Ex. No. : 3 Date:

Register No.: 231701045 Name: P. Sahaana

2D Transformations – Translation, Scaling, and Rotation AIM:

To write a program that performs:

* Translation
* Scaling
* Rotation on basic 2D shapes (e.g., triangle, rectangle) using transformation matrices.

Procedure:

1. Define the 2D shape using its vertices.
2. Use matrix multiplication to perform:
   * Translation by adding offsets to coordinates. o Scaling by multiplying coordinates with scale factors.
   * Rotation by applying rotation matrix.
3. Display original and transformed shapes.

Program:

import numpy as np import

matplotlib.pyplot as plt

def draw\_shape(points, label, color): x, y = zip(\*points) x += (x[0],) y +=

(y[0],) plt.plot(x, y, color=color, label=label)

def translate(points, tx, ty):

T = np.array([[1, 0, tx],

[0, 1, ty],

[0, 0, 1]])

return apply\_transform(points, T)

def scale(points, sx, sy):

S = np.array([[sx, 0, 0],

[0, sy, 0], [0, 0, 1]])

return apply\_transform(points, S)

def rotate(points, angle\_deg): angle\_rad = np.radians(angle\_deg)

R = np.array([[np.cos(angle\_rad), -np.sin(angle\_rad), 0],

[np.sin(angle\_rad), np.cos(angle\_rad), 0],

[0, 0, 1]])

return apply\_transform(points, R)

def apply\_transform(points, matrix): transformed = [] for x, y in points: vec = np.array([x, y, 1]) result = matrix @ vec

transformed.append((result[0], result[1])) return transformed

# Original triangle triangle = [(0, 0),

(100, 0), (50, 80)]

# Transformations

translated = translate(triangle, 120, 50) scaled

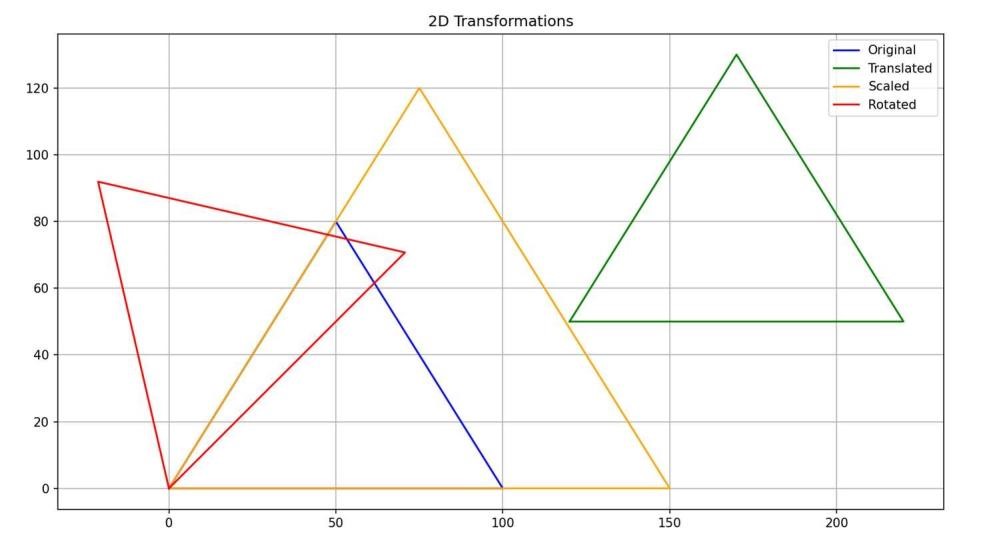
= scale(triangle, 1.5, 1.5) rotated = rotate(triangle, 45)

# Plot

plt.figure(figsize=(8, 8)) draw\_shape(triangle, "Original", 'blue') draw\_shape(translated, "Translated", 'green') draw\_shape(scaled, "Scaled", 'orange') draw\_shape(rotated, "Rotated", 'red')

plt.title("2D Transformations")

plt.legend() plt.grid(True) plt.axis("equal") plt.show()



Result:

The 2D transformations (translation, scaling, and rotation) were successfully applied using matrix operations on a triangle.