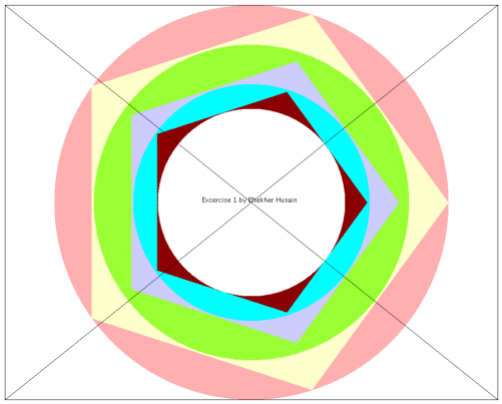
Exercise 1

CSC 211, 1XD

Eftekher Husain | Prof. Hesham A Auda | June 26, 2018



Solution to Polygon Class Problems

a. GetArea method

Since the polygon is a regular polygon and the sides and angles are equal, the area of this polygon can be stated as...

Area of Polygon = (¼) * N * (sideLength * sideLength) / (tan(pi/N))

Where **N** is the number of sides and **sideLength** is the length of each side. Since we few things are still unknown, we use a general equation.

Codes Developed

```
//returns the area of Polygon
public double getArea()
{
    double area = (1/4) * N * ((sideLength * sideLength)/(Math.tan(Math.PI/N)));
    return area;
}
```

b. GetPerimeter

To get the perimeter of a regular polygon, all we have to do is multiply its length of each sides by number of its side.

Perimeter = N * sideLength.

Codes Developed

```
//returns the Perimeter of Polygon
public int getPerimeter()
{return N * sideLength;}
```

c. getAngle

In general to get the angle of a side of a regular polygon, we would divide the number of sides by 360 degrees. But to get the interior angle of the regular polygon we would use a general equation.

180 degree - (360 degree / N) **or** (N-2) * 180 degrees / N.

Codes Developed

```
//returns the interior angle (in degrees) of the Polygon
public double getAngle()
{   double angleInterior = (N-2) * 180/N;
    return angleInterior;
}
```

d. getSide

To get the side of the regular polygon, we first have to declare a variable called side and then we call it through the constructor. Then we use the getSide method which then reuturns the side length of the polygon.

Codes Developed

```
    private int sideLength;


    public Polygon(Color color, int N, int sideLength) {
        super(color);
         this.N = N;
         this.sideLength = sideLength;
    }


//returns the sideLength length of the Polygon
public int getsideLength()
{return sideLength;}
```

Solution to Circle Class Problems

1. getArea

To get the area of a circle, we use the general formula of the area of a circle which is …

Area of a cirlce = Pi * radius * radius

Since Java provides Pi from it's Math interface, we use it to get our area for the circle

Codes Developed

```
//returns the area of circle
public double getArea()
{return Math.PI * radius * radius;}
```

2. getPerimeter

To get the perimeter of a circle, we use the general formula of perimeter of a circle which is…

Perimeter of a circle = 2 * Pi * radius

Codes Developed

```
//returns the perimeter of circle
public double getPerimeter()
{return 2 * Math.PI * radius;}
```

3. getRadius

to get the radius of a circle we first have to declare it in our program. Once declared then we call it in our constructor. And then we call it in our getRadius method as the field of out return.

Codes developed

```
private double radius;

public Circle(double x, double y, Color color, double r) {
    super(x, y, color);
    radius = r;
}


 public double getRadius()
 {return radius;}
```

4. toString

What the toString method does is print out the objects description as a String. For a circle it will return a string representation of circle object which are radius, perimeter and area.

Codes developed

```
    public String toString()
    {   String result = "Circle[ Radius=" +getRadius()+
    ", Perimeter=" + getPerimeter()+ ", Area = " +getArea()+ "]";
        return result;
    }
```

All the Codes Put together

Shape Class (Super Class)

Shape.java

```
 1 importjava.awt.Canvas;
 6
 7 public abstract class Shape extends Canvas {
 8         private int x;
 9         private int y;
10         private int dx;
11         private int dy;
12         private Color color; 13
14         //constructor for the shape Class
15         public Shape(int x, int y, Color color) {
16                 super();
17                 this.x = x;
18                 this.y = y;
19         this.color = color; 20 }
21
22
23         //getter methods
24         public int getX() {
25         return x; 26  }
27         public int getY() {
28         return y; 29  }
30         public int getDx() {
31         return dx; 32}
33         public int getDy() {
34         return dy; 35}
36         public Color getColor() {
37         return color; 38        }
39
40
41         //setter methods
42         public void setX(int x) {
43         this.x = x; 44 }
45         public void setY(int y) {
```

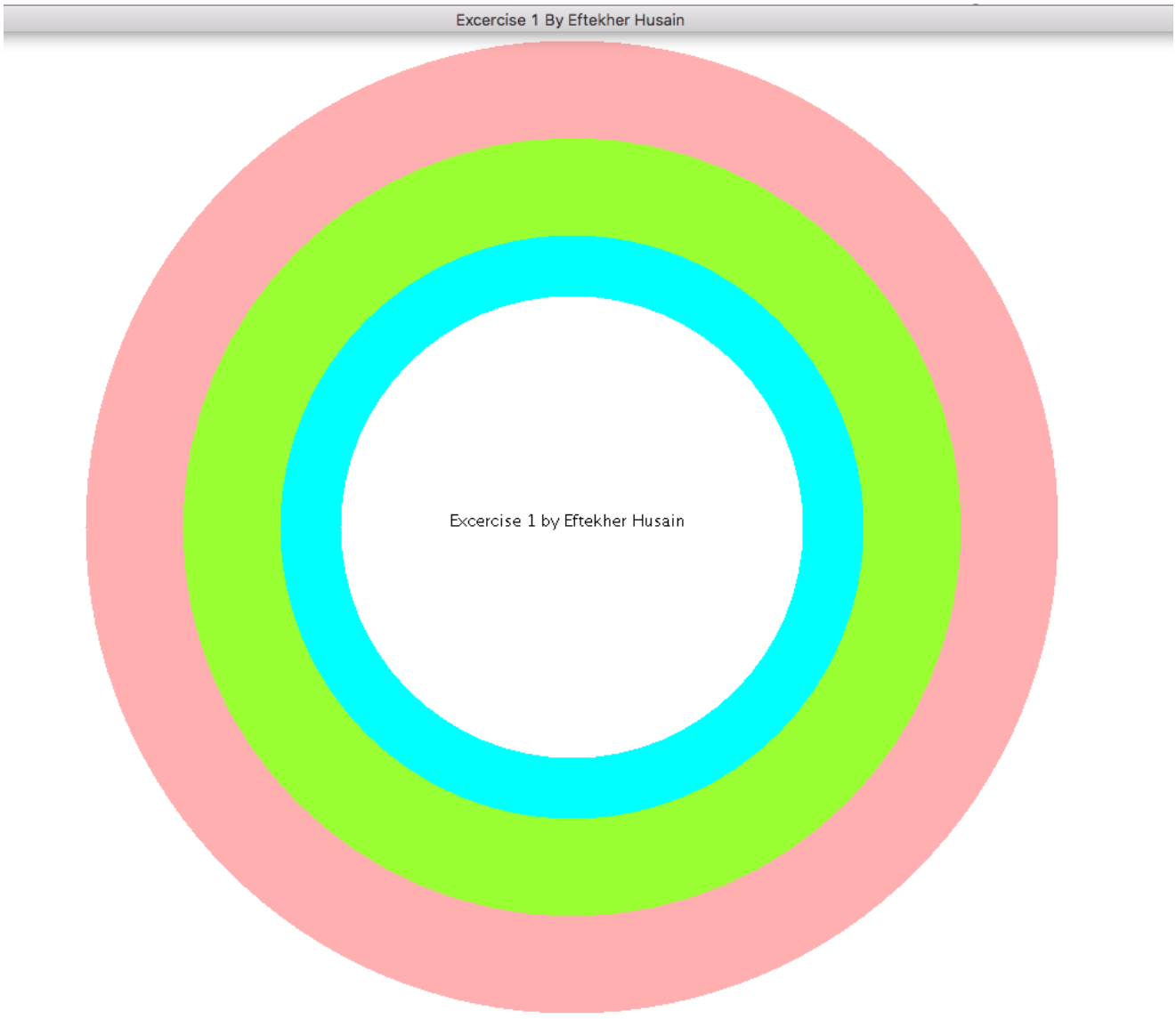
```
46          this.y = y; 47 }
48          public void setDx(int dx) {
49          this.dx = dx; 50          }
51          public void setDy(int dy) {
52          this.dy = dy; 53          }
54          public void setColor(Color color) {
55          this.color = color; 56 }
57
58          //moves point (x, y) by (Δx, Δy);
59          public void shiftXY(int dx, int dy) 60    {
61                  x += dx;
62          y =+ dy; 63    }
64
65          //returns the object's description as a String
66          public String toString()
67          {         String shape = "Shape[ x =" + getX() + ",y = " + getY() +"color
                    = " + getColor();
68          return shape; 69          }
70
69          // draw method to be overridden in it's subclasses

72          public abstract void draw(Graphics g);
73
74      }
75
```

Circle class All Codes

```
1 importjava.awt.Color;
5
6 public class Circle extendsShape 7{
8       private int radius; //rad. of circle 9
10      //circle constructor
11      public Circle(int x, int y, Color color, int radius) {
12              super(x, y, color);
13              this.radius = radius;
14              setX(x);
15              setY(y);
16      setColor(color); 17     }
18
19      //getters
20      public int getRadius() {
21      return this.radius; 22}
23
24      //returns the area of circle
25      public double getArea()
26      {return Math.PI * radius * radius;} 27
28      //returns the perimeter of circle
29      public double getPerimeter()
30      {return 2 * Math.PI * radius;} 31
32
33      public String toString()
34      {       String result = "Circle[ Radius=" +getRadius()+
35      ", Perimeter=" + getPerimeter()+ ", Area = " +getArea()+ "]";
36      return result;
37      }
38
39  @Override
40      public void draw(Graphics g) 41         {
42  g.setColor(getColor());
43  g.fillOval( getX()-getRadius(), getY()-getRadius() ,getRadius()* 2, getRadius() * 2);
```

```
44
45          }
46
47}
```

Final Output of the Circle Class in ShapeTest Class



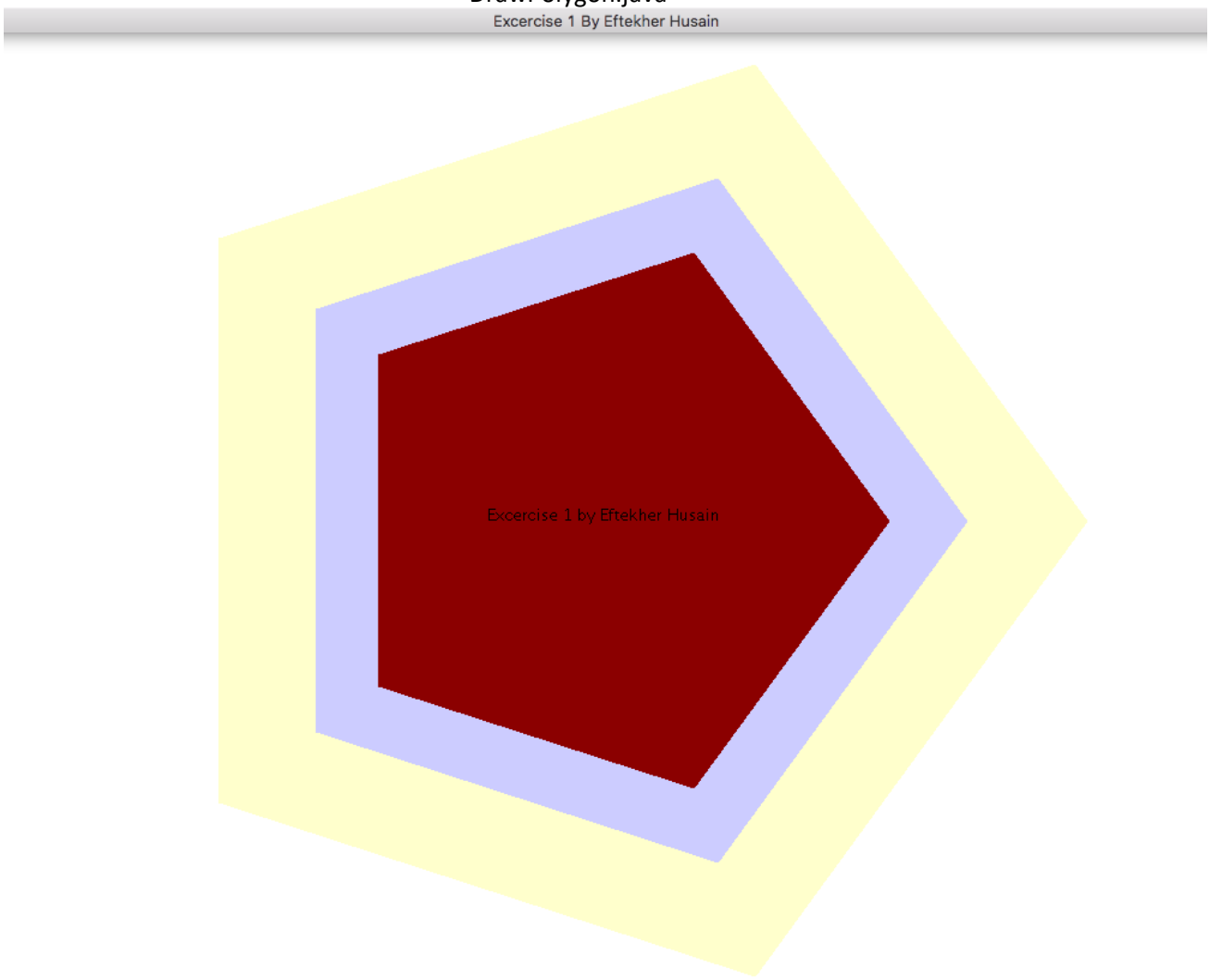
Polygon class All Codes

```
1 importjava.awt.Color;
6
7public          class DrawPolygon extends Shape 8{
9          private int N;
10         private int radius; //rad. of circle 11
12         //polygon Constructor
13         public DrawPolygon(int x, int y, Color color, int N,int radius) {
14                 super(x, y, color);
15                 this.N = N;
16         this.radius = radius; 17          }
18         //getters
19         public int getRadius() {
20         return radius; 21       }
22
23         //setters
24         public void setRadius(int radius) {
25         this.radius = radius; 26          }
27
28
29         //returns the area of Polygon
30         public double getArea() 31    {
32 double area = (1/4) * N * ((getSide() * getSide())/ (Math.tan(Math.PI/N)));
33         return area; 34          }
35
36
37         //returns the Perimeter of Polygon
38         public int getPerimeter()
39         {return N * getSide();} 40
41
42         //returns the interior angle (in degrees) of the Polygon
43         public double getAngle()
44         {       double angleInterior = (N-2) * 180/N;
```

```
45          return angleInterior; 46        }
47
48
49
50          public int getSide() 51         {
52                  int sideLength = (int) (2 * getRadius() * Math.PI/N);
53          return sideLength; 54           }
55
56
57          public String toString() {
58          String result = "( side length = " + getSide() +
59                          ", interior angle = " + getAngle()                  + ", perimeter = " +
60                          getPerimeter() + ", area = " + getArea() + ")";
61          return result; 62       }
63
64          @Override
65          public void draw(Graphics g) 66         {
67                  Polygon p = new Polygon(); 68
69
70                  for (int i = 0; i < N; i++)
71                  {
72  //p.addPoint(x, y);
73  p.addPoint((int) (getX() + getRadius() * Math.cos(i * 2 * Math.PI / N)),
74  (int) (getY()- getRadius() * Math.sin(i * 2 * Math.PI / N)));
75                  }
76
77                    g.setColor(getColor());
78                    g.fillPolygon(p);
79          }
80
81}
82
```

Final Output of the Polygon Class in ShapeTest Class

DrawPolygon.java



All Shapes are Tested in DrawShape class (All Codes)

```
 1 importjava.awt.BorderLayout;
 9
10  public class DrawShapeTest extends Canvas {
11          public DrawShapeTest() {
12          setBackground(Color.WHITE); 13        }
14          public void paint(Graphics g) {
15                  int centerX = getWidth()/2;
16                  int centerY = getHeight()/2; 17
18  Shape c1 = new                          Circle(centerX, centerY, Color.PINK, 400);
19  Shape c2 = new Circle(centerX, centerY, Color.decode("#99FF33"), 320);
20  Shape c3 = new Circle(centerX, centerY, Color.CYAN, 240);
21                  Shape c4 = new Circle(centerX, centerY, Color.WHITE, 190); 22
23  Shape p1 = new DrawPolygon(centerX, centerY, Color.decode("#FFFFCC"),
    5,400);
24  Shape p2 = new DrawPolygon(centerX, centerY, Color.decode("#CCCCFF"),
    5,300);
25  Shape p3 = new DrawPolygon(centerX, centerY, Color.decode("#8B0000"),
    5,235);
26
27                  c1.draw(g);
28                  p1.draw(g);
29                  c2.draw(g);
30                  p2.draw(g);
31                  c3.draw(g);
32                  p3.draw(g);
33                  c4.draw(g); 34
35  g.setColor(Color.BLACK);
36  g.drawString("Excercise 1 by Eftekher Husain", centerX-100, centerY);
37
38
39                  int width = getWidth();
40                  int height = getHeight();
41                  g.setColor(Color.BLACK);
42                  g.drawRect(220, 50, 1000, 800);
43                  g.drawLine(220, 50, 1220, 850);
```

```
44              g.drawLine(220, 850, 1220, 50);
45
46          }
47
48          public static void main(String[] args) { 49
50
51              JFrame frame = new JFrame("Excercise 1 By Eftekher Husain");
52              DrawShapeTest drawShapes = new DrawShapeTest(); 53
54
55              frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); 56
57              frame.add(drawShapes); 58
59              frame.setSize(1000, 1000);
60              frame.setVisible(true); 61          }
62}
63
```

Final Output