**Practical No. 9**

***Aim:-*** Write Program for **2-D Transformations→ Rotation.**

***Relevant Course Outcome(s):-***

Perform and demonstrate basic and composite graphical transformations on given object.

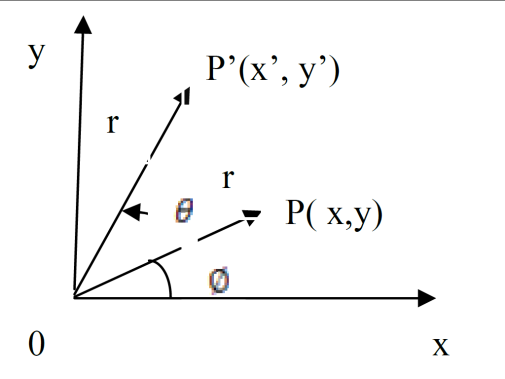
***Resources Required (Hardware & Softwares):-***

1. A Desktop PC/ Laptop
2. Ansi C/ Turbo C/ (Any distribution) installed

***Theory:-***

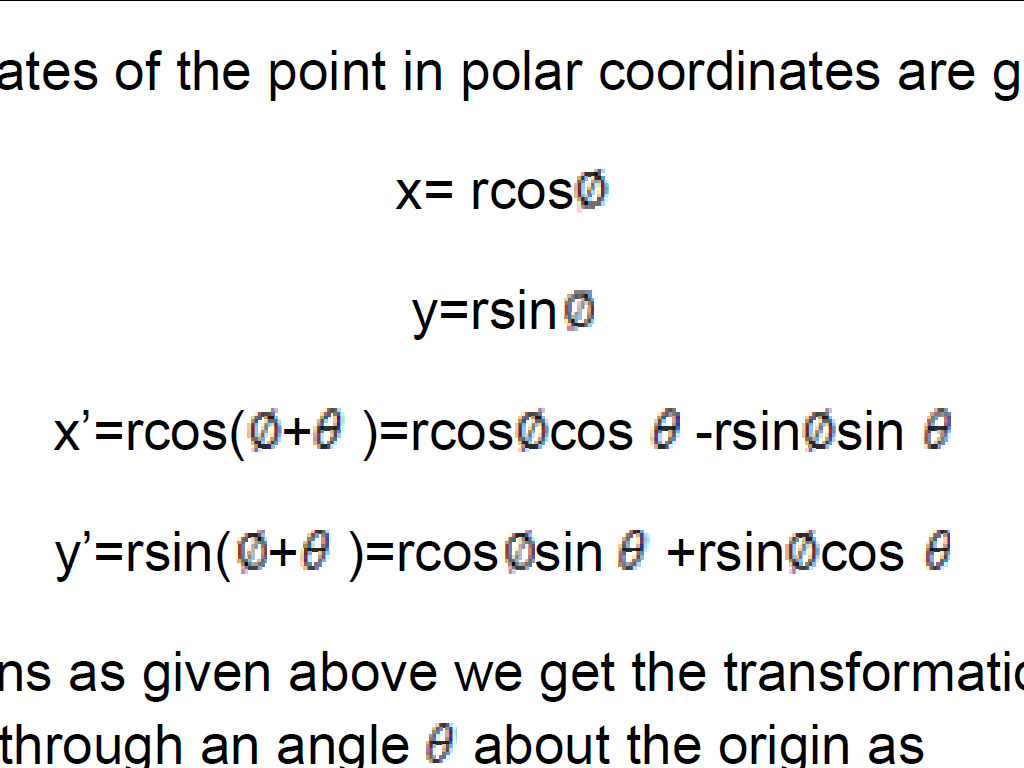
***Rotation***

A two dimensional rotation is applied to an object by repositioning it along a circular path in the xy plane. To generate a rotation, we specify a rotation angle and the position of the rotation point about which the object is to be rotated.

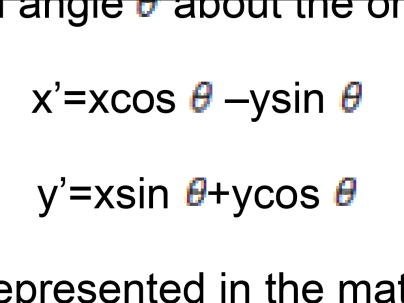


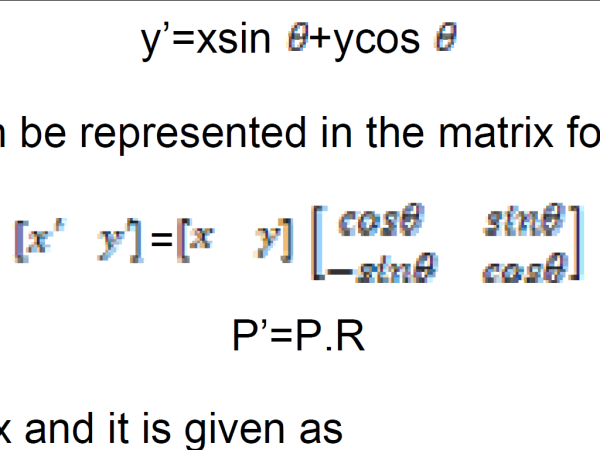
Let us consider the rotation of the object about the origin as shown above.

Here r is the constant distance of the point from the origin, angle is the original angular position of the point from the horizontal and is the rotation angle. Using standard trigonometric equations, we can express the transformed coordinates in terms of and as

The original coordinates of the point in polar coordinates are given as 

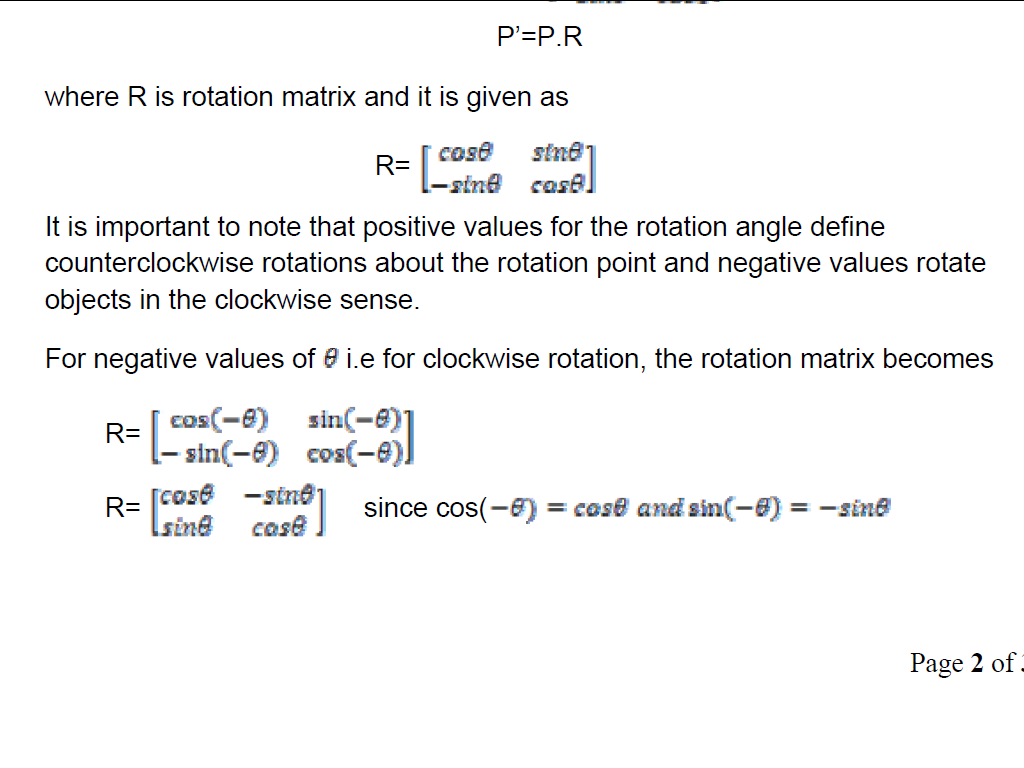
Substituting equations as given above we get the transformation equations for rotating a point(x,y) through an angle about the origin as



The above equations can be represented in the matrix form as given below

It is important to note that positive values for the rotation angle define counterclockwise rotations about the rotation point and negative values rotate objects in the clockwise sense.

For negative values of i.e for clockwise rotation, the rotation matrix becomes



***Algorithm:-***

1. **Rotation:-**

1. Read the coordinates of the point to be rotated about origin.

2. Read the angle of rotation.

3. Plot the line between origin and coordinates of the point.

4. Multiply the original coordinates with Rotation Matrix to obtain transformed coordinates.

5. Plot the line between origin and transformed coordinates

6. Stop.

**Program For Rotation**

#include <graphics.h>

#include <math.h>

#include <conio.h>

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int x1 = 100, y1 = 100, x2 = 200, y2 = 100;

double angle = 45 \* (M\_PI / 180);

int xr = 150, yr = 150;

line(x1, y1, x2, y2);

int x1\_new = xr + (x1 - xr) \* cos(angle) - (y1 - yr) \* sin(angle);

int y1\_new = yr + (x1 - xr) \* sin(angle) + (y1 - yr) \* cos(angle);

int x2\_new = xr + (x2 - xr) \* cos(angle) - (y2 - yr) \* sin(angle);

int y2\_new = yr + (x2 - xr) \* sin(angle) + (y2 - yr) \* cos(angle);

setcolor(RED);

line(x1\_new, y1\_new, x2\_new, y2\_new);

getch();

closegraph();

return 0;

}

***Output:-* ( Paste your own Output )**

***Conclusion:-***

Thus, we have written & implemented Program for **2-D Transformations→** **Rotation.**