**A**

**LAB REPORT**

**ON**

**COMPUTER GRAPHICS**

**By**

**RASMI JATI**

**Exam Roll No: 7930/17**



**Submitted to:**

**Indra PC**

**Kantipur College of Management and Information Technology**

In Partial Fulfillment of the Requirements for the Course Computer Graphics

Mid Baneshwor, Kathmandu

March, 2020

Table of Contents

[1 Write a program that uses Digital Differential Analyzer algorithm to draw line. 3](#_Toc64387332)

[1.1 Source Code: 3](#_Toc64387333)

[1.2 Output: 6](#_Toc64387334)

[2 Write a program that uses Bresenham Line Drawing algorithm to draw a line. 7](#_Toc64387335)

[2.1 Source Code: 7](#_Toc64387336)

[2.2 Output: 12](#_Toc64387337)

[3 Write a program to draw circle. 13](#_Toc64387338)

[3.1 Source Code: 13](#_Toc64387339)

[3.2 Output: 15](#_Toc64387340)

[4 Write a program to scale a line about origin. 16](#_Toc64387341)

[4.1 Source Code: 16](#_Toc64387342)

[4.2 Output: 22](#_Toc64387343)

[5 Write a program to scale a line about arbitrary point. 23](#_Toc64387344)

[5.1 Source Code: 23](#_Toc64387345)

[5.2 Output: 29](#_Toc64387346)

[6 Write a program to translate a line. 30](#_Toc64387347)

[6.1 Source Code: 30](#_Toc64387348)

[6.2 Output: 35](#_Toc64387349)

[7 Write a program to rotate a triangle about arbitrary point. 36](#_Toc64387350)

[7.1 Source Code: 36](#_Toc64387351)

[7.2 Output: 41](#_Toc64387352)

[8 Write a program to shear a rectangle through x- direction. 42](#_Toc64387353)

[8.1 Source Code: 42](#_Toc64387354)

[8.2 Output: 45](#_Toc64387355)

# Write a program that uses Digital Differential Analyzer algorithm to draw line.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class DDA extends JFrame implements ActionListener{

JLabel lblx1,lblx2,lbly1,lbly2,lbl;

JTextField tfx1,tfx2,tfy1,tfy2;

JButton btnDraw;

public DDA(){

lbl = new JLabel("Digital Differential Analyzer");

lblx1 = new JLabel("Enter x1");

lblx2 = new JLabel("Enter x2");

lbly1 = new JLabel("Enter y1");

lbly2 = new JLabel("Enter y2");

tfx1 = new JTextField(80);

tfx2 = new JTextField(80);

tfy1 = new JTextField(80);

tfy2 = new JTextField(80);

btnDraw = new JButton("Draw");

btnDraw.setBounds(152,150,100,30);

tfy2.setBounds(293,90,100,30);

tfy1.setBounds(93,90,100,30);

tfx2.setBounds(293,40,100,30);

tfx1.setBounds(93,40,100,30);

lbly2.setBounds(210,90,80,30);

lbly1.setBounds(10,90,80,30);

lblx2.setBounds(210,40,80,30);

lbl.setBounds(105,5,200,25);

lblx1.setBounds(10,40,80,30);

add(lbl); add(lblx1); add(tfx1); add(lblx2); add(tfx2);

add(lbly1); add(tfy1); add(lbly2); add(tfy2); add(btnDraw);

btnDraw.addActionListener(this);

setLayout(null); setSize(500,400);

setTitle("Rasmi Jati"); setVisible(true);

}

public void actionPerformed(ActionEvent e){

new DDAClass(tfx1.getText(),tfy1.getText(),tfx2.getText(),tfy2.getText());

}

public static void main(String[] args) {

new DDA();

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

class DDAClass extends JFrame{

double x1,x2,y2,y1,x,y,steps,i,dx,dy;

public DDAClass(String a1, String b1, String a2, String b2){

JPanel p = new JPanel();

getContentPane().add(p);

x1 = Double.parseDouble(a1); y1 = Double.parseDouble(b1);

x2 = Double.parseDouble(a2); y2 = Double.parseDouble(b2);

setLayout(null); setSize(800,500);

setTitle("Rasmi Jati"); setVisible(true);

}

public void paint(Graphics g){

super.paint(g);

Graphics2D g2 = (Graphics2D) g;

if(x1<x2){

x= x1; y = y1;

}else{

x = x2; y = y2;

}

dx = x2-x1; dy = y2-y1;

if(Math.abs(dx)>Math.abs(dy)){

steps = Math.abs(dx);

}else{

steps = Math.abs(dy);

}

double xIncreament = dx/steps;

double yIncreament = dy/steps;

for(i=0;i<steps;i++){

g2.draw(new Ellipse2D.Double(x,y,1,1));

if(x1<x2){

x = x+xIncreament;

}else{

x = x-xIncreament;

}

if(y1<y2){

y = y+yIncreament;

}else{

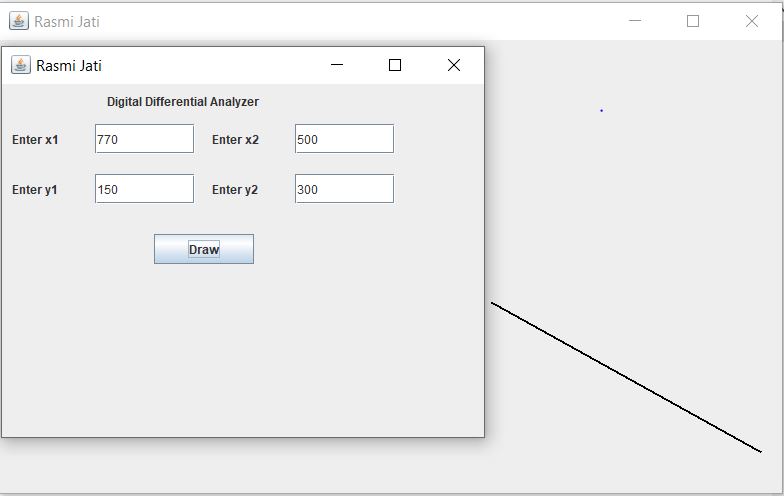
y = y-yIncreament;

}}

}

}

## Output:



# Write a program that uses Bresenham Line Drawing algorithm to draw a line.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class DrawingBresenhamLine extends JFrame implements ActionListener{

JLabel lblx1,lblx2,lbly1,lbly2,lbl;

JTextField tx1,tx2,ty1,ty2;

JButton btnDraw;

public DrawingBresenhamLine(){

lbl = new JLabel("Bresenham Line Drawing");

lblx1 = new JLabel("Enter x1");

lblx2 = new JLabel("Enter x2");

lbly1 = new JLabel("Enter y1");

lbly2 = new JLabel("Enter y2");

tx1 = new JTextField(80);

tx2 = new JTextField(80);

ty1 = new JTextField(80);

ty2 = new JTextField(80);

btnDraw = new JButton("Draw");

lbl.setBounds(105,5,150,25);

lblx1.setBounds(10,40,80,30);

lblx2.setBounds(210,40,80,30);

lbly1.setBounds(10,90,80,30);

lbly2.setBounds(210,90,80,30);

tx1.setBounds(93,40,100,30);

tx2.setBounds(293,40,100,30);

ty1.setBounds(93,90,100,30);

ty2.setBounds(293,90,100,30);

btnDraw.setBounds(152,150,100,30);

add(lbl);add(lblx1); add(tx1);add(lblx2); add(tx2);

add(lbly1); add(ty1);add(lbly2);add(ty2);add(btnDraw);

btnDraw.addActionListener(this);

setLayout(null);setSize(500,400);

setTitle("Rasmi Jati");setVisible(true);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e){

new BLD(tx1.getText(),ty1.getText(),tx2.getText(),ty2.getText());

}

public static void main(String[] args){

new DrawingBresenhamLine();

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

public class BLD extends JFrame{

double x1,x2,y1,y2,dx,dy,p;

public BLD(String a1, String b1, String a2, String b2){

JPanel p = new JPanel();

getContentPane().add(p);

x1 = Double.parseDouble(a1);

y1 = Double.parseDouble(b1);

x2 = Double.parseDouble(a2);

y2 = Double.parseDouble(b2);

setLayout(null);

setSize(800,500);

setTitle("Rasmi Jati");

setVisible(true);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void paint(Graphics g){

super.paint(g);

Graphics2D g2 = (Graphics2D) g;

dx=Math.abs(x2-x1);

dy=Math.abs(y2-y1);

if (dx>dy){

if((x2-x1)>=0){

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g2.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1++;

p+=2\*dy;

}else{

if((y2-y1)>=0){

x1++; y1++;

p+=(2\*dy)-(2\*dx);

}else{

x1++; y1--;

p+=(2\*dy)-(2\*dx);

}

}

}

}else{

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g2.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1--;

p+=2\*dy;

}else{

if((y2-y1)>=0){

x1--; y1++;

p+=(2\*dy)-(2\*dx);

}else{

x1--; y1--;

p+=(2\*dy)-(2\*dx);

}

}

}

}

}else{

if((y2-y1)>0){

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g2.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1++;

p+=2\*dx;

}else{

if((x2-x1)>=0){

y1++; x1++;

p+=(2\*dx)-(2\*dy);

}else{

y1++; x1--;

p+=(2\*dx)-(2\*dy);

}

}

}

}else{

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g2.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1--;

p+=2\*dx;

}else{

if((x2-x1)>=0){

y1--; x1++;

p+=(2\*dx)-(2\*dy);

}else{

y1--; x1--;

p+=(2\*dx)-(2\*dy);

}

}

}

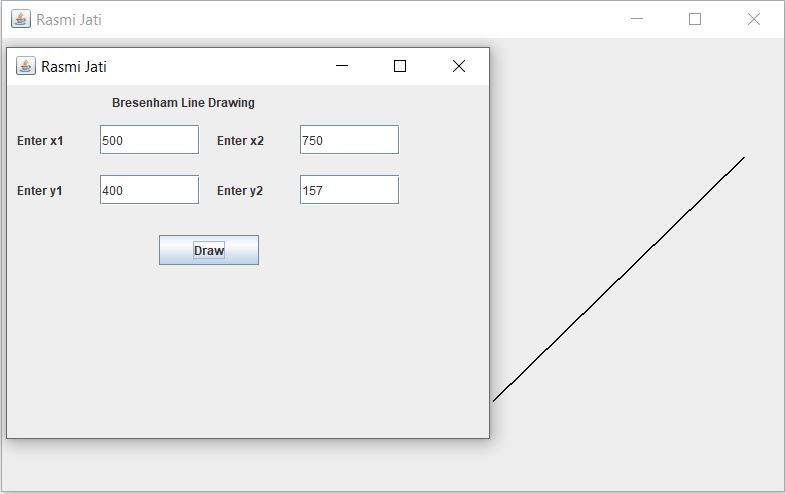
}

}

}

}

## Output:



# Write a program to draw circle.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class DrawingCircle extends JFrame implements ActionListener{

JLabel lblMessage, lblRadius, lblx, lbly;

JTextField tfr,tfx,tfy;

JButton btnDraw;

public DrawingCircle(){

lblMessage = new JLabel("Enter x,y and radius");

lblx = new JLabel("X");lbly = new JLabel("Y");

lblRadius = new JLabel("Radius");

tfx = new JTextField(80);tfy = new JTextField(80);

tfr = new JTextField(80);

btnDraw = new JButton("Draw");

lblMessage.setBounds(105,5,150,25);

lblx.setBounds(10,40,80,30);

lbly.setBounds(10,90,80,30);

lblRadius.setBounds(10,140,80,30);

tfx.setBounds(93,40,100,30);

tfy.setBounds(93,90,100,30);

tfr.setBounds(93,140,100,30);

btnDraw.setBounds(85,200,100,30);

add(lblMessage); add(lblx); add(tfx); add(lbly);

add(tfy); add(lblRadius); add(tfr); add(btnDraw);

btnDraw.addActionListener(this);

setLayout(null); setSize(400,300);

setTitle("Rasmi Jati"); setVisible(true);

}

public void actionPerformed(ActionEvent e){

new CircleClass(tfx.getText(),tfy.getText(),tfr.getText());

}

public static void main(String[] args) {

new DrawingCircle();

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

public class CircleClass extends JFrame{

int x,y,x1,y1,r,p;

public CircleClass(String x0, String y0, String r0){

JPanel p = new JPanel();

getContentPane().add(p);

x = Integer.parseInt(x0);y = Integer.parseInt(y0);

r = Integer.parseInt(r0);

x1 = 0; y1 = r;

setSize(800,500); setVisible(true); setTitle("Circle");

}

public void paint(Graphics g){

super.paint(g);

Graphics2D g2=(Graphics2D) g;

p = 1-r;

while(x1<y1){

g2.draw(new Ellipse2D.Double(x1+x,y1+y,1,1));

g2.draw(new Ellipse2D.Double(x1+x,-y1+y,1,1));

g2.draw(new Ellipse2D.Double(-x1+x,y1+y,1,1));

g2.draw(new Ellipse2D.Double(-x1+x,-y1+y,1,1));

g2.draw(new Ellipse2D.Double(y1+x,x1+y,1,1));

g2.draw(new Ellipse2D.Double(y1+x,-x1+y,1,1));

g2.draw(new Ellipse2D.Double(-y1+x,x1+y,1,1));

g2.draw(new Ellipse2D.Double(-y1+x,-x1+y,1,1));

if(p<0){

p+=(2\*x1+2)+1;

x1++;

}

else{

p+=(2\*x1+2)-(2\*y1-2)+1;

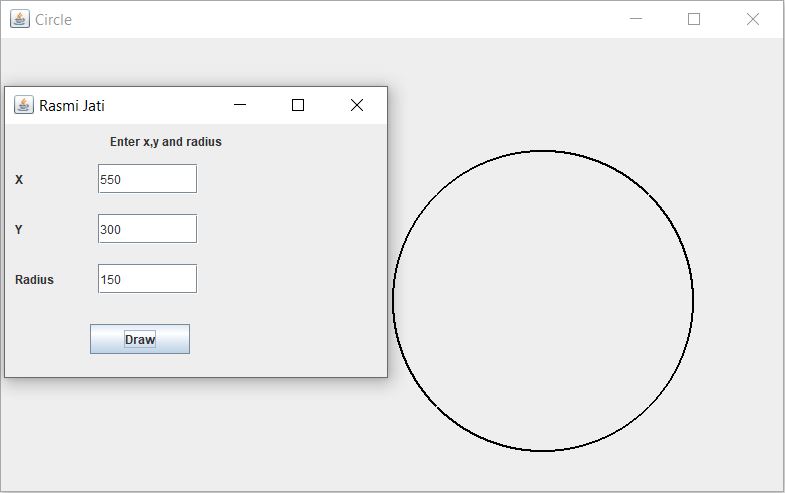
x1++;y1--;

}

}

}}

## Output:



# Write a program to scale a line about origin.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class ScaleOrigin extends JFrame implements ActionListener{

JLabel lblx1,lblx2,lbly1,lbly2,lblx3,lbly3,lblsx,lblsy,lbl;

JTextField tfx1,tfx2,tfy2,tfy1,tfx3,tfy3,tfsx,tfsy;

JButton btnScale;

public ScaleOrigin(){

lbl = new JLabel("Scaling a triangle About Origin");

lblx1 = new JLabel("Enter x1");

lblx2 = new JLabel("Enter x2");

lblx3 = new JLabel("Enter x3");

lbly1 = new JLabel("Enter y1");

lbly2 = new JLabel("Enter y2");

lbly3 = new JLabel("Enter y3");

lblsx = new JLabel("Enter sx");

lblsy = new JLabel("Enter sy");

btnScale = new JButton("Scale");

tfx1 = new JTextField(80); tfx2 = new JTextField(80);

tfx3 = new JTextField(80); tfy1 = new JTextField(80);

tfy2 = new JTextField(80); tfy3 = new JTextField(80);

tfsx = new JTextField(80); tfsy = new JTextField(80);

lbl.setBounds(105,5,200,25);

lblx1.setBounds(10,40,80,30);

lbly1.setBounds(210,40,80,30);

lblx2.setBounds(10,90,80,30);

lbly2.setBounds(210,90,80,30);

lblx3.setBounds(10,140,80,30);

lbly3.setBounds(210,140,80,30);

tfx1.setBounds(93,40,100,30);

tfy1.setBounds(293,40,100,30);

tfx2.setBounds(93,90,100,30);

tfy2.setBounds(293,90,100,30);

tfx3.setBounds(93,140,100,30);

tfy3.setBounds(293,140,100,30);

lblsx.setBounds(10,190,80,30);

lblsy.setBounds(210,190,80,30);

tfsx.setBounds(93,190,100,30);

tfsy.setBounds(293,190,100,30);

btnScale.setBounds(150,220,100,30);

add(lbl); add(lblx1); add(tfx1); add(lblx2); add(tfx2);

add(lblx3); add(tfx3); add(lbly1); add(tfy1); add(lbly2);

add(tfy2); add(lbly3); add(tfy3); add(lblsx); add(tfsx);

add(lblsy); add(tfsy); add(btnScale);

btnScale.addActionListener(this);

setLayout(null); setSize(500,400); setVisible(true);

}

public void actionPerformed(ActionEvent e){

new ScaleOriginClass(tfx1.getText(),tfy1.getText(),tfx2.getText(),tfy2.getText(),tfx3.getText(),tfy3.getText(),tfsx.getText(),tfsy.getText());

}

public static void main(String[] args) {

new ScaleOrigin();

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

class ScaleOriginClass extends JFrame{

double sx,sy,prod = 0,xIncreament,yIncreament;

double x,y,steps,i,dx,dy,p,x1,x2,y2,y1,y3,x3;

public ScaleOriginClass(String a1, String b1, String a2, String b2, String a3, String b3,String sa, String sb){

JPanel p = new JPanel();

getContentPane().add(p);

x1 = Double.parseDouble(a1); y1 = Double.parseDouble(b1);

x2 = Double.parseDouble(a2); y2 = Double.parseDouble(b2);

x3 = Double.parseDouble(a3); y3 = Double.parseDouble(b3);

sx = Double.parseDouble(sa); sy = Double.parseDouble(sb);

setSize(900,600); setTitle("Rasmi Jati"); setVisible(true);

}

public void sketch(double x1,double y1,double x2,double y2,Graphics2D g){

dx = Math.abs(x2-x1); dy = Math.abs(y2-y1);

if (dx>dy){

if((x2-x1)>=0){

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1++; p+=2\*dy;

}else{

if((y2-y1)>=0){

x1++; y1++; p+=(2\*dy)-(2\*dx);

}else{

x1++; y1--; p+=(2\*dy)-(2\*dx);

}

}

}

}else{

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1--; p+=2\*dy;

}else{

if((y2-y1)>=0){

x1--; y1++; p+=(2\*dy)-(2\*dx);

}else{

x1--; y1--; p+=(2\*dy)-(2\*dx);

}

}

}

}

}else{

if((y2-y1)>0){

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1++; p+=2\*dx;

}else{

if((x2-x1)>=0){

y1++; x1++; p+=(2\*dx)-(2\*dy);

}else{

y1++; x1--; p+=(2\*dx)-(2\*dy);

}

}

}

}else{

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1--; p+=2\*dx;

}else{

if((x2-x1)>=0){

y1--; x1++; p+=(2\*dx)-(2\*dy);

}else{

y1--; x1--; p+=(2\*dx)-(2\*dy);

}

}

}

}

}

}

public void paint(Graphics g){

super.paint(g);

Graphics2D g2 = (Graphics2D) g;

sketch(x1,y1,x2,y2,g2);

sketch(x1,y1,x3,y3,g2);

sketch(x2,y2,x3,y3,g2);

double[][] scale = {{sx,0,0},{0,sy,0},{0,0,1}};

double[][] coor = {{x1,x2,x3},{y1,y2,y3},{1,1,1}};

double[][] mat = new double[3][3] ;

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

for(int j=0;j<3;j++) {

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

prod = prod+scale[l][k]\*coor[k][j];

}

mat [l][j]= prod;

prod = 0; }

}

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

for(int j=0;j<3;j++){

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

if(l==0){

x1 = mat[0][0];

x2 = mat[0][1];

x3 = mat[0][2];

}

else if(l==1){

y1 = mat[1][0];

y2 = mat[1][1];

y3 = mat[1][2];

}

}

}

}

sketch(x1,y1,x2,y2,g2);

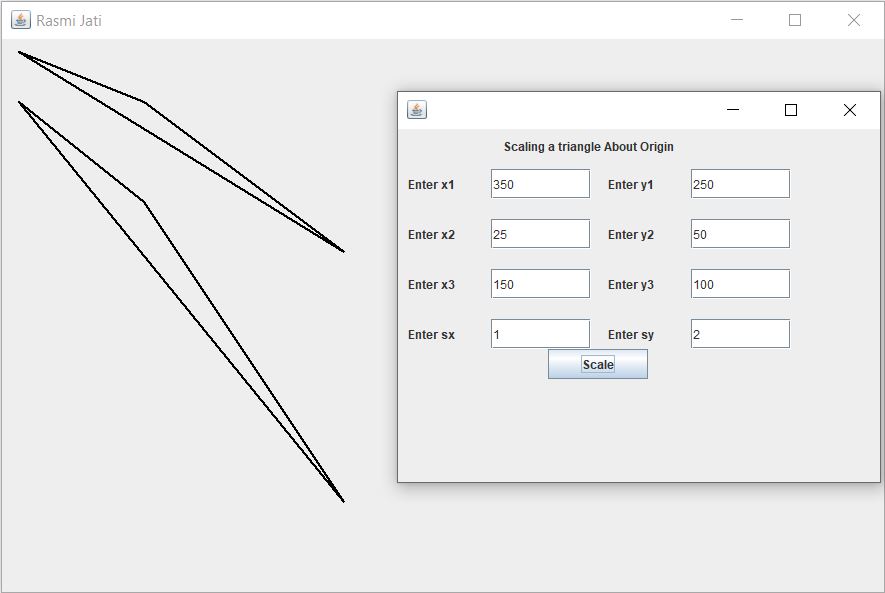
sketch(x1,y1,x3,y3,g2);

sketch(x2,y2,x3,y3,g2);

}

}

## Output:



# Write a program to scale a line about arbitrary point.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class ScaleArbitary extends JFrame implements ActionListener{

JLabel lblx1,lblx2,lbly1,lbly2,lblx3;

JLabel lbly3,lblsx,lblsy,lbl,lblxc,lblyc;

JTextField tfx1,tfx2,tfy2,tfy1,tfx3,tfy3,tfsx,tfsy,tfxc,tfyc;

JButton btnScale;

public ScaleArbitary(){

lbl = new JLabel("Scaling a triangle About Arbitary");

lblx1 = new JLabel("Enter x1");

lblx2 = new JLabel("Enter x2");

lblx3 = new JLabel("Enter x3");

lbly1 = new JLabel("Enter y1");

lbly2 = new JLabel("Enter y2");

lbly3 = new JLabel("Enter y3");

lblsx = new JLabel("Enter sx");

lblsy = new JLabel("Enter sy");

lblxc = new JLabel("Enter xc");

lblyc = new JLabel("Enter yc");

btnScale = new JButton("Scale");

tfx1 = new JTextField(80); tfx2 = new JTextField(80);

tfx3 = new JTextField(80); tfy1 = new JTextField(80);

tfy2 = new JTextField(80); tfy3 = new JTextField(80);

tfsx = new JTextField(80); tfsy = new JTextField(80);

tfxc = new JTextField(80); tfyc = new JTextField(80);

lbl.setBounds(105,5,200,25);

lblx1.setBounds(10,40,80,30);

lbly1.setBounds(210,40,80,30);

lblx2.setBounds(10,90,80,30);

lbly2.setBounds(210,90,80,30);

lblx3.setBounds(10,140,80,30);

lbly3.setBounds(210,140,80,30);

lblsx.setBounds(10,190,80,30);

lblsy.setBounds(210,190,80,30);

lblxc.setBounds(10,240,80,30);

lblyc.setBounds(210,240,80,30);

tfx1.setBounds(93,40,100,30);

tfy1.setBounds(293,40,100,30);

tfx2.setBounds(93,90,100,30);

tfy2.setBounds(293,90,100,30);

tfx3.setBounds(93,140,100,30);

tfy3.setBounds(293,140,100,30);

tfsx.setBounds(93,190,100,30);

tfsy.setBounds(293,190,100,30);

tfxc.setBounds(93,240,100,30);

tfyc.setBounds(293,240,100,30);

btnScale.setBounds(150,300,100,30);

add(lbl); add(lblx1); add(tfx1); add(lblx2); add(tfx2);

add(lblx3); add(tfx3); add(lbly1); add(tfy1); add(lbly2);

add(tfy2); add(lbly3); add(tfy3); add(lblsx); add(tfsx);

add(lblsy); add(tfsy); add(lblxc); add(tfxc); add(lblyc);

add(tfyc); add(btnScale); btnScale.addActionListener(this);

setLayout(null); setSize(500,400); setVisible(true);

}

public void actionPerformed(ActionEvent e){

new ScaleArbitaryClass(tfx1.getText(), tfy1.getText(), tfx2.getText(),tfy2.getText(),tfx3.getText(),tfy3.getText(),tfsx.getText(),tfsy.getText(),tfxc.getText(),tfyc.getText());

}

public static void main(String[] args) {

new ScaleArbitary();

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

class ScaleArbitaryClass extends JFrame{

double xc,yc,sx,sy,prod = 0,xIncreament,yIncreament;

double x,y,steps,i,dx,dy,p,x1,x2,y2,y1,y3,x3;

public ScaleArbitaryClass(String a1, String b1, String a2, String b2, String a3, String b3,String sa, String sb,String ac, String bc){

JPanel p = new JPanel();

getContentPane().add(p);

x1 = Double.parseDouble(a1); y1 = Double.parseDouble(b1);

x2 = Double.parseDouble(a2); y2 = Double.parseDouble(b2);

x3 = Double.parseDouble(a3); y3 = Double.parseDouble(b3);

sx = Double.parseDouble(sa); sy = Double.parseDouble(sb);

xc = Double.parseDouble(ac); yc = Double.parseDouble(bc);

setSize(900,600); setTitle("Rasmi Jati"); setVisible(true);

}

public void sketch(double x1,double y1,double x2,double y2,Graphics2D g){

dx = Math.abs(x2-x1); dy = Math.abs(y2-y1);

if (dx>dy){

if((x2-x1)>=0){

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1++; p+=2\*dy;

}else{

if((y2-y1)>=0){

x1++; y1++; p+=(2\*dy)-(2\*dx);

}else{

x1++; y1--; p+=(2\*dy)-(2\*dx);

}

}

}

}else{

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1--; p+=2\*dy;

}else{

if((y2-y1)>=0){

x1--; y1++; p+=(2\*dy)-(2\*dx);

}else{

x1--; y1--; p+=(2\*dy)-(2\*dx);

}

}

}

}

}else{

if((y2-y1)>0){

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1++; p+=2\*dx;

}else{

if((x2-x1)>=0){

y1++; x1++; p+=(2\*dx)-(2\*dy);

}else{

y1++; x1--; p+=(2\*dx)-(2\*dy);

}

}

}

}else{

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1--; p+=2\*dx;

}else{

if((x2-x1)>=0){

y1--; x1++; p+=(2\*dx)-(2\*dy);

}else{

y1--; x1--; p+=(2\*dx)-(2\*dy);

}

}

}

}

}

}

public void paint(Graphics g){

super.paint(g);

Graphics2D g2 = (Graphics2D) g;

sketch(x1,y1,x2,y2,g2);

sketch(x1,y1,x3,y3,g2);

sketch(x2,y2,x3,y3,g2);

double[][] scale={{sx,0,xc\*(1-sx)},{0,sy,yc\*(1-sy)},{0,0,1}};

double[][] coor = {{x1,x2,x3},{y1,y2,y3},{1,1,1}};

double[][] mat = new double[3][3] ;

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

for(int j=0;j<3;j++) {

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

prod = prod+scale[l][k]\*coor[k][j];

}

mat [l][j]= prod;

prod = 0; // resetting value of prod

}

}

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

for(int j=0;j<3;j++){

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

if(l==0){

x1 = mat[0][0];

x2 = mat[0][1];

x3 = mat[0][2];

}else if(l==1){

y1 = mat[1][0];

y2 = mat[1][1];

y3 = mat[1][2];

}

}

}

}

sketch(x1,y1,x2,y2,g2);

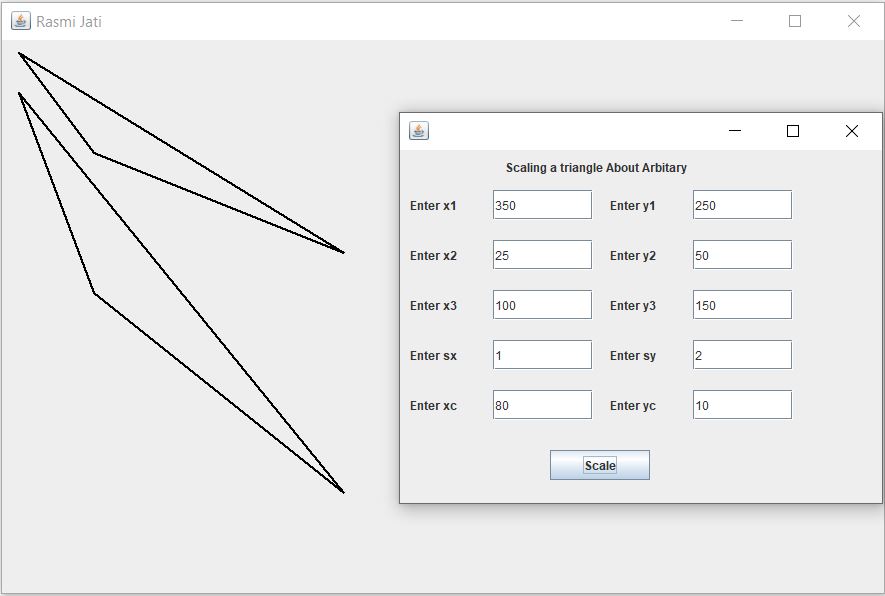
sketch(x1,y1,x3,y3,g2);

sketch(x2,y2,x3,y3,g2);

}

}

## Output:



# Write a program to translate a line.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class TranslationLine extends JFrame implements ActionListener{

JLabel lblx1,lblx2,lbly1,lbly2,lbltx,lblty,lbl;

JTextField tfx1,tfx2,tfy2,tfy1,tftx,tfty;

JButton btnTranslate;

public TranslationLine(){

lbl = new JLabel("Translation of Line");

lbl.setBounds(105,5,150,25);

lblx1 = new JLabel("Enter x1");

lblx1.setBounds(10,40,80,30);

lblx2 = new JLabel("Enter x2");

lblx2.setBounds(210,40,80,30);

lbly1 = new JLabel("Enter y1");

lbly1.setBounds(10,90,80,30);

lbly2 = new JLabel("Enter y2");

lbly2.setBounds(210,90,80,30);

lbltx = new JLabel("Enter tx");

lbltx.setBounds(10,140,80,30);

lblty = new JLabel("Enter ty");

lblty.setBounds(210,140,80,30);

tfx1 = new JTextField(80); tfx1.setBounds(93,40,100,30);

tfx2 = new JTextField(80); tfx2.setBounds(293,40,100,30);

tfy1 = new JTextField(80); tfy1.setBounds(93,90,100,30);

tfy2 = new JTextField(80); tfy2.setBounds(293,90,100,30);

tftx = new JTextField(80); tftx.setBounds(93,140,100,30);

tfty = new JTextField(80); tfty.setBounds(293,140,100,30);

btnTranslate = new JButton("Translate");

btnTranslate.setBounds(150,190,100,30);

add(lbl); add(lblx1); add(tfx1); add(lblx2); add(tfx2);

add(lbly1); add(tfy1); add(lbly2); add(tfy2); add(lbltx);

add(tftx); add(lblty); add(tfty); add(btnTranslate);

btnTranslate.addActionListener(this);

setLayout(null); setSize(500,400); setVisible(true);

}

public void actionPerformed(ActionEvent e){

new TranslationClass(tfx1.getText(), tfy1.getText(), tfx2.getText(), tfy2.getText(),tftx.getText(), tfty.getText());

}

public static void main(String []args){

new TranslationLine();

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

class TranslationClass extends JFrame{

double x1,x2,y2,y1,x,y,steps,i,dx,dy;

double tx,ty,prod = 0,xIncreament,yIncreament;

public TranslationClass(String a1, String b1, String a2, String b2,String ta, String tb){

JPanel p = new JPanel();

getContentPane().add(p);

x1 = Double.parseDouble(a1); y1 = Double.parseDouble(b1);

x2 = Double.parseDouble(a2); y2 = Double.parseDouble(b2);

tx = Double.parseDouble(ta); ty = Double.parseDouble(tb);

setSize(800,500); setTitle("Rasmi Jati"); setVisible(true);

}

public void paint(Graphics g){

super.paint(g);

Graphics2D g2 = (Graphics2D) g;

if(x1<x2){

x= x1; y = y1;

}else{

x = x2; y = y2;

}

dx = x2-x1; dy = y2-y1;

if(Math.abs(dx)>Math.abs(dy)){

steps = Math.abs(dx);

}else{

steps = Math.abs(dy);

}

double xIncreament = dx/steps;

double yIncreament = dy/steps;

for(i=0;i<steps;i++){

g2.draw(new Ellipse2D.Double(x,y,1,1));

if(x1<x2){

x = x+xIncreament;

}else{

x = x-xIncreament;

}

if(y1<y2){

y = y+yIncreament;

}else{

y = y-yIncreament;

}

}

double[][] tran = {{1,0,tx},{0,1,ty},{0,0,1}};

double[][] coor = {{x1,x2},{y1,y2},{1,1}};

double[][] mat = new double[3][2] ;

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

for(int j=0;j<2;j++) {

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

prod = prod+tran[l][k]\*coor[k][j];

}

mat [l][j]= prod; prod = 0;

}

}

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

for(int j=0;j<2;j++){

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

if(l==0){

x1 = mat[0][0];x2 = mat[0][1];

}else if(l==1){

y1 = mat[1][0]; y2 = mat[1][1];

}

}

}

}

if(x1<x2){

x= x1;y = y1;

}else{

x = x2;y = y2;

}

dx = x2-x1;dy = y2-y1;

if(Math.abs(dx)>Math.abs(dy)){

steps = Math.abs(dx);

}else{

steps = Math.abs(dy);

}

xIncreament = dx/steps;yIncreament = dy/steps;

for(i=0; i<steps; i++){

if(x1<x2){

x = x+xIncreament;

}else{

x = x-xIncreament;

}

if(y1<y2){

y = y+yIncreament;

}else{

y = y-yIncreament;

}

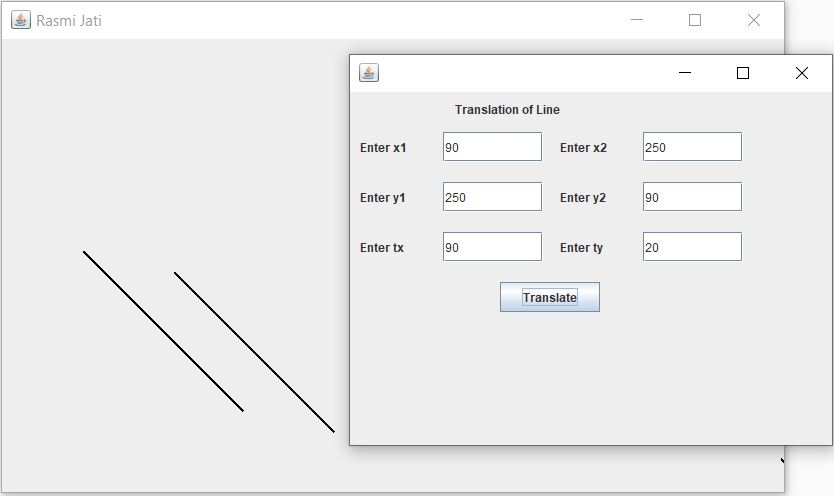
g2.draw(new Ellipse2D.Double(x,y,1,1));

}

}

}

## Output:



# Write a program to rotate a triangle about arbitrary point.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class RotationArbitary extends JFrame implements ActionListener{

JLabel lblx1,lblx2,lbly1,lbly2,lbl,lblx3,lbly3,lblangle;

JLabel lblxc,lblyc; JButton btnDraw;

JTextField tx1,tx2,ty1,ty2,tx3,ty3,tangle,txc,tyc;

public RotationArbitary(){

lbl = new JLabel("Rotate Triangle");

lbl.setBounds(105,5,150,25);

lblx1 = new JLabel("Enter x1");

lblx1.setBounds(10,40,80,30);

lblx2 = new JLabel("Enter x2");

lblx2.setBounds(210,40,80,30);

lbly1 = new JLabel("Enter y1");

lbly1.setBounds(10,90,80,30);

lbly2 = new JLabel("Enter y2");

lbly2.setBounds(210,90,80,30);

lblx3 = new JLabel("Enter x3");

lblx3.setBounds(10,140,80,30);

lbly3 = new JLabel("Enter y3");

lbly3.setBounds(210,140,80,30);

lblxc = new JLabel("Enter xc");

lblxc.setBounds(10,190,80,30);

lblyc = new JLabel("Enter yc");

lblyc.setBounds(210,190,80,30);

lblangle = new JLabel("Enter angle");

lblangle.setBounds(10,240,80,30);

tx1 = new JTextField(80); tx1.setBounds(93,40,100,30);

tx2 = new JTextField(80); tx2.setBounds(293,40,100,30);

ty1 = new JTextField(80); ty1.setBounds(93,90,100,30);

ty2 = new JTextField(80); ty2.setBounds(293,90,100,30);

tx3 = new JTextField(80); tx3.setBounds(93,140,100,30);

ty3 = new JTextField(80); ty3.setBounds(293,140,100,30);

txc = new JTextField(80); txc.setBounds(93,190,100,30);

tyc = new JTextField(80); tyc.setBounds(293,190,100,30);

tangle = new JTextField(80);

tangle.setBounds(93,240,100,30);

btnDraw = new JButton("Draw");

btnDraw.setBounds(152,290,100,30);

btnDraw.addActionListener(this);

add(lbl); add(lblx1); add(tx1); add(lblx2); add(tx2);

add(lbly1); add(ty1); add(lbly2);add(ty2); add(lblx3);

add(tx3); add(lbly3); add(ty3); add(lblxc); add(txc);

add(lblyc); add(tyc);add(lblangle); add(tangle);

add(btnDraw); btnDraw.addActionListener(this);

setLayout(null); setSize(500,400);

setTitle("Rasmi Jati"); setVisible(true);

}

public void actionPerformed(ActionEvent e){

new RotationArbitaryClass(tx1.getText(),ty1.getText(),tx2.getText(),ty2.getText(),tx3.getText(),ty3.getText(),txc.getText(),tyc.getText(),tangle.getText());

}

public static void main(String[] args){

new RotationArbitary() ;

}}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

public class RotationArbitaryClass extends JFrame{

double x,y,steps,i,dx,dy,p,x1,x2,y2,y1,y3,x3,angle,c,s,xc,yc;

public RotationArbitaryClass(String a1, String b1, String a2, String b2,String a3, String b3,String ac, String bc,String a){

JPanel p = new JPanel();

getContentPane().add(p);

x1 = Double.parseDouble(a1);x2 = Double.parseDouble(a2);

y1 = Double.parseDouble(b1);y2 = Double.parseDouble(b2);

x3 = Double.parseDouble(a3);y3 = Double.parseDouble(b3);

xc = Double.parseDouble(ac);yc = Double.parseDouble(bc);

angle = Double.parseDouble(a);

angle = Math.toRadians(angle);

setSize(800,500);setTitle("Rasmi Jati");setVisible(true);

}

public void sketch(double x1,double y1,double x2,double y2,Graphics2D g){

dx=Math.abs(x2-x1); dy=Math.abs(y2-y1);

if (dx>dy){

if((x2-x1)>=0){

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1++; p+=2\*dy;

}else{

if((y2-y1)>=0){

x1++; y1++; p+=(2\*dy)-(2\*dx);

}else{

x1++; y1--; p+=(2\*dy)-(2\*dx);

}

}

}

}else{

p=(2\*dy)-dx;

for(int i=0;i<=dx;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

x1--; p+=2\*dy;

}else{

if((y2-y1)>=0){

x1--; y1++; p+=(2\*dy)-(2\*dx);

}else{

x1--; y1--; p+=(2\*dy)-(2\*dx);

}

}

}

}

}else{

if((y2-y1)>0){

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1++; p+=2\*dx;

}else{

if((x2-x1)>=0){

y1++; x1++; p+=(2\*dx)-(2\*dy);

}else{

y1++; x1--; p+=(2\*dx)-(2\*dy);

}

}

}

}else{

p=(2\*dx)-dy;

for(int i=0;i<=dy;i++){

g.draw(new Ellipse2D.Double(x1, y1, 1, 1));

if(p<0){

y1--; p+=2\*dx;

}else{

if((x2-x1)>=0){

y1--; x1++; p+=(2\*dx)-(2\*dy);

}else{

y1--; x1--; p+=(2\*dx)-(2\*dy);

}

}

}

}

}

}

public void paint(Graphics g){

super.paint(g);

Graphics2D g2 = (Graphics2D) g;

sketch(x1,y1,x2,y2,g2);

sketch(x2,y2,x3,y3,g2);

sketch(x1,y1,x3,y3,g2);

c = Math.cos(angle);

s = Math.sin(angle);

x1 = xc\*(1-c)+(yc\*s)+(x1\*c)-(y1\*s);

y1 = yc\*(1-c)-(xc\*s)+(x1\*s)+(y1\*c);

x2 = xc\*(1-c)+(yc\*s)+(x2\*c)-(y2\*s);

y2 = yc\*(1-c)-(xc\*s)+(x2\*s)+(y2\*c);

x3 = xc\*(1-c)+(yc\*s)+(x3\*c)-(y3\*s);

y3 = yc\*(1-c)-(xc\*s)+(x3\*s)+(y3\*c);

sketch(x1,y1,x2,y2,g2);

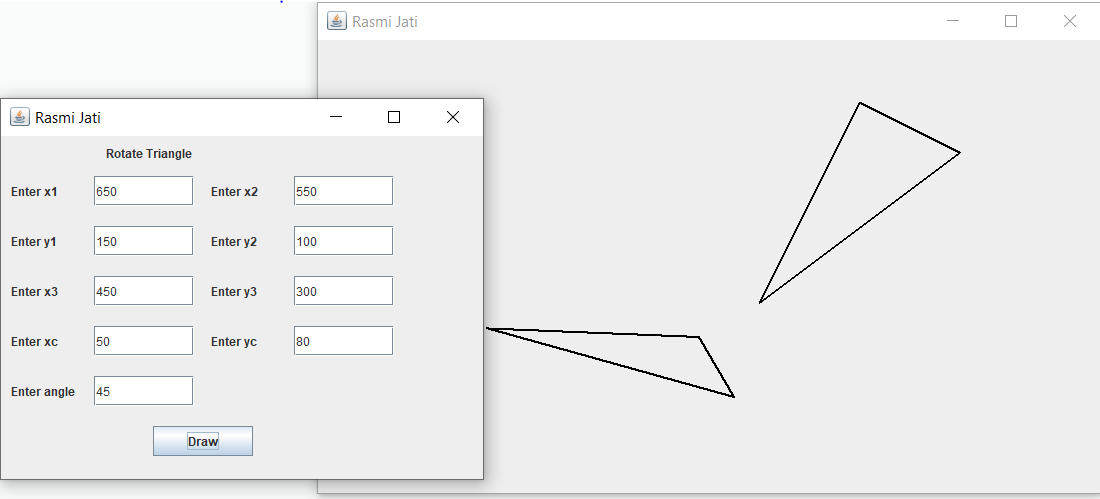
sketch(x2,y2,x3,y3,g2);

sketch(x1,y1,x3,y3,g2);

}

}

## Output:



# Write a program to shear a rectangle through x- direction.

## Source Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class ShearX extends JFrame implements ActionListener{

JLabel lblx1,lblx2,lbly1,lbly2,lblx3,lblx4,lbly3,lbly4,lbl,lbl1;

JTextField tfx1,tfx2,tfy1,tfy2,tfx3,tfx4,tfy3,tfy4,tf;

JButton btnDraw;

public ShearX(){

lbl1 = new JLabel("Shearing X-Direction");

lblx1 = new JLabel("Enter x1");

lblx2 = new JLabel("Enter x2");

lbly1 = new JLabel("Enter y1");

lbly2 = new JLabel("Enter y2");

lblx3 = new JLabel("Enter x3");

lblx4 = new JLabel("Enter x4");

lbly3 = new JLabel("Enter y3");

lbly4 = new JLabel("Enter y4");

lbl = new JLabel("Enter shx");

tfx1 = new JTextField(80);

tfx2 = new JTextField(80);

tfx3 = new JTextField(80);

tfx4 = new JTextField(80);

tfy1 = new JTextField(80);

tfy2 = new JTextField(80);

tfy3 = new JTextField(80);

tfy4 = new JTextField(80);

tf = new JTextField(80);

btnDraw = new JButton("Draw");

lbl1.setBounds(105,5,200,25);

lblx1.setBounds(10,40,80,30);

lblx2.setBounds(210,40,80,30);

lbly1.setBounds(10,90,80,30);

lbly2.setBounds(210,90,80,30);

lblx3.setBounds(10,140,80,30);

lblx4.setBounds(210,140,80,30);

lbly3.setBounds(10,190,80,30);

lbly4.setBounds(210,190,80,30);

lbl.setBounds(10,240,80,30);

tfx1.setBounds(93,40,100,30);

tfx2.setBounds(293,40,100,30);

tfy1.setBounds(93,90,100,30);

tfy2.setBounds(293,90,100,30);

tfx3.setBounds(93,140,100,30);

tfx4.setBounds(293,140,100,30);

tfy3.setBounds(93,190,100,30);

tfy4.setBounds(293,190,100,30);

tf.setBounds(93,240,100,30);

btnDraw.setBounds(152,290,100,30);

add(lbl1); add(lbl); add(tf); add(lblx1); add(tfx1);

add(lblx2); add(tfx2); add(lblx3); add(tfx3);

add(lblx4); add(tfx4); add(lbly1); add(tfy1);

add(lbly2); add(tfy2); add(lbly3); add(tfy3);

add(lbly4); add(tfy4); add(btnDraw);

btnDraw.addActionListener(this);

setLayout(null); setSize(500,400);

setTitle("Rasmi Jati"); setVisible(true);

}

public void actionPerformed(ActionEvent e){

new ShearXClass(tfx1.getText(),tfy1.getText(),tfx2.getText(),tfy2.getText(),tfx3.getText(),tfy3.getText(),tfx4.getText(),tfy4.getText(),tf.getText());

}

public static void main(String[] args) {

new ShearX();

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

class ShearXClass extends JFrame{

int x1,x2,x3,x4,y1,y2,y3,y4,shx,px1,px2,px3,px4;

public ShearXClass(String a1, String b1, String a2, String b2,String a3,String b3,String a4, String b4,String x){

x1 = Integer.parseInt(a1);

y1 = Integer.parseInt(b1);

x2 = Integer.parseInt(a2);

y2 = Integer.parseInt(b2);

x3 = Integer.parseInt(a3);

y3 = Integer.parseInt(b3);

x4 = Integer.parseInt(a4);

y4 = Integer.parseInt(b4);

shx = Integer.parseInt(x);

px1 = x1+y1\*shx;

px2 = x2+y2\*shx;

px3 = x3+y3\*shx;

px4 = x4+y4\*shx;

setSize(800,500);

setTitle("Rasmi Jati");

setVisible(true);

}

public void paint(Graphics g){

g.drawLine(x1,y1,x2,y2);

g.drawLine(x2,y2,x3,y3);

g.drawLine(x3,y3,x4,y4);

g.drawLine(x4,y4,x1,y1);

g.drawLine(px1,y1,px2,y2);

g.drawLine(px2,y2,px3,y3);

g.drawLine(px3,y3,px4,y4);

g.drawLine(px4,y4,px1,y1);

}

}

## Output:

