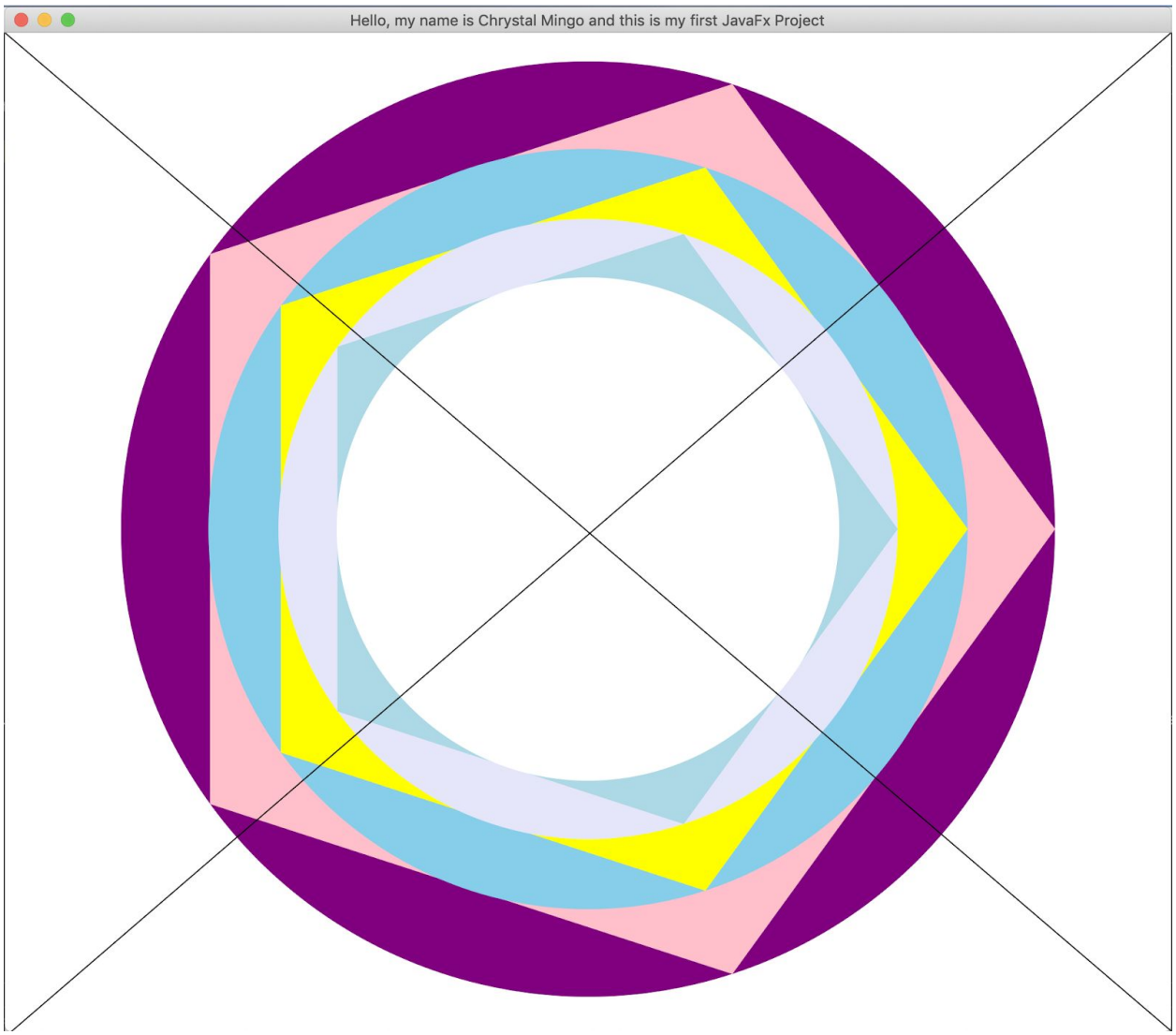
Chrystal Mingo
CSC 22100
Professor Auda

Here is my beautiful output:



This is my xxxShape class:

```
package sample;

import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;
public abstract class xxxShape {
  private double x;
  private double y;
  private Color color;

  public xxxShape(double x, double y, Color color) {
    // TODO Auto-generated constructor stuff
    this.x = x;
    this.y = y;
    this.color = color;
  }
  //setX, setY, setColor – set the point (x, y) and color for the xxxShape object;
  public void setX(double X) {
    this.x = X;
  }
  public void setY(double Y) {
    this.y = Y;
  }
  public void setColor(Color color) {
    this.color = color;
  }
  //getX, getY, getColor – return the point (x, y) and color of the xxxShape object;
  public double getX(){
    return this.x;
  }
  public double getY(){
    return this.y;
  }
  public Color getColor(){
    return this.color;
  }
  //toString() returns the object's description as a String
  public abstract String toString();
  //draws on canvas
  public abstract void draw(GraphicsContext gc);

}
```

