----------------------------------------------------------------------Write C++/Java program to draw 2-D object and perform following basic transformations,

1. Scaling
2. Translation
3. Rotation.

Use operator overloading

----------------------------------------------------------------------#include "transform.h"

#include "ui\_transform.h"

#include <QPainter>

#include<math.h>

Transform::Transform(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::Transform)

{

ui->setupUi(this);

i=0;

}

Transform::~Transform()

{

delete ui;

}

void Transform :: paintEvent(QPaintEvent \*event)

{

V=ui->vert->toPlainText().toInt();

if (flag==1)

draw(coor);

if (flag==2)

{

int l,m;

for(l=0;l<3;l++)

for(m=0;m<3;m++)

scal\_mat[l][m]=0;

Sx=ui->sx->toPlainText().toInt();

Sy=ui->sy->toPlainText().toInt();

scal\_mat[0][0]=Sx;

scal\_mat[1][1]=Sy;

scal\_mat[2][2]=1;

scale();

}

if (flag==3)

{

int l,m;

for(l=0;l<3;l++)

for(m=0;m<3;m++)

scal\_mat[l][m]=0;

Tx=ui->tx->toPlainText().toInt();

Ty=ui->ty->toPlainText().toInt();

scal\_mat[0][0]=1;

scal\_mat[1][1]=1;

scal\_mat[2][0]=Tx;

scal\_mat[2][1]=Ty;

scal\_mat[2][2]=1;

translate();

}

if(flag==4)

{

int l,m;

for(l=0;l<3;l++)

for(m=0;m<3;m++)

rot[l][m]=0;

angle=ui->theta->toPlainText().toInt();

rot[0][0]=cos(3.14\*angle/180);

rot[1][1]=cos(3.14\*angle/180);

rot[0][1]=-sin(3.14\*angle/180);

rot[2][1]=sin(3.14\*angle/180);

rot[2][2]=1;

rotate();

}

if (flag==5)

{ i=0;

ui->Next->setVisible(true);

}

}

void Transform:: draw(float abc[][3])

{

QPainter p(this);

abc[V][0]=abc[0][0];

abc[V][1]=abc[0][1];

int j;

for(j=0;j<V;j++)

p.drawLine(abc[j][0],abc[j][1],abc[j+1][0],abc[j+1][1]);

}

void Transform::scale()

{

int l,m,k;

for(l=0;l<V;l++)

for(m=0;m<3;m++)

{

coor2[l][m]=0;

for(k=0;k<3;k++)

{

coor2[l][m]=coor2[l][m]+coor[l][k]\*scal\_mat[k][m];

}

}

draw(coor2);

}

void Transform::translate()

{

int l,m,k;

for(l=0;l<V;l++)

for(m=0;m<3;m++)

{

coor2[l][m]=0;

for(k=0;k<3;k++)

{

coor2[l][m]=coor2[l][m]+coor[l][k]\*scal\_mat[k][m];

}

}

draw(coor2);

}

void Transform::rotate()

{

int l,m,k;

for(l=0;l<V;l++)

for(m=0;m<3;m++)

{

coor2[l][m]=0;

for(k=0;k<3;k++)

{

coor2[l][m]=coor2[l][m]+(float)coor[l][k]\*rot[k][m];

}

}

for(int j=0;j<V;j++)

coor2[j][0]-=coor2[j][1];

draw(coor2);

}

void Transform::on\_Next\_clicked()

{

if(i==(V-1))

ui->Next->setVisible(false);

if(i<V)

{

coor[i][0]=ui->x->toPlainText().toInt();

coor[i][1]=ui->y->toPlainText().toInt();

coor[i][2]=1;

i++;

}

ui->x->QTextEdit::clear();

ui->y->QTextEdit::clear();

}

void Transform::on\_Draw\_clicked()

{

flag=1;

update();

}

void Transform::on\_reset\_clicked()

{

i=0;

ui->Next->setVisible(true);

}

void Transform::on\_Scale\_clicked()

{

flag=2;

update();

}

void Transform::on\_translate\_clicked()

{

flag=3;

update();

}

void Transform::on\_Rotate\_clicked()

{

flag=4;

update();

}

**OUTPUT:**

For polygon = (1,-1)(3,-1)(3,-5)

after translation by Ty = -2

final polygon = (4,1)(3,4)(3,0)

