```
/*NAME:-
ROLL_NO:-
DIV:-
Program No:-04A
AIM:-Write C++/Java program to draw 2-D object and perform following basic transformations,
a) Scaling
b) Translation
c) Rotation
Use operator overloading.*/


#include<iostream>
#include<graphics.h>
#include<math.h>
using namespace std;

int pi[10];                         //total points i.e x & y
double b[3][3]={1,0,0,0,1,0,0,0,1};
int c[1][1];
float a[1][1];

void matmul(float[]);

int main()
{
        int i,x,y,tx,ty,sx,sy,angle=10,xmax,ymax,xmid,ymid,op;
        int gm,gd=DETECT;
        float p1[10]={50,50,
        100,50,
        100,100,
        50,100,
        50,50};

        cout<<"\nSelect transformation:";
        cout<<"\n1.Translation";
        cout<<"\n2.Rotation:";
        cout<<"\n3.Scaling:";
        cout<<"\nEnter the option:";
        cin>>op;

        switch(op)
        {
                case 1:
```

```
        cout<<"\nEnter x translation:";
        cin>>tx;
        cout<<"\nEnter y translation:";
        cin>>ty;

        b[0][0]=1;
        b[0][1]=0;
        b[0][2]=0;

        b[1][0]=0;
        b[1][1]=1;
        b[1][2]=0;

        b[2][0]=tx;
        b[2][1]=ty;
        b[2][2]=1;

        break;

case 2:
        cout<<"\nEnter rotation angle:";
        cin>>angle;

        b[0][0]=cos(angle*3.14/180);
        b[0][1]=sin(angle*3.14/180);
        b[0][2]=0;

        b[1][0]=-sin(angle*3.14/180);
        b[1][1]=cos(angle*3.14/180);
        b[1][2]=0;

        b[2][0]=0;
        b[2][1]=0;
        b[2][2]=1;

        break;

case 3:
        cout<<"\nEnter x scaling:";
        cin>>sx;
        cout<<"\nEnter y scaling:";
        cin>>sy;
```

```
                    b[0][0]=sx;
                    b[0][1]=0;
                    b[0][2]=0;

                    b[1][0]=0;
                    b[1][1]=sy;
                    b[1][2]=0;

                    b[2][0]=0;
                    b[2][1]=0;
                    b[2][2]=1;

                    break;
        }

initgraph(&gd,&gm,NULL);

xmax=getmaxx();
ymax=getmaxy();
xmid=xmax/2;
ymid=ymax/2;

setcolor(1);
line(xmid,0,xmid,ymax);
line(0,ymid,xmax,ymid);
setcolor(4);

for(i=0;i<8;i=i+2)
{
        line(p1[i]+xmid,ymid-p1[i+1],xmid+p1[i+2],ymid-p1[i+3]);
}

matmul(p1);
setcolor(15);

for(i=0;i<8;i=i+2)
{
        line(xmid+pi[i],ymid-pi[i+1],xmid+pi[i+2],ymid-pi[i+3]);
}

getch();
closegraph();
return 0;
```

```
}

void matmul(float p[10])
{
        int i;
        for(i=0;i<9;i=i+2)
        {
                a[0][0]=p[i];
                a[0][1]=p[i+1];
                c[0][0]=a[0][0]*b[0][0]+a[0][1]*b[1][0]+b[2][0];
                c[0][1]=a[0][0]*b[0][1]+a[0][1]*b[1][1]+b[2][1];
                pi[i]=c[0][0];
                pi[i+1]=c[0][1];

        }
}

/*
------------------------OUTPUT---------------------
Select transformation:
1.Translation
2.Rotation:
3.Scaling:
Enter the option:1

Enter x translation:10

Enter y translation:10
root1@root1-ThinkCentre-M60e:~$ ./a.out

Select transformation:
1.Translation
2.Rotation:
3.Scaling:
Enter the option:2

Enter rotation angle:45

root1@root1-ThinkCentre-M60e:~$ ./a.out

Select transformation:
1.Translation
2.Rotation:
```

```
3.Scaling:
Enter the option:3

Enter x scaling:2

Enter y scaling:2

*/
```