This is my xxxLine class:

```
package sample;

import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;
public class xxxline extends xxxShape {
  private double x1;
  private double y1;
  private Color color;
  private double x2;
  private double y2;


  //Set values in constructor;
  public xxxline(double x1, double y1, Color color, double x2, double y2) {
     super(x1, y1, color);
     this.x1 = x1;
     this.y1 = y1;
     this.x2 = x2;
     this.y2 = y2;
     this.color = color;
  }

  //Need a setter for each x1; y1; x2; y2; and color
  public void setX1(double X1) {
     this.x1 = X1;
  }

  public void setY1(double Y1) {
     this.y1 = Y1;
  }

  public void setX2(double X2) {
     this.x2 = X2;
  }

  public void setY2(double Y2) {
     this.y2 = Y2;
  }

  public void setColor(Color color) {
     this.color = color;
  }
  //After setting it is necessary to implement a getter for each x1; y1; x2; y2; and color
  public double getX1() {
     return this.x1;
  }
```

```
  public double getY1() {
     return this.y1;
  }

  public double getX2() {
     return this.x2;
  }

  public double getY2() {
     return this.y2;
  }

  public Color getColor() {
     return this.color;
  }

  //I implemented this function called getDistance() to calculate the length, that would be called in the toString()
function.
  public double getDistance() { //calculating distance = length
     double length;
     length = Math.sqrt(Math.pow((x2 - x1), 2) + Math.pow((y2 - y1), 2)); //distance formula
     return length;
  }
  //I implemented this function called getAngle() to calculate the angle of the line, that would be called in the
toString() function.
  public double getAngle() {
     double angle;
     angle = (y2 - y1) * (x2 - x1); //calculations for angle --> formula is the same as slope
     return angle;
  }
  //toString — returns a string representation of the xxxLine object: length and angle with the x-axis;
  public String toString(){ //put length and angle into toString
     return "Length :" + getDistance() + " Angle :" + getAngle();
  }
  //draw — draws a xxxLine object [(x1, y1), (x2, y2)].
  public void draw(GraphicsContext gc){ //fill body with the necessary code to draw a line
     gc.setStroke(color);
     gc.strokeLine(x1, y1, x2, y2);
  }
}
```

This is my xxxCircle class:

```
package sample;

import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;

public class xxxCircle extends xxxShape {
  private double radius;
  private double area; //need for toString();
  private double perimeter;
  private double x;
  private double y;
  private Color color;

  public xxxCircle(double r, double x, double y, Color color){
     // TODO Auto-generated constructor stub
     super(x,y,color);
     this.radius = r;
     this.x = x;
     this.y = y;
     this.color = color;
  }
  //getRadius — returns the radius of the xxxCircle object;
  //setRadius — sets the radius of the xxxCircle object;
  public void setRadius(double r) {
     this.radius = r;
  }

  public double getRadius() {
     return this.radius;
  }
  //I made a setter and getter function to collect the area and perimeter needed for the toString() function
  public void setArea(double a) {
     this.area = a;
  }
  public void setPerimeter(double p) {
     this.perimeter = p;
  }
  public double getArea() {
     area = Math.PI*radius*radius; //formula of area of a circle
     return area;
  }
  public double getPerimeter() {//formula for the perimeter of a circle
     perimeter = 2*Math.PI*radius;
     return perimeter;
  }
  //returns a string representation of the xxxCircle object: radius,
```

```
  //perimeter, and area;
  public String toString(){
     return "Radius :" + getRadius() + " Area :" + getArea() + " Perimeter :" +getPerimeter();
  }
  //draws a xxxCircle object of radius radius. The center point of the circle is
  //defined in class xxxShape.
  public void draw(GraphicsContext gc){
     gc.setFill(color);
     gc.setStroke(color);
     gc.strokeOval(x, y, radius, radius);
     gc.fillOval(x, y, radius, radius);

  }
}
```

This is my xxxPolygon class:

```
package sample;


import javafx.scene.paint.Color;
//import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;

public class xxxPolygon extends xxