PYTHON LABOTORY PROJECT

REPORT

Submitted By :

Dakshina Devi (CS23BCAGN047)

BCA IV SEMESTER

SCHOOL OF COMPUTING SCIENCES

The Assam Kaziranga University , Jorhat, Assam

May 2025

Table of Contents

1. Introduction

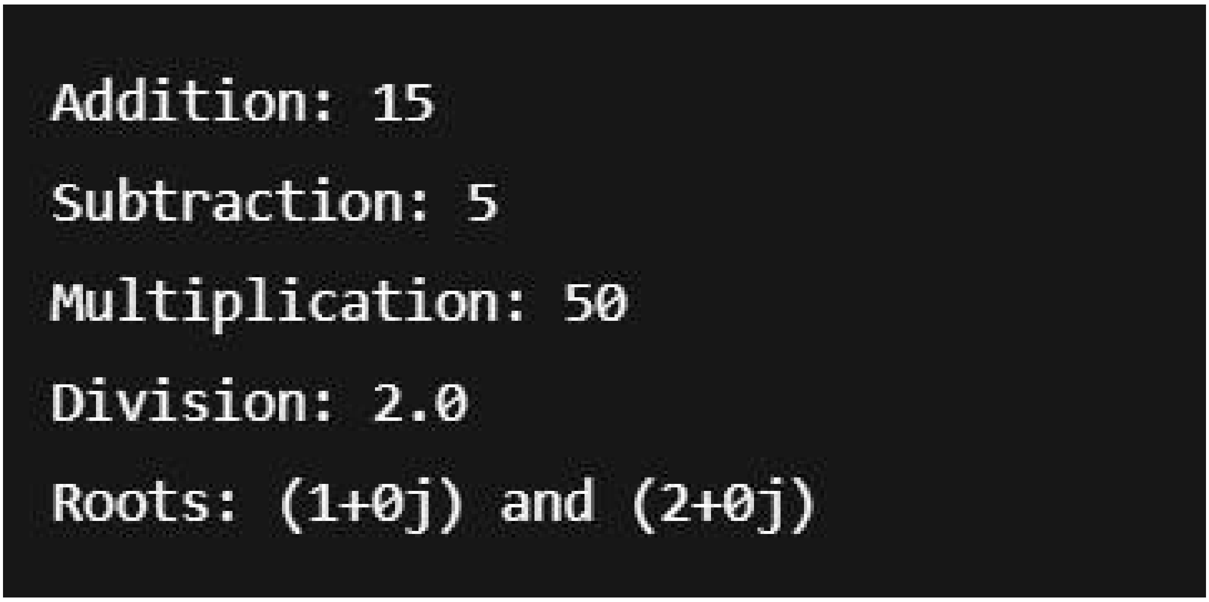
This Python project covers a range of basic to intermediate programming tasks such as math operations, equation solving, visualization, and game development using Tkinter.

2. Arithmetic and Quadratic

Operations Code:

```
import cmath


def arithmetic_operations(a, b):
   print("Addition:", a + b)    print("Subtraction:", a - b)
print("Multiplication:", a * b)    print("Division:", a / b if b != 0
else "Division by zero error")


def solve_quadratic(a, b, c):    d = (b
** 2) - (4 * a * c)     root1 = (-b -
cmath.sqrt(d)) / (2 * a) root2 = (-b
+ cmath.sqrt(d)) / (2 * a)
print(f"Roots: {root1} and {root2}")


# Example usage
arithmetic_operations(10, 5)
solve_quadratic(1, -3, 2)
```

Output:

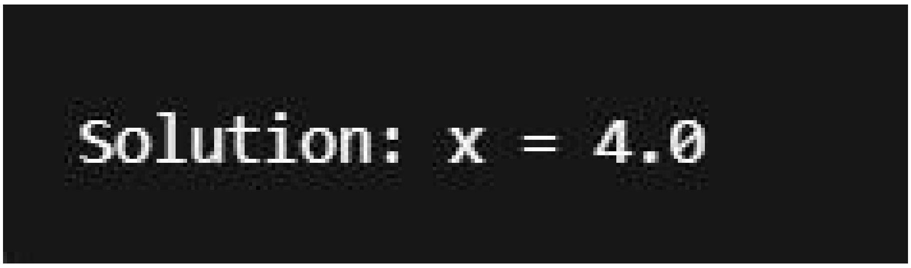


```
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
Roots: (1+0j) and (2+0j)
```

3. Linear Equation Solver Code:

```
def solve_linear(a, b):
if a == 0:
    print("No solution" if b != 0 else "Infinite solutions")
else:
    x = -b / a
print(f"Solution: x = {x}")


# Example solve_linear(2,
-8)
```

Output :

