SE CMPN A Roll no:30

/\*Program to implement Midpoint circle drawing algorithm \*/

import java.util.\*;

import java.applet.Applet;

import java.awt.\*;

/\*<applet code="MidCircle.class" height=500 width=500>

</applet>\*/

public class MidCircle extends Applet

{

int r;

double p0;

int x,y;

public void init()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the radius of circle");

r=sc.nextInt();

}

public void draw(int p,Graphics g)

{

x=0;

y=r;

p0=(5/4)-(r);

do{

g.drawLine((int)-x+150,(int)y+150,(int)x+150,(int)-y+150);

g.drawLine((int)x+150,(int)-y+150,(int)-x+150,(int)y+150);

g.drawLine((int)-x+150,(int)-y+150,(int)x+150,(int)y+150);

g.drawLine((int)x+150,(int)y+150,(int)-x+150,(int)-y+150);

g.drawLine((int)-y+150,(int)x+150,(int)y+150,(int)-x+150);

g.drawLine((int)y+150,(int)-x+150,(int)-y+150,(int)x+150);

g.drawLine((int)-y+150,(int)-x+150,(int)y+150,(int)x+150);

g.drawLine((int)y+150,(int)-x+150,(int)-y+150,(int)-x+150);

if(p0<0)

{

x=x+1;

y=y;

p0=p0+(2\*(x))+1;

}

else

{

x=x+1;

y=y-1;

p0=p0+(2\*x)-(2\*y)+3;

}

}while(x<y);

}

public void paint(Graphics g)

{ int i,j;

Thread t =new Thread();

try

{

for( i=15;i<=50;)

{

g.setColor(Color.black);

draw(i,g);

t.sleep(1000);

r=r+5;

i=i+5;

g.setColor(Color.white);

draw(i,g);

t.sleep(1000);

}

for( j=50;j>=15;)

{

g.setColor(Color.white);

draw(j,g);

t.sleep(1000);

r=r-5;

j=j-5;

g.setColor(Color.black);

draw(j,g);

t.sleep(1000);

}

}

catch(InterruptedException e)

{

}

}

}

/\*

Output:

D:\Flevia 30>javac MidCircle.java

D:\Flevia 30>appletviewer MidCircle.java

Enter the radius of circle

15





