Nachiket Abhay Vaidya

Test ID: 450011854000582 | 7066131709 | nachiket.vaidya24@vit.edu

Test Date: May 13, 2025

Logical Ability	Quantitative Ability (Advanced)	English Comprehension
54 /100	78 /100	68 /100

Logical Ability

54 / 100

Inductive Reasoning	Deductive Reasoning	Abductive Reasoning
55 / 100	55 / 100	53 / 100

Quantitative Ability (Advanced)

78 / 100

Basic Mathematics	Advanced Mathematics	Applied Mathematics
76 / 100	78 / 100	81 / 100

English Comprehension

68 / 100 | CEFR: **C1**

Grammar	Vocabulary	Comprehension
71 / 100	73 / 100	61 / 100

1 | Introduction

About the Report

This report provides a detailed analysis of the candidate's performance on different assessments. The tests for this job role were decided based on job analysis, O*Net taxonomy mapping and/or criterion validity studies. The candidate’s responses to these tests help construct a profile that reflects her/his likely performance level and achievement potential in the job role

This report has the following sections:

The **Summary** section provides an overall snapshot of the candidate’s performance. It includes a graphical representation of the test scores and the subsection scores.

The **Insights** section provides detailed feedback on the candidate’s performance in each of the tests. The descriptive feedback includes the competency definitions, the topics covered in the test, and a note on the level of the candidate’s performance.

The **Learning Resources** section provides online and offline resources to improve the candidate's knowledge, abilities, and skills in the different areas on which s/he was evaluated.

Score Interpretation

All the test scores are on a scale of 0-100. All the tests except personality and behavioural evaluation provide absolute scores. The personality and behavioural tests provide a norm-referenced score and hence, are percentile scores. Throughout the report, the colour codes used are as follows:

- Scores between 67 and 100
- Scores between 33 and 67
- Scores between 0 and 33

2 | Insights

English Comprehension

68 / 100 | CEFR: **C1**

This test aims to measure your vocabulary, grammar and reading comprehension skills.

You have a good understanding of commonly used grammatical constructs. You are able to read and understand articles, reports and letters/mails related to your day-to-day work. The ability to read, understand and interpret business-related documents is essential in most jobs, especially the ones that involve research, technical reading and content writing.

Logical Ability

54 / 100

Inductive Reasoning

55 / 100

This competency aims to measure the your ability to synthesize information and derive conclusions.

You are able to work out rules based on specific information and solve general work problems using these rules. This skill is required in data-driven research jobs where one needs to formulate new rules based on variable trends.

Deductive Reasoning

55 / 100

This competency aims to measure the your ability to synthesize information and derive conclusions.

You are able to work out rules based on specific information and solve general work problems using these rules. This skill is required in data-driven research jobs where one needs to formulate new rules based on variable trends.

Abductive Reasoning

53 / 100

Quantitative Ability (Advanced)

78 / 100

This test aims to measure your ability to solve problems on basic arithmetic operations, probability, permutations and combinations, and other advanced concepts.

It is commendable that you are able to understand and solve complex arithmetic problems. You are able to solve basic problems of probability, logarithms, permutations, and combinations. This skill will help you in jobs where one needs to work with statistical data and make probabilistic predictions.

3 | Learning Resources

English Comprehension

- Read opinions to improve your comprehension
- Improve your vocabulary of terms used in business
- Improve your knowledge of Business English

Logical Ability

- Test your application of inductive logic!
- Play Sudoku and develop your skills of deduction!
- Learn about the art of deduction

Quantitative Ability (Advanced)

- Learn about the real life applications of logarithms
- Learn about the application of Bayes' Theorem in varied fields
- Learn about Fermet's last theorem

Icon Index

Free Tutorial | Paid Tutorial | Youtube Video | Web Source

Wikipedia | Text Tutorial | Video Tutorial | Google Playstore

Tutorial No 5

Name : Sarthak Shrikant Bagul

Branch : ENTC

Div :- A

PRN :- 12410555

Roll No :- 25

Subject :- Computer Graphics (MDM)

Title / Problem Statement:

2D Transformations in Computer Graphics – Translation, Scaling, Rotation, Reflection, and Shearing

Understand and apply 2D transformations on graphical objects such as points, lines, and polygons. Explore different types of transformations and their effects on the shape, position, and orientation of objects in 2D space.

