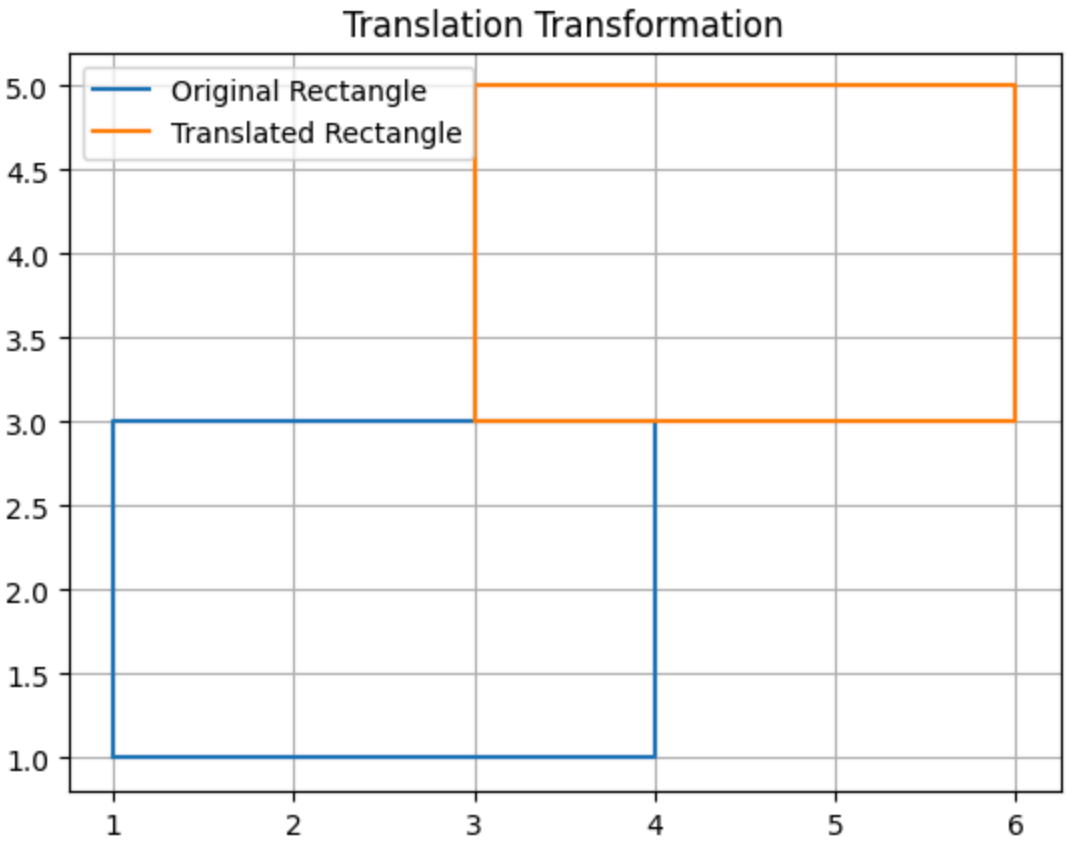
Amit Divekar | Practical 7

Polygon and Shape Transformations

In [1]:
```
import matplotlib.pyplot as plt
from sympy import Polygon, Point, RegularPolygon, pi, Matrix
```

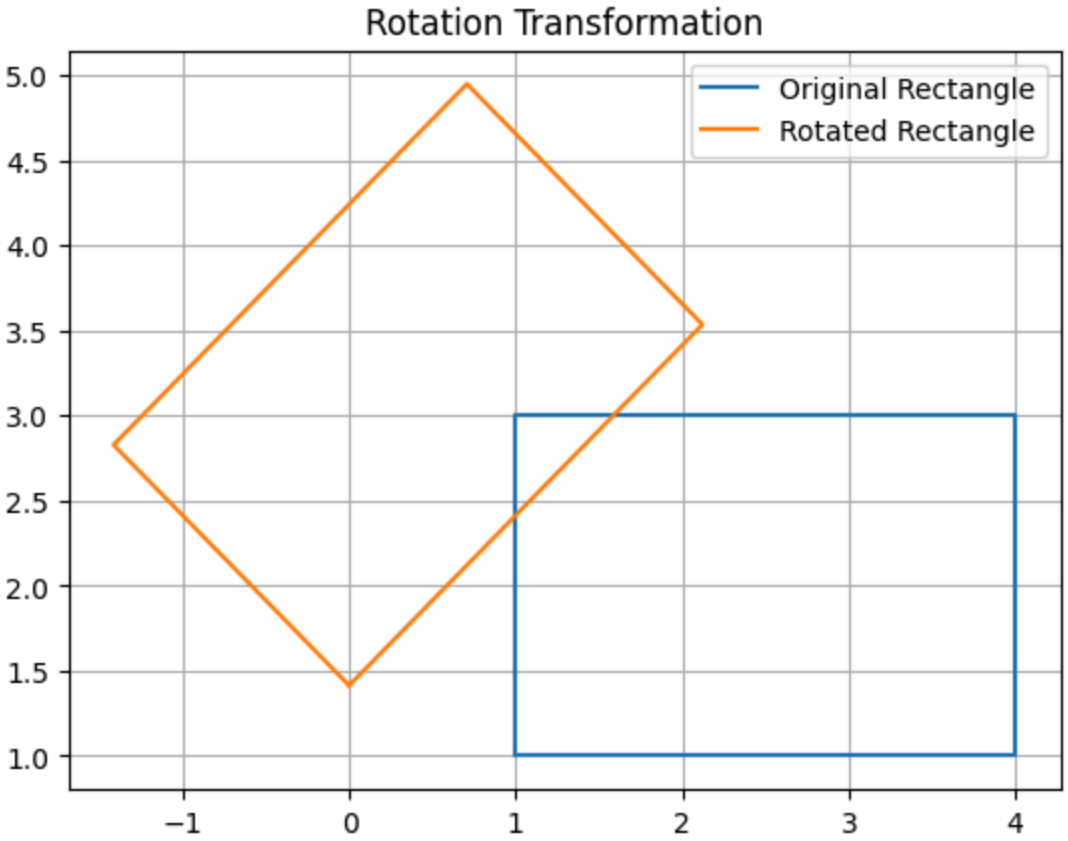
Q1. Plot a rectangle and its translated image.

In [2]:
```
import matplotlib.pyplot as plt
from sympy import Polygon, Point
R = Polygon(Point(1,1), Point(4,1), Point(4,3), Point(1,3))
R_t = R.translate(2, 2)
def draw(P, label):
    x = [p.x for p in P.vertices] + [P.vertices[0].x]
    y = [p.y for p in P.vertices] + [P.vertices[0].y]
    plt.plot(x, y, label=label)
draw(R, "Original Rectangle")
draw(R_t, "Translated Rectangle")
plt.legend()
plt.grid()
plt.title("Translation Transformation")
plt.show()
```



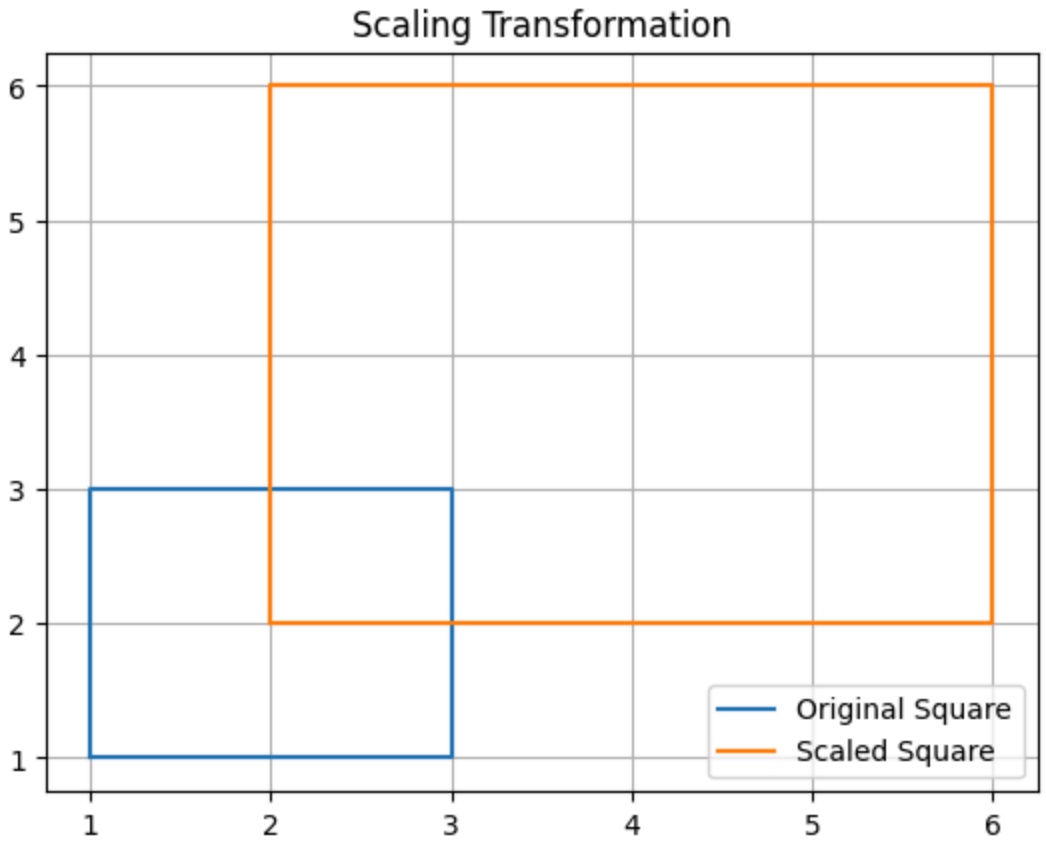
Q2. Plot a rectangle and its rotated image by 45 degrees.

In [3]:
```
import matplotlib.pyplot as plt
from sympy import Polygon, Point, pi
R = Polygon(Point(1,1), Point(4,1), Point(4,3), Point(1,3))
R_r = R.rotate(pi/4)
def draw(P, label):
    x = [p.x for p in P.vertices] + [P.vertices[0].x]
    y = [p.y for p in P.vertices] + [P.vertices[0].y]
    plt.plot(x, y, label=label)
draw(R, "Original Rectangle")
draw(R_r, "Rotated Rectangle")
plt.legend()
plt.grid()
plt.title("Rotation Transformation")
plt.show()
```



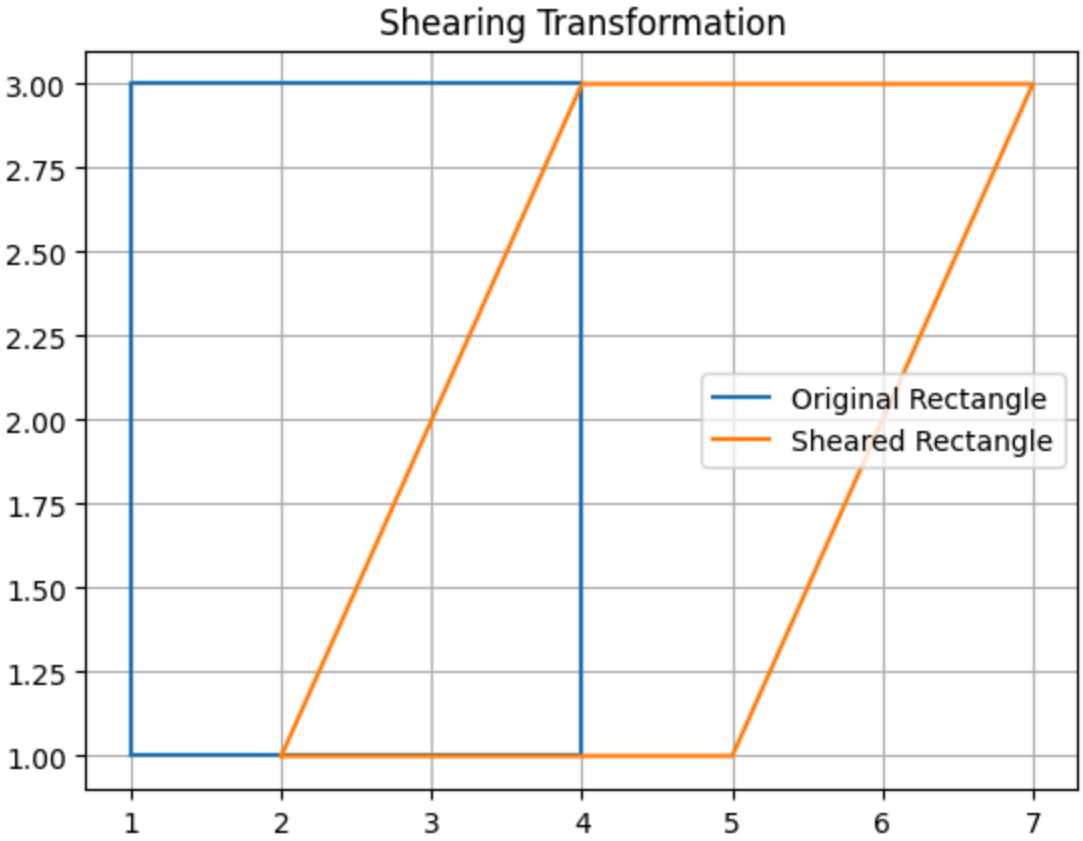
Q3. Plot a square and its scaled image.

In [4]:
```
import matplotlib.pyplot as plt
from sympy import Polygon, Point
S = Polygon(Point(1,1), Point(3,1), Point(3,3), Point(1,3))
S_s = S.scale(2, 2)
def draw(P, label):
    x = [p.x for p in P.vertices] + [P.vertices[0].x]
    y = [p.y for p in P.vertices] + [P.vertices[0].y]
    plt.plot(x, y, label=label)
draw(S, "Original Square")
draw(S_s, "Scaled Square")
plt.legend()
plt.grid()
plt.title("Scaling Transformation")
plt.show()
```



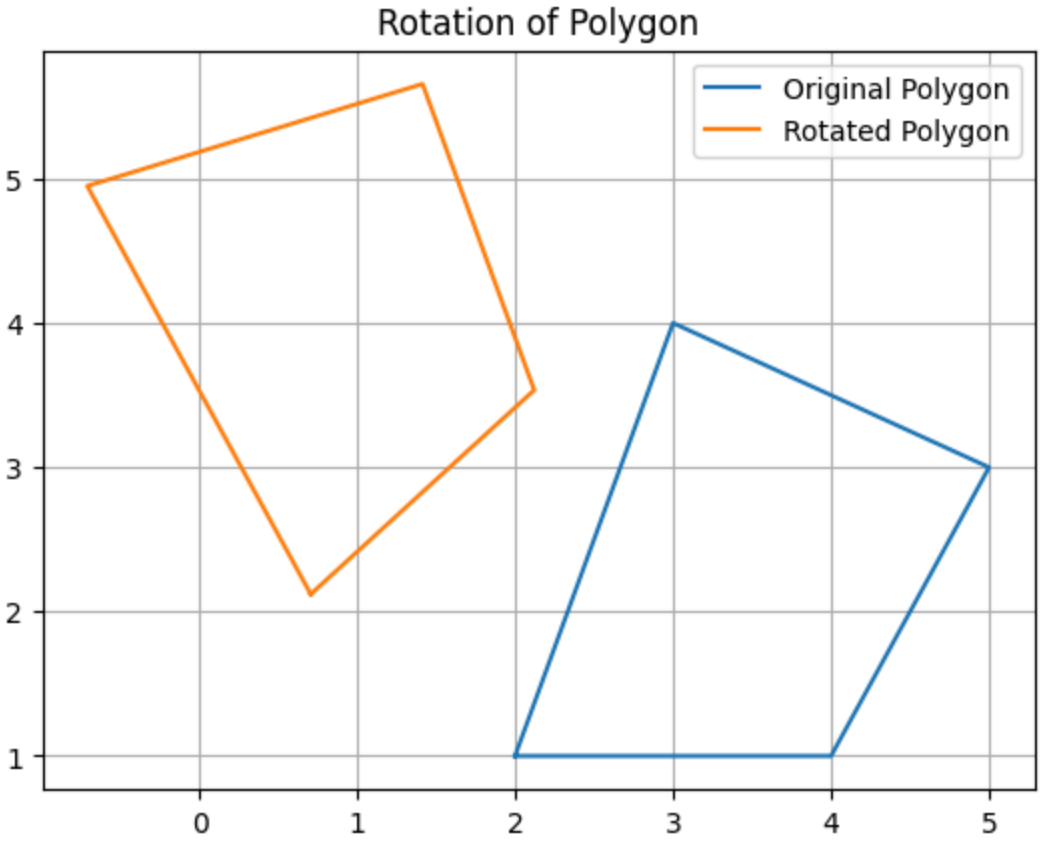
Q4. Plot a rectangle and its sheared image along the x-axis.

In [5]:
```
import matplotlib.pyplot as plt
from sympy import Polygon, Point, Matrix
R = Polygon(Point(1,1), Point(4,1), Point(4,3), Point(1,3))
shear = Matrix([[1, 1], [0, 1]])
R_sh = [Point(shear * Matrix(p)) for p in R.vertices]
def plot_points(points, label):
    x = [p.x for p in points] + [points[0].x]
    y = [p.y for p in points] + [points[0].y]
    plt.plot(x, y, label=label)
plot_points(R.vertices, "Original Rectangle")
plot_points(R_sh, "Sheared Rectangle")
plt.legend()
plt.grid()
plt.title("Shearing Transformation")
plt.show()
```



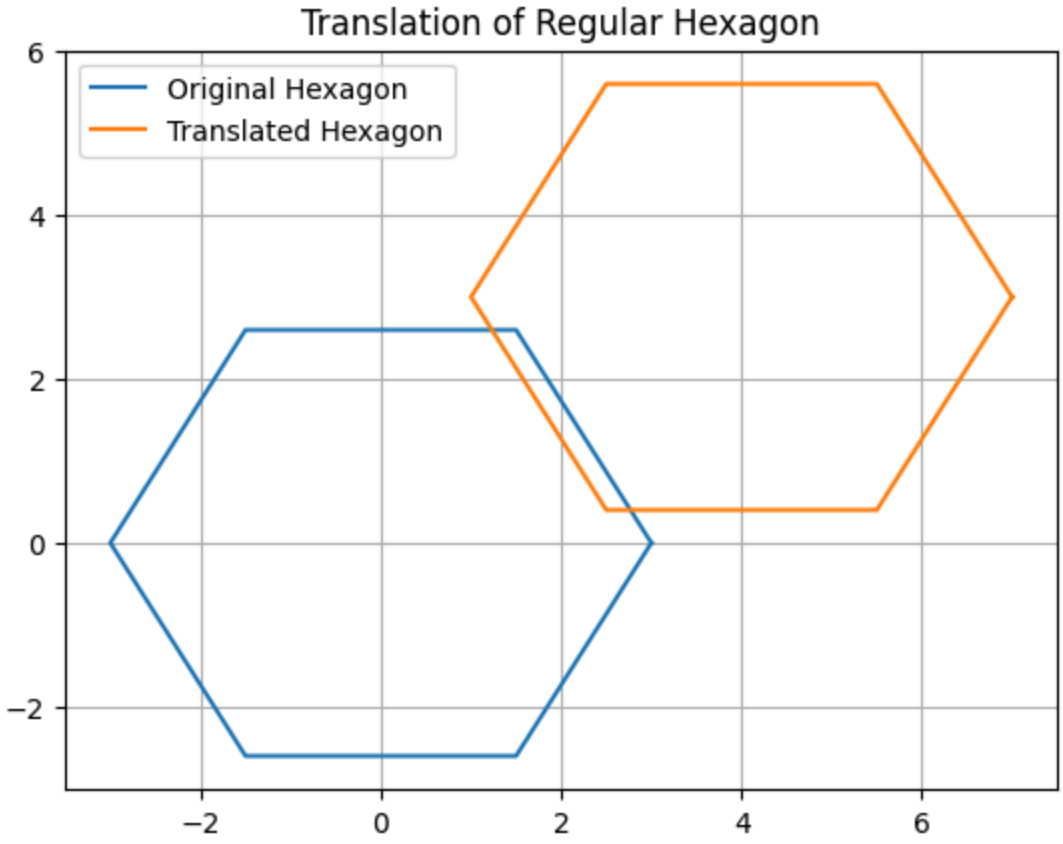
Q5. Plot a polygon and its image after rotation by 45 degrees.

In [6]:
```
import matplotlib.pyplot as plt
from sympy import Polygon, Point, pi
P = Polygon(Point(2,1), Point(4,1), Point(5,3), Point(3,4))
P_r = P.rotate(pi/4)
def draw(poly, label):
    x = [p.x for p in poly.vertices] + [poly.vertices[0].x]
    y = [p.y for p in poly.vertices] + [poly.vertices[0].y]
    plt.plot(x, y, label=label)
draw(P, "Original Polygon")
draw(P_r, "Rotated Polygon")
plt.legend()
plt.grid()
plt.title("Rotation of Polygon")
plt.show()
```



Q6. Plot a regular hexagon and its translated image.

In [7]:
```
import matplotlib.pyplot as plt
from sympy import RegularPolygon, Point
H = RegularPolygon(Point(0,0), 3, 6)
H_t = H.translate(4, 3)
def plot_hex(hexagon, label):
    x = [v.x for v in hexagon.vertices] + [hexagon.vertices[0].x]
    y = [v.y for v in hexagon.vertices] + [hexagon.vertices[0].y]
    plt.plot(x, y, label=label)
plot_hex(H, "Original Hexagon")
plot_hex(H_t, "Translated Hexagon")
plt.legend()
plt.grid()
plt.title("Translation of Regular Hexagon")
plt.show()
```


Translation of Regular Hexagon
Original Hexagon
Translated Hexagon
6
4
2
0
−2
−2
0
2
4
6