

ASSIGNMENT-3

NAME-ANUBHAV ANAND

ENROLLMENT NUMBER-2020CSB102

**SUBJECT-ASSIGNMENT 3 OF
COMPUTER GRAPHICS**

G-suite Id-

**[2020CSB102.anubhav@students.iiests
.ac.in](mailto:2020CSB102.anubhav@students.iiests.ac.in)**

1.Q.Develop a class for circle using Midpoint circle drawing algorithm.

Ans-Making Circle- Circle.java

```
/*This is the program for zoom-out and zoom-in using graphics */
/*Thus program can run only on Java-jdk(8) */

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class Circle extends Applet implements
ActionListener,MouseWheelListener{
    //It is for generating a rectangle corresponding to a particular
    point in cartesian coordinate syatem
    int gap = 4;
    ArrayList<Circle1>arr=new ArrayList<Circle1>(1000);
    public void plotPoint(Graphics g,int x,int y,Color c)
    {
        g.setColor(c);
        g.fillRect(
            (getX()+getWidth())/2+(x*gap)-(gap),
            (getY()+getHeight())/2-(y*gap)-(gap),
            gap,gap
        );
    }
    //function to make grid
    public void paintGrid(Graphics g,int gap,int originx,int originy)
    {
        g.setColor(Color.yellow);

        for(int i = gap;i<=getWidth();i+=gap)
        {
            g.drawLine(originx+i, originy-getHeight()/2, originx+i,
            originy+getHeight()/2);
            g.drawLine(originx-i, originy-getHeight()/2, originx-i,
            originy+getHeight()/2);
        }
        for(int i = gap;i<=getHeight();i+=gap)
```

```

        {
            g.drawLine(originx-getWidth()/2, originy+i,
originx+getWidth()/2, originy+i);
            g.drawLine(originx-getWidth()/2, originy-i,
originx+getWidth()/2, originy-i);

        }
    }
    //function to return line
    public void Circles(Graphics g,int radius,int x1,int y1)
    {
        int x=0;
        int y=radius;
        double p=1.25-radius;
        plotPoint(g,x+x1,y+y1,Color.green);
        plotPoint(g,x+x1,-y+y1,Color.green);
        plotPoint(g,-x+x1,y+y1,Color.green);
        plotPoint(g,-x+x1,-y+y1,Color.green);
        while(x<y)
        {
            if(p<0)
            {
                x=x+1;
                p=p+2*x+1;

            }
            else
            {
                x=x+1;
                y=y-1;
                p=p+2*x+1-2*y;
            }
            plotPoint(g,x+x1,y+y1,Color.green);
            plotPoint(g,y+x1,x+y1,Color.green);
            plotPoint(g,x+x1,-y+y1,Color.green);
            plotPoint(g,-x+x1,y+y1,Color.green);
            plotPoint(g,y+x1,-x+y1,Color.green);
            plotPoint(g,-y+x1,x+y1,Color.green);
            plotPoint(g,-x+x1,-y+y1,Color.green);
            plotPoint(g,-y+x1,-x+y1,Color.green);
        }
    }
}

// }
//It is for initialisation purpose
public void init(){
    addMouseWheelListener(this);
    button1 = new Button("+");
}

```

```

        add(button1);
        button1.addActionListener(this);
        button2 = new Button("-");
        add(button2);
        button1.setBackground(Color.white);
        button2.setBackground(Color.white);
        button2.addActionListener(this);
        setForeground(Color.green);
        setBackground(Color.black);
    }
    //it is for implementing button function
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource() == button1){
            gap+=gap+gap/10;
            repaint();
        }
        else if(e.getSource()==button2)
        {
            gap-=gap/10;
            repaint();
        }
    }
    //It is for mouse wheel operation
    public void mouseWheelMoved(MouseWheelEvent e)
    {
        int z=e.getWheelRotation();
        gap+=z;
        repaint();
    }

    Button button1, button2;
    public void paint(Graphics g){

        g.setColor(Color.orange);
        int originx=getX()+getWidth()/2;
        int originy=getY()+getHeight()/2;
        g.drawLine(originx-getWidth()/2, originy, originx+getWidth()/2,
originy);
        g.drawLine(originx, originy-getHeight()/2, originx,
originy+getHeight()/2);
        // paintGrid(g,gap,originx,originy);
        paintGrid(g,gap,originx,originy);
        Color c1=new Color(100,100,100);

        Circles(g,100,0,0);
        Circles(g,46,46,27);
        Circles(g,46,-46,27);
    }

```

```
Circles(g, 46, 0, -54);
Circles(g, 8, 0, 0);
Circles(g, 22, 0, 78);
Circles(g, 22, 65, -38);
Circles(g, 22, -65, -38);
Circles(g, 10, 33, 82);
Circles(g, 10, -33, 82);
Circles(g, 10, -88, -14);
Circles(g, 10, 88, -14);
Circles(g, 10, -54, -70);
Circles(g, 10, 54, -70);
Circles(g, 5, 50, 80);
Circles(g, 5, -50, 80);
Circles(g, 6, 94, 3);
Circles(g, 6, -94, 3);
Circles(g, 6, 45, -83);
Circles(g, 6, -45, -83);
Circles(g, 4, 58, 76);
Circles(g, 4, -58, 76);
Circles(g, 3, 95, 14);
Circles(g, 3, -95, 14);
Circles(g, 3, 36, -88);
Circles(g, 3, -36, -88);
Circles(g, 4, 0, 50);
Circles(g, 4, 42, -24);
Circles(g, 4, -42, -24);
Circles(g, 2, 65, 72);
Circles(g, 2, -65, -72);
Circles(g, 3, 95, 22);
    Circles(g, 3, -95, 22);
    Circles(g, 2, 31, -92);
    Circles(g, 2, -31, -92);
    Circles(g, 4, 23, 92);
    Circles(g, 4, -23, 92);
    Circles(g, 4, 91, -30);
    Circles(g, 4, -91, -30);
    Circles(g, 4, 69, -64);
    Circles(g, 4, -69, -64);
Circle1 c = new
Circle1(100, 0, 0);
arr.add(c);
c = new Circle1(46, 46, 27);
arr.add(c);
c=new Circle1(46, -46, 27);
arr.add(c);
c=new Circle1(46, 0, -54);
arr.add(c);
//
```

```
c=new Circle1(8,0,0);
arr.add(c);
c=new Circle1(22,0,78);
arr.add(c);
c=new Circle1(22,65,-38);
arr.add(c);
c=new Circle1(22,-65,-38);
arr.add(c);
c=new Circle1(10,33,82);
arr.add(c);
c=new Circle1(10,-33,82);
arr.add(c);
c=new Circle1(10,-88,-14);
arr.add(c);
c=new Circle1(10,88,-14);
arr.add(c);
c=new Circle1(10,-54,-70);
arr.add(c);
c=new Circle1(10,54,-70);
arr.add(c);
c=new Circle1(5,50,80);
arr.add(c);
c=new Circle1(5,-50,80);
arr.add(c);
c=new Circle1(6,94,3);
arr.add(c);
c=new Circle1(6,-94,3);
arr.add(c);
c=new Circle1(6,45,-83);
arr.add(c);
c=new Circle1(6,-45,-83);
arr.add(c);
c=new Circle1(4,58,76);
arr.add(c);
c=new Circle1(4,-58,76);
arr.add(c);
c=new Circle1(3,95,14);
arr.add(c);
c=new Circle1(3,-95,14);
arr.add(c);
c=new Circle1(3,36,-88);
arr.add(c);
c=new Circle1(3,-36,-88);
arr.add(c);
c=new Circle1(4,0,50);
arr.add(c);
c=new Circle1(4,42,-24);
arr.add(c);
```

```

c=new Circle1(4,-42,-24);
arr.add(c);
c=new Circle1(2,65,72);
arr.add(c);
c=new Circle1(2,-65,-72);
arr.add(c);
c=new Circle1(3,95,22);
arr.add(c);
c=new Circle1(3,-95,22);
arr.add(c);
c=new Circle1(2,31,-92);
arr.add(c);
c=new Circle1(2,-31,-92);
arr.add(c);
c=new Circle1(4,23,92);
arr.add(c);
c=new Circle1(4,-23,92);
arr.add(c);
c=new Circle1(4,91,-30);
arr.add(c);
c=new Circle1(4,-91,-30);
arr.add(c);
c=new Circle1(4,69,-64);
arr.add(c);
c=new Circle1(4,-69,-64);
arr.add(c);
int p,q,r1=0;

int i=0;
Circle1 largec = arr.get(0);
int flag=0;
while(i<200){
    try
    {
        Thread.sleep(5);
    }
    catch(Exception e)
    {
    }
    flag=0;
    p=(int) (Math.random()*200)-100;
    q=(int) (Math.random()*200)-100;
    r1 = 0;
    while(true){
        for(int it=0;it<arr.size();it++)
        {
            Circle1 m=arr.get(it);

```

```

        int r=m.r;
        int x1=m.x1;
        int y1=m.y1;
        int f=0;
        if(it>0)
        {
            int d=(int) (Math.sqrt (Math.pow (p-
x1,2)+Math.pow (q-y1,2)));
            if (r+r1>d)
            {
                flag=1;
                break;
            }
        }
        else{
            int d=(int) (Math.sqrt (Math.pow (p-
x1,2)+Math.pow (q-y1,2)));
            if (r-r1<d)
            {
                flag=1;
                break;
            }
        }

        if (flag==1)
            break;
        r1++;
    }

    if (r1>0){
        Circles(g,r1,p,q);
        c = new Circle1(r1,p,q);
        arr.add(c);
        // System.out.println("c =" +p+" "+q+" r="+r1);
    }
    i++;
}
arr.clear();
}
}

```

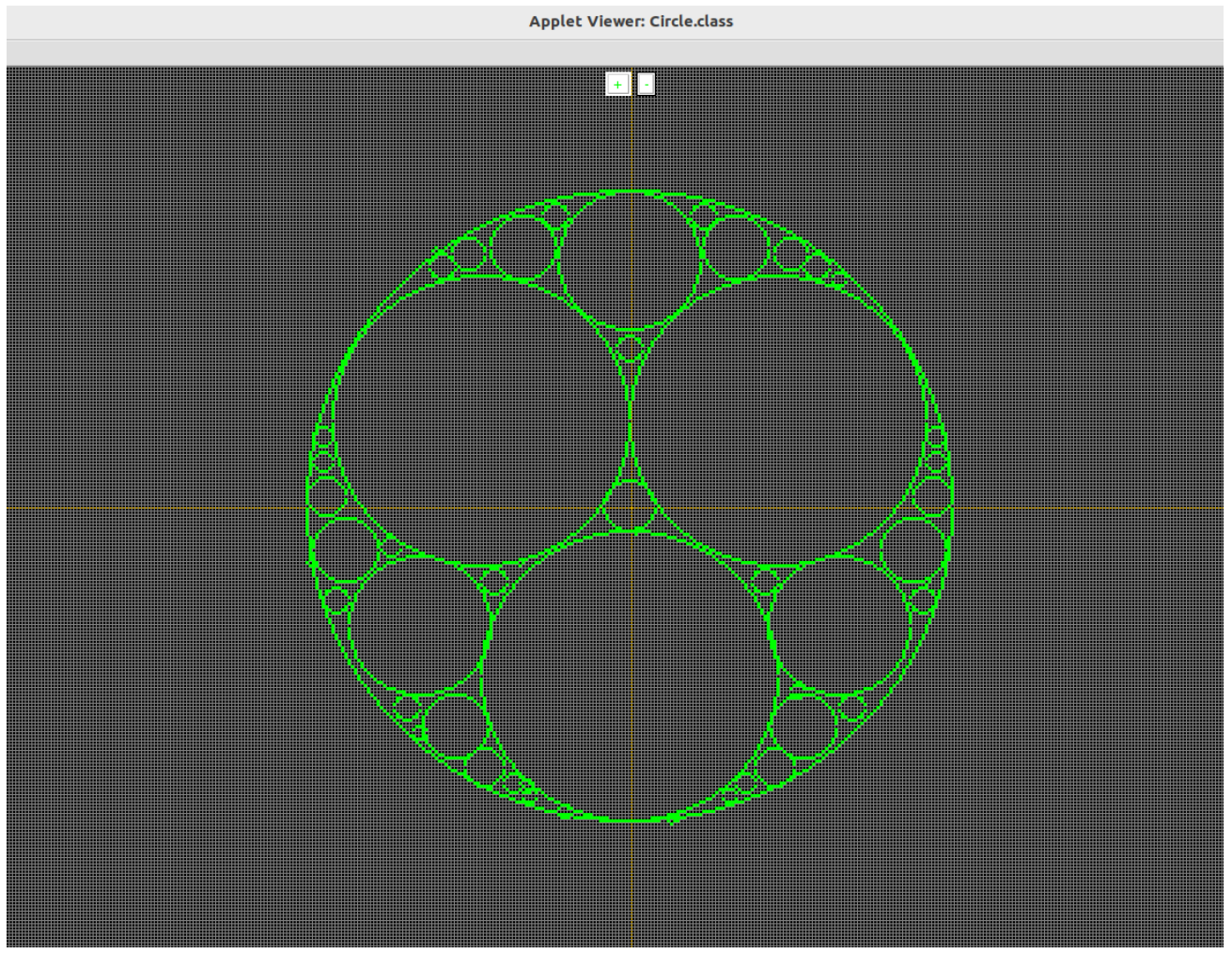

Circle.html-

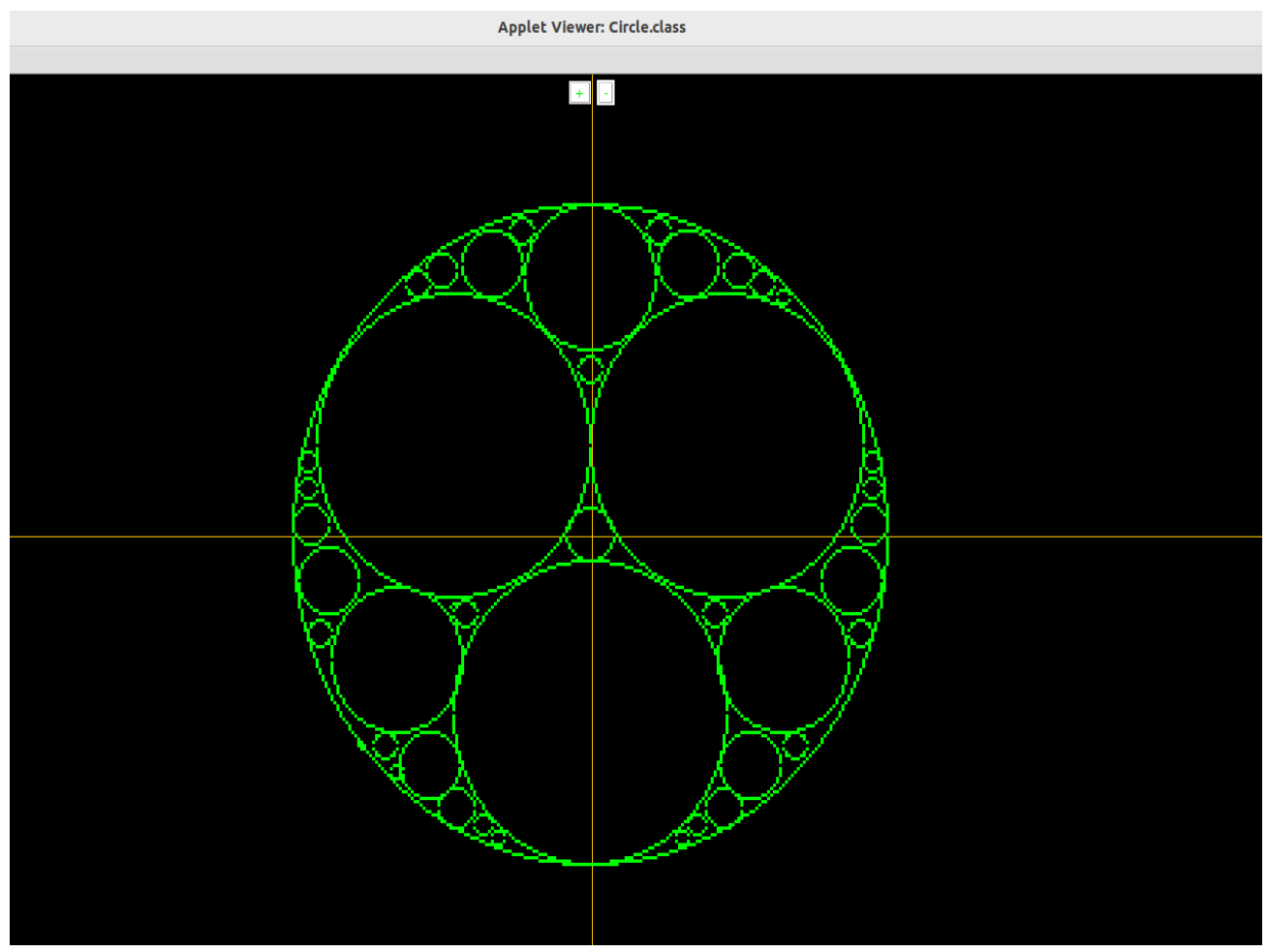
```
<html>
  <head></head>
  <body>
    <applet code ="Circle.class"  height="1000"
width="1000"></applet>
  </body>
</html>
```

Circle1.java-

```
public class Circle1{
    int r,x1,y1;
    Circle1(int a,int b,int c)
    {
        r=a;
        x1=b;
        y1=c;
    }
}
```

Now output-





2>Q. Develop a class for ellipse using Midpoint ellipse drawing algorithm.

Ans-

Ellipse Making-

Ellipse.java-

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class Ellipse extends Applet implements
ActionListener,MouseWheelListener{
    //It is for generating a rectangle corresponding to a particular
    point in cartesian coordinate syatem
    int gap = 100;
    public void plotPoint(Graphics g,int x,int y,Color c)
    {
        g.setColor(c);
        g.fillRect(
            (getX()+getWidth())/2+(x*gap)-(gap/4),
            (getY()+getHeight())/2-(y*gap)-(gap/4),
            gap/2,gap/2
        );
    }
    public int slope(int x1,int x2,int y1,int y2)
    {
        int x=x2-x1;
        int y=y2-y1;
        int m=y/x;
        return m;
    }

    //Ellipse drawing algorithm
    public void midptellipse(Graphics g,float rx, float ry,
```

```

float xc, float yc)
{

float dx, dy, d1, d2, x, y;
x = 0;
y = ry;

// Initial decision parameter of region 1
d1 = (ry * ry) - (rx * rx * ry) +
      (0.25f * rx * rx);
dx = 2 * ry * ry * x;
dy = 2 * rx * rx * y;
// DecimalFormat df = new DecimalFormat("#,###,##0.00000");

// For region 1
while (dx < dy)
{

plotPoint(g, (int) (x+xc), (int) (y+yc), Color.red);
plotPoint(g, (int) (-x+xc), (int) (y+yc), Color.red);
plotPoint(g, (int) (x+xc), (int) (-y+yc), Color.red);
plotPoint(g, (int) (-x+xc), (int) (-y+yc), Color.red);

    if (d1 < 0)
    {
        x++;
        dx = dx + (2 * ry * ry);
        d1 = d1 + dx + (ry * ry);
    }
    else
    {
        x++;
        y--;
        dx = dx + (2 * ry * ry);
        dy = dy - (2 * rx * rx);
        d1 = d1 + dx - dy + (ry * ry);
    }
}

// Decision parameter of region 2
d2 = ((ry * ry) * ((x + 0.5f) * (x + 0.5f)))
      + ((rx * rx) * ((y - 1) * (y - 1)))
      - (rx * rx * ry * ry);

// Plotting points of region 2
while (y >= 0) {

    // printing points based on 4-way symmetry

```

```

        plotPoint(g, (int) (x+xc), (int) (y+yc), Color.red);
        plotPoint(g, (int) (-x+xc), (int) (y+yc), Color.red);
        plotPoint(g, (int) (x+xc), (int) (-y+yc), Color.red);
        plotPoint(g, (int) (-x+xc), (int) (-y+yc), Color.red);

        if (d2 > 0) {
            y--;
            dy = dy - (2 * rx * rx);
            d2 = d2 + (rx * rx) - dy;
        }
        else {
            y--;
            x++;
            dx = dx + (2 * ry * ry);
            dy = dy - (2 * rx * rx);
            d2 = d2 + dx - dy + (rx * rx);
        }
    }
}

public void paintGrid(Graphics g, int gap, int originx, int originy)
{
    g.setColor(Color.yellow);

    for(int i = gap; i <= getWidth(); i += gap)
    {
        g.drawLine(originx+i, originy-getHeight()/2, originx+i,
originy+getHeight()/2);
        g.drawLine(originx-i, originy-getHeight()/2, originx-i,
originy+getHeight()/2);
    }
    for(int i = gap; i <= getHeight(); i += gap)
    {
        g.drawLine(originx-getWidth()/2, originy+i,
originx+getWidth()/2, originy+i);
        g.drawLine(originx-getWidth()/2, originy-i,
originx+getWidth()/2, originy-i);
    }
}

}

//It is for initialisation purpose
public void init(){

```

```

        addMouseWheelListener(this);
        button1 = new Button("+");
        add(button1);
        button1.addActionListener(this);
        button2 = new Button("-");
        add(button2);
        button1.setBackground(Color.white);
        button2.setBackground(Color.white);
        button2.addActionListener(this);
        setForeground(Color.green);
        setBackground(Color.black);
    }
    //it is for implementing button function
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource() == button1){
            gap+=gap+gap/10;
            repaint();
        }
        else if(e.getSource()==button2)
        {
            gap-=gap/10;
            repaint();
        }
    }
    //It is for mouse wheel operation
    public void mouseWheelMoved(MouseWheelEvent e)
    {
        int z=e.getWheelRotation();
        gap+=z;
        repaint();
    }

    Button button1, button2;
    public void paint(Graphics g){

        g.setColor(Color.orange);
        int originx=getX()+getWidth()/2;
        int originy=getY()+getHeight()/2;
        g.drawLine(originx-getWidth()/2, originy,
originx+getWidth()/2, originy);
        g.drawLine(originx, originy-getHeight()/2, originx,
originx+getHeight()/2);
        // paintGrid(g,gap,originx,originy);
        Color c=new Color(100,100,100);
        int i=0;
        int x1=200,y1=101;
        midptellipse(g,(float)6,(float)3,(float)1,(float)2);
    }

```

```
        midptellipse(g, (float)10, (float)20, (float)1, (float)2);  
        paintGrid(g, gap, originx, originy);  
    }  
}
```

Ellipse.html-

```
<html>  
  <head></head>  
  <body>  
    <applet code ="Ellipse.class"  height="1000"  
width="1000"></applet>  
  </body>  
</html>
```

Ellipse Diagram-

