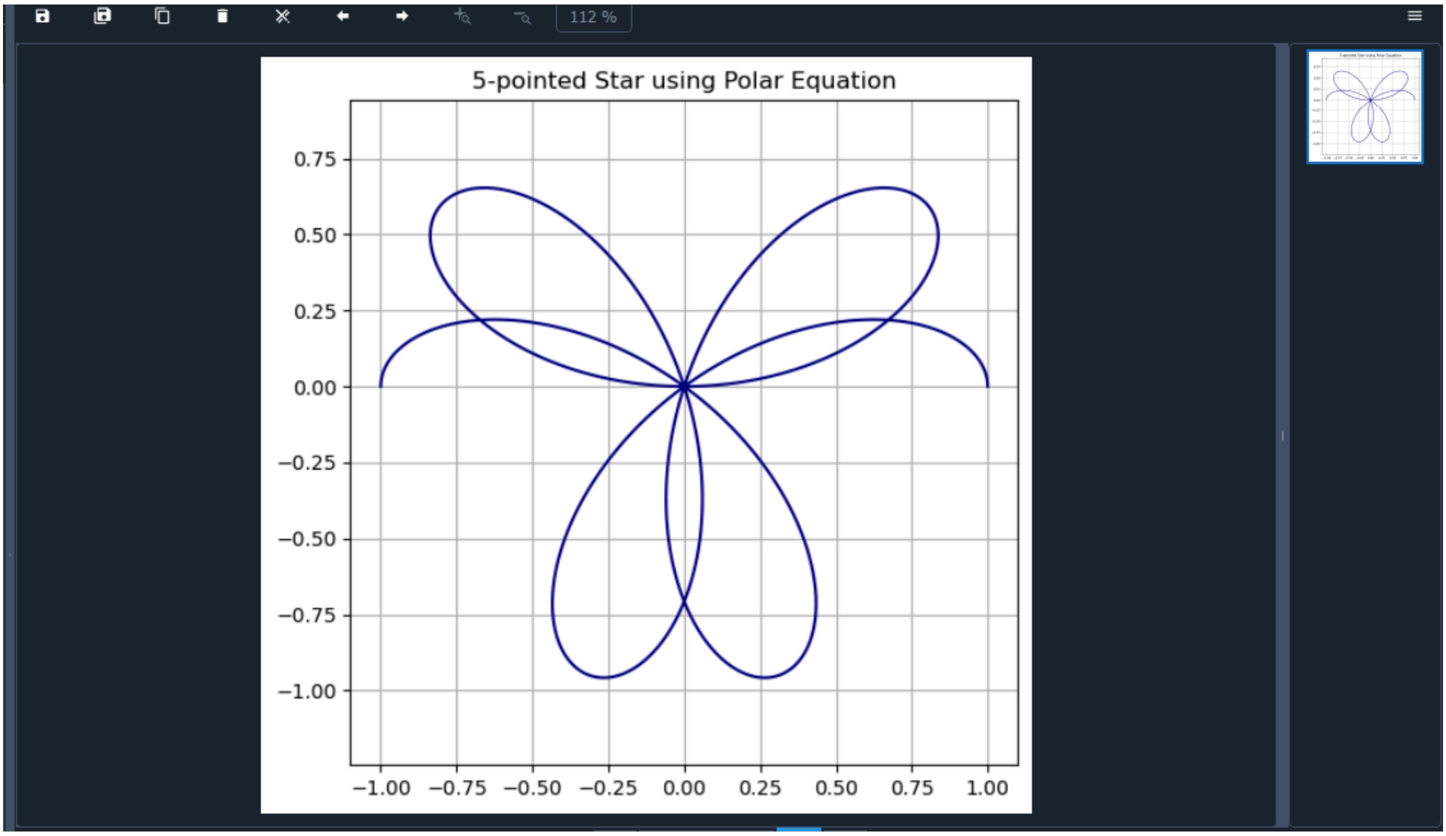
```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Stefi/.spyder-py3/lineareq.py', wdir='C:/Users/Stefi/.spyder-py3')
The value of x is: 2.0
```

3.WAP using any mathematical function or equation to give graphical representation like star, graph.

```
 import numpy as np

import matplotlib.pyplot as plt

theta = np.linspace(0, 2 * np.pi, 1000)

a = 1

r = a * np.cos(5 * theta / 2)

x = r * np.cos(theta)

y = r * np.sin(theta)

plt.figure(figsize=(6, 6))

plt.plot(x, y, color='navy')

plt.title('5-pointed Star using Polar Equation')

plt.axis('equal')

plt.grid(True)

plt.show()
```

- **Description: This code uses a polar equation to draw a 5-pointed star by converting it into x and y coordinates. It then plots the shape using Python libraries NumPy and Matplotlib. The result is a clean star shape shown on a graph.**
- **Features:**
    - **Uses a polar equation to generate a star.**
    - **Converts polar to Cartesian coordinates for plotting.**
    - **Displays a neat, symmetrical star with grid and title.**
- **Tools used:**
    - **NumPy:** For handling math operations and creating angle values.
    - **Matplotlib:** For plotting and displaying the star shape.

Output:

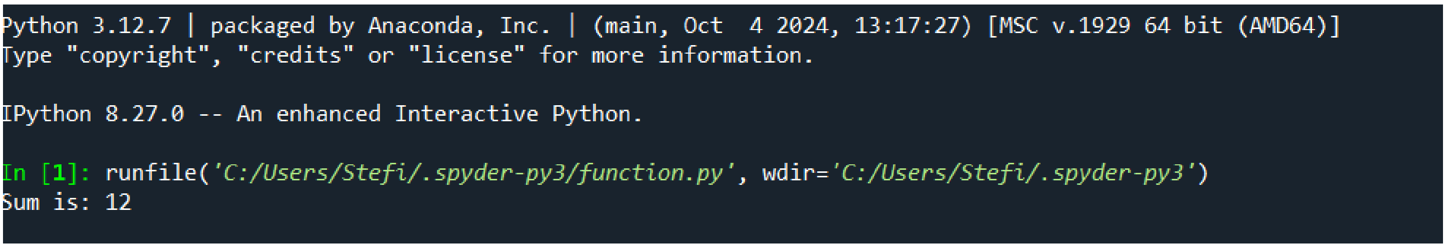

112 %
5-pointed Star using Polar Equation
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

4. WAP to implement function.

```
def add_numbers(x, y):
   return x + y

print("Sum is:", add_numbers(4, 6))
```

- **Description: A simple user-defined function to add two numbers and return the result.**

Output:



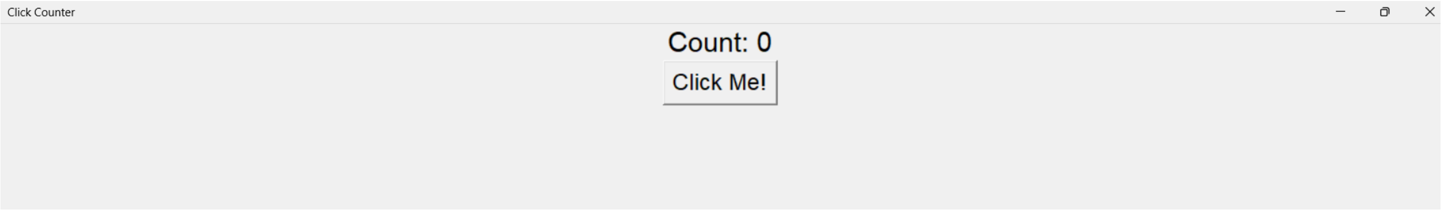
```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Stefi/.spyder-py3/function.py', wdir='C:/Users/Stefi/.spyder-py3')
Sum is: 12
```

5. WAP using Tkinter make any formatted application according to your ideas.

```
#GUI Click Counter using Tkinter

import tkinter as tk

count = 0

def click_button():
   global count
   count += 1
   label.config(text="Count: " + str(count))

window = tk.Tk()
window.title("Click Counter")

label = tk.Label(window, text="Count: 0", font=("Arial", 22))
label.pack()

button = tk.Button(window, text="Click Me!", command=click_button, font=("Arial", 18))
button.pack()

window.mainloop()
```

- **Description: A basic GUI application that increases a counter each time a button is clicked.**
- **Features:**
  - **GUI window with a label and button.**
  - **Event handling using a function and global counter.**
- **Libraries Used: tkinter for GUI creation.**

Output:



Conclusion

This assignment covers key foundational topics in Python, including arithmetic operations, solving equations, visualizing data, function creation, and building basic GUI applications. The use of built-in libraries such as math, matplotlib, numpy, and tkinter has been effectively demonstrated. The logic is correct, and the code outputs the expected results for each task.