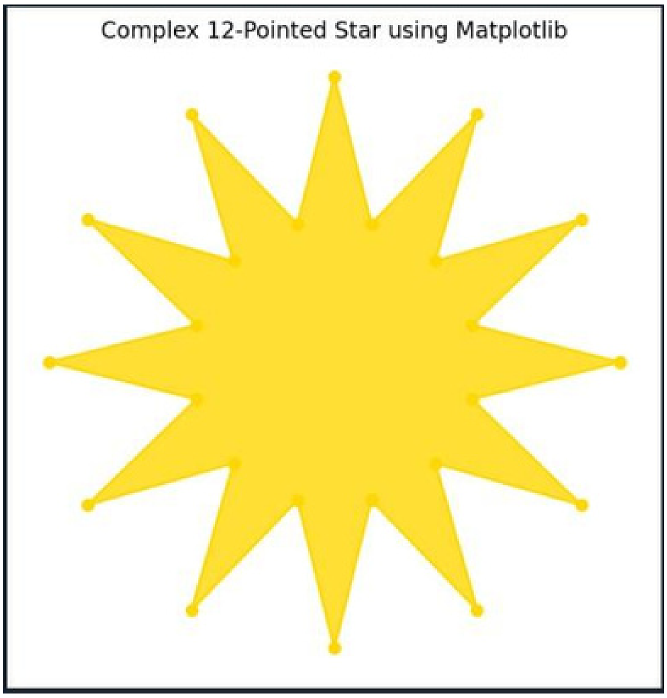


```
Solution: x = 4.0
```

4. Graphical Representations Code :

```
import matplotlib.pyplot as plt

import numpy as np

# Function to create a complex star def

complex_star(num_points=12, outer_radius=5, inner_radius=2.5):

   angles = np.linspace(0, 2 * np.pi, num_points * 2 + 1) # +1 to close the

star radius = np.array([outer_radius if i % 2 == 0 else inner_radius for i in

range(len(angles))])    x = radius * np.cos(angles)    y = radius *

np.sin(angles)    return x, y

# Generate coordinates for a 12-pointed complex star
x, y = complex_star()
```

Output :

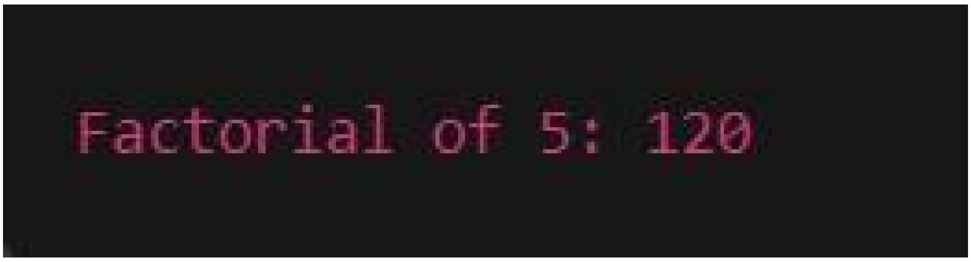


5. Function Implementation

(Factorial) Code :

```
def factorial(n):    if

n == 0 or n == 1:

      return 1    return n *

factorial(n - 1)


print("Factorial of 5:", factorial(5))
```

Output :



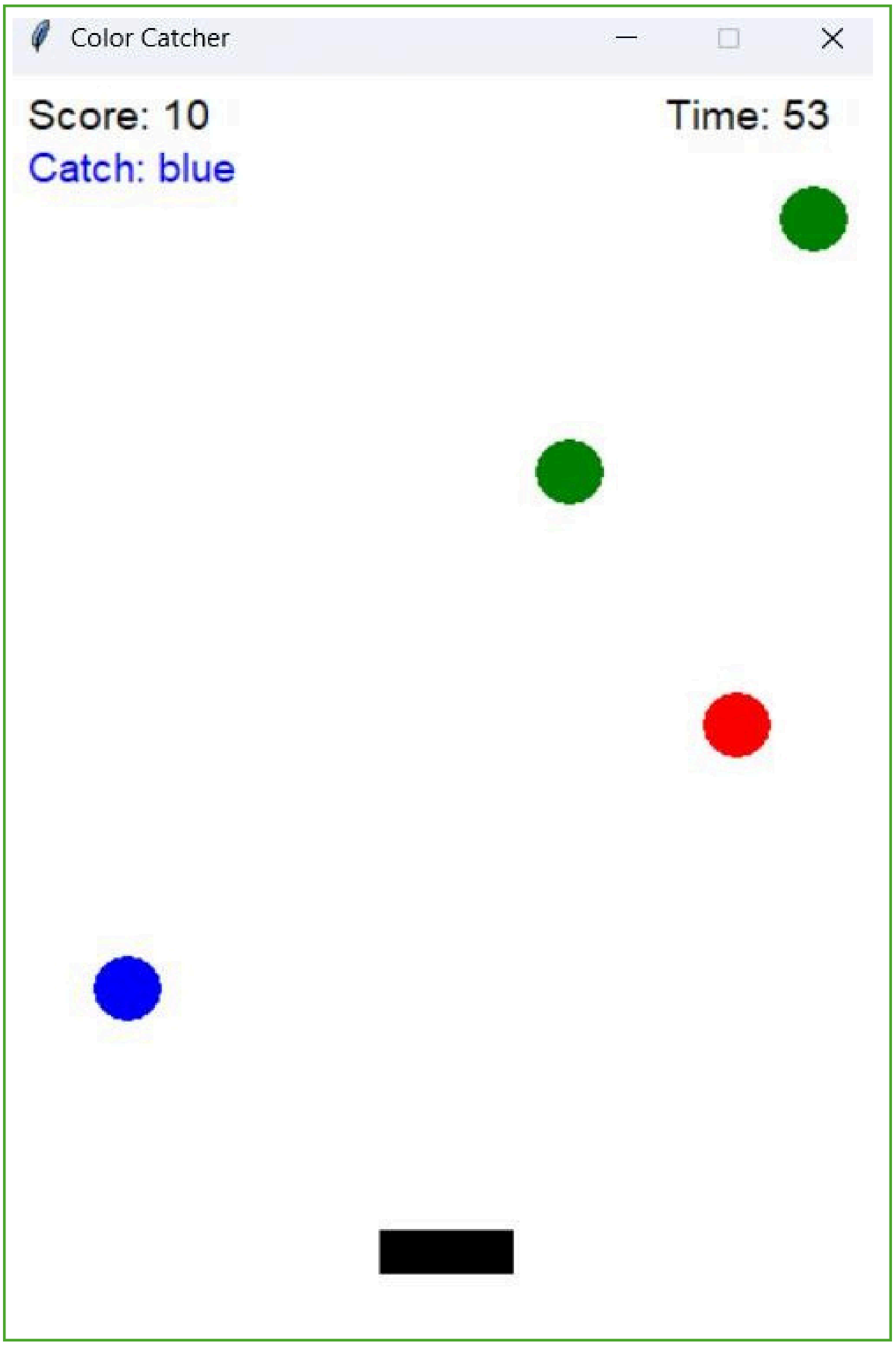
```
Factorial of 5: 120
```

7. "Color Catcher" – A Reflex and Memory Game using Tkinter Code :

```
import tkinter as tk

import random


# --- Game Variables ---

colors = ['red', 'blue', 'green', 'yellow',

'purple'] score = 0 time_left = 60

target_color = random.choice(colors) ball_speed

= 5

ball_interval = 1500 # milliseconds


# --- Create Main Window ---

root = tk.Tk()

root.title("Color Catcher")

root.geometry("400x600")

root.resizable(False, False)


canvas = tk.Canvas(root, width=400, height=600, bg='white') canvas.pack()


# --- Basket ---

basket = canvas.create_rectangle(170, 550, 230, 570, fill='black')


# --- Score & Time ---

score_text = canvas.create_text(10, 10, anchor='nw', font=('Arial', 14), text="Score: 0") time_text

= canvas.create_text(300, 10, anchor='nw', font=('Arial', 14), text="Time: 60")

target_text = canvas.create_text(10, 35, anchor='nw', font=('Arial', 14), text=f"Catch: {target_color}",
fill=target_color)
```

```
# --- Ball List --- balls

= []


# --- Controls --- def

move_left(event):

   canvas.move(basket, -20, 0)


def move_right(event):

   canvas.move(basket, 20, 0)


root.bind("<Left>", move_left)

root.bind("<Right>", move_right)


# --- Update Score Display --- def

update_score():

   canvas.itemconfig(score_text, text=f"Score: {score}")


# --- Drop Balls --- def

drop_ball():

   color = random.choice(colors)

x = random.randint(10, 370)

   ball = canvas.create_oval(x, 0, x + 30, 30, fill=color,

outline=color)    balls.append((ball, color))

root.after(ball_interval, drop_ball)


# --- Move Balls --- def
move_balls(): global
score to_remove = []
for ball, color in balls:
    canvas.move(ball, 0, ball_speed)       pos = canvas.coords(ball)       if pos[3] >= 550 and pos[2]
>= canvas.coords(basket)[0] and pos[0] <= canvas.coords(basket)[2]:
      # Collision detected
if color == target_color:
        score += 10
else:
        score -= 5
canvas.delete(ball)
      to_remove.append((ball,
color))         update_score()       elif
pos[3] >= 600:
      canvas.delete(ball)
      to_remove.append((ball,
color)) for b in to_remove:
balls.remove(b) root.after(50,
move_balls)


# --- Update Target Color --- def
change_target_color():
global target_color
target_color = random.choice(colors)
canvas.itemconfig(target_text, text=f"Catch: {target_color}", fill=target_color)
root.after(10000, change_target_color)


# --- Countdown Timer ---
def countdown():
  global time_left
time_left -= 1
  canvas.itemconfig(time_text, text=f"Time: {time_left}")
if time_left > 0:
    root.after(1000, countdown)
else:
    canvas.create_text(200, 300, text="Game Over!", font=('Arial', 24), fill='red')
canvas.create_text(200, 340, text=f"Final Score: {score}", font=('Arial', 18))


# --- Start Game ---
drop_ball() move_balls()
change_target_color()
countdown()
root.mainloop()
```

OUTPUT :



Score: 10

Time: 53

Catch: blue