Objectives: By the end of this tutorial, students should be able to:

- Define and explain the different types of 2D transformations.
- Apply transformations such as **translation, scaling, rotation, reflection, and shearing** to 2D objects.
- Represent transformations mathematically using **homogeneous matrices**.
- Understand and analyze the effects of transformations on 2D objects in computer graphics.

Prerequisites:

- Knowledge of coordinate geometry (points, lines, polygons).
- Basic linear algebra (matrices, matrix multiplication).
- Basic trigonometry (sine, cosine for rotation).
- Understanding of computer graphics basics (pixels, raster displays).

Theory Overview:

1. **2D Transformations:**
   Transformations are operations that alter the position, size, orientation, or shape of graphical objects in 2D space.

2. **Types of Transformations:**
   (a) **Translation:** Moves an object from one location to another without changing its shape or orientation.

- Formula:

$$x' = x + tx, \quad y' = y + ty$$

(b) **Scaling:** Changes the size of an object relative to a fixed point (usually origin).

- Formula:

$$x' = Sx \cdot x, \quad y' = Sy \cdot y$$

- Scaling around a pivot requires moving the object to origin, scaling, and moving back.

(c) **Rotation:** Rotates an object around a fixed point (usually origin) by angle θ.

- Formula:

$$x' = x\cos\theta - y\sin\theta, \quad y' = x\sin\theta + y\cos\theta$$

(d) **Reflection:** Produces a mirror image of an object about a line (x-axis, y-axis, y=x, etc.).

(e) **Shearing:** Slants the shape of an object along x or y axis.

- X-shear: $x' = x + sh_x \cdot y$
- Y-shear: $y' = y + sh_y \cdot x$

3. **Homogeneous Coordinates:**

- Transformations can be represented by 3×3 matrices to combine multiple operations conveniently:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- This allows **composite transformations** like scaling + rotation + translation in a single operation.

4. **Order of Transformations:**

- The sequence of transformations matters:
    - **Rotation then translation ≠ Translation then rotation**
- Composite transforms often use a **pivot point** to control rotation or scaling.

Algorithm / Steps:

A. Translation:

1. For each point (x, y), calculate:

$$x' = x + tx,\ y' = y + ty$$

2. Plot the new points.

B. Scaling:

1. Select scale factors Sx,Sy.
2. For each point (x, y), calculate:

$$x' = x \cdot Sx\ ,\ \ y' = y \cdot Sy$$

3. Plot scaled points.

C. Rotation:

1. Select rotation angle θ.
2. For each point (x, y), calculate:

$$x' = x\cos\theta - y\sin\theta,\ y' = x\sin\theta + y\cos\theta$$

3. Plot rotated points.

D. Reflection:

1. Choose the axis of reflection (x-axis, y-axis, y=x).
2. Apply the corresponding reflection formula.

E. Shearing:

1. Select shear factors sh_x or sh_y.
2. For each point (x, y), apply:

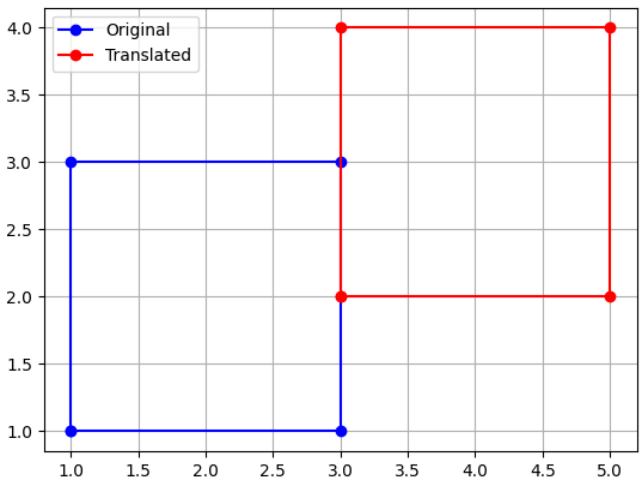
$$x' = x + sh_x \cdot y,\ \ y' = y + sh_y \cdot x$$

3. Plot sheared points.

Code Implementation (Python + Matplotlib)

(i) Translation:

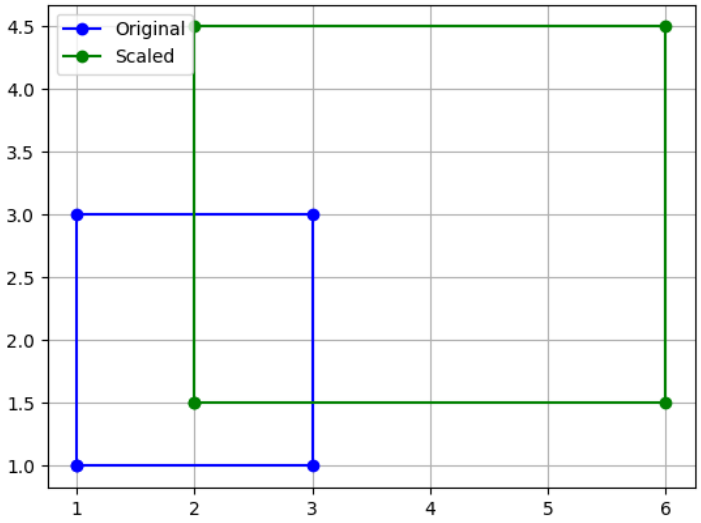
```
import matplotlib.pyplot as plt

def translate(points, tx, ty):

return [(x + tx, y + ty) for x, y in points]

# Original square

square = [(1,1),(1,3),(3,3),(3,1),(1,1)]

translated_square = translate(square, 2, 1)

x_orig, y_orig = zip(*square)

x_trans, y_trans = zip(*translated_square)

plt.plot(x_orig, y_orig, 'b-o', label='Original')

plt.plot(x_trans, y_trans, 'r-o', label='Translated')

plt.legend()

plt.grid(True)

plt.show()
```



(ii) Scaling:

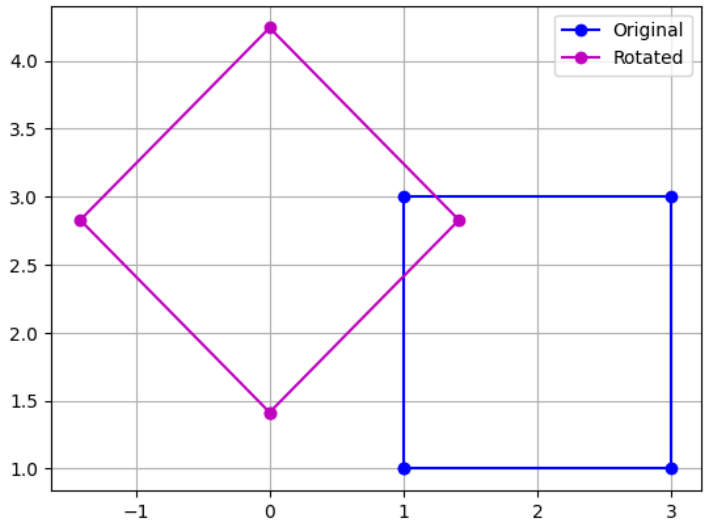
```
def scale(points, sx, sy):

  return [(x * sx, y * sy) for x, y in points]
```

```
scaled_square = scale(square, 2, 1.5)

x_scaled, y_scaled = zip(*scaled_square)

plt.plot(x_orig, y_orig, 'b-o', label='Original')

plt.plot(x_scaled, y_scaled, 'g-o', label='Scaled')

plt.legend()

plt.grid(True)

plt.show()
```



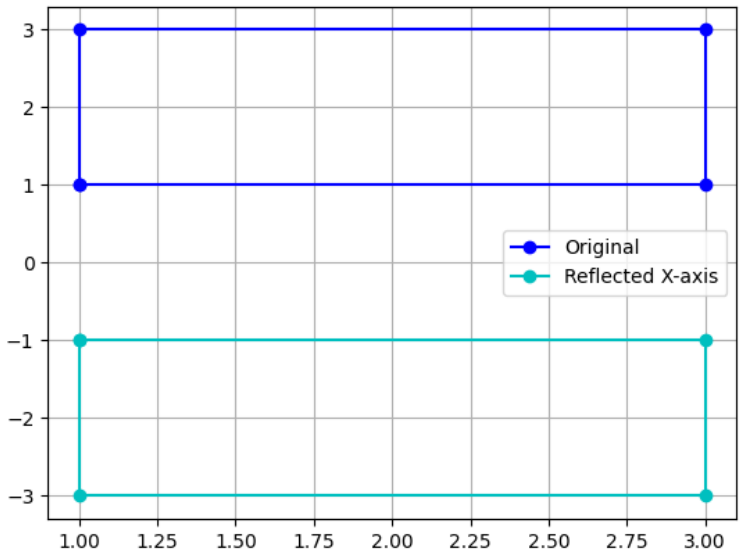
(iii) Rotation:

```
import math

def rotate(points, theta):

    rad = math.radians(theta)

    return [(x*math.cos(rad) - y*math.sin(rad), x*math.sin(rad) + y*math.cos(rad)) for x,y in points]

rotated_square = rotate(square, 45)

x_rot, y_rot = zip(*rotated_square)

plt.plot(x_orig, y_orig, 'b-o', label='Original')

plt.plot(x_rot, y_rot, 'm-o', label='Rotated')

plt.legend()

plt.grid(True)

plt.show()
```



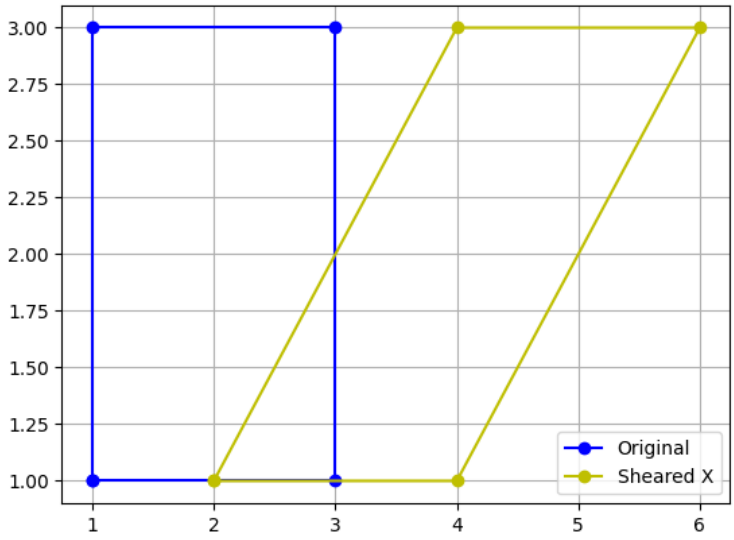
(iv) Reflection (about x-axis):

```
def reflect_x(points):
    return [(x, -y) for x, y in points]
reflected_square = reflect_x(square)
x_ref, y_ref = zip(*reflected_square)
plt.plot(x_orig, y_orig, 'b-o', label='Original')
plt.plot(x_ref, y_ref, 'c-o', label='Reflected X-axis')
plt.legend()
plt.grid(True)
plt.show()
```



(v) Shearing (x-direction):

```
def shear_x(points, shx):
    return [(x + shx*y, y) for x, y in points]
sheared_square = shear_x(square, 1)
x_shear, y_shear = zip(*sheared_square)
plt.plot(x_orig, y_orig, 'b-o', label='Original')
plt.plot(x_shear, y_shear, 'y-o', label='Sheared X')
plt.legend()
plt.grid(True)
plt.show()
```



Result / Observation / Conclusion:

- **Translation** moves the object without changing shape.
- **Scaling** changes object size; non-uniform scaling distorts shape.
- **Rotation** changes orientation while keeping shape and size intact.
- **Reflection** creates mirror images along specified axes.
- **Shearing** slants the object, altering angles but not parallelism.
- **Homogeneous matrices** allow combining multiple transformations efficiently.
- These transformations demonstrate how mathematical operations directly manipulate pixel positions on raster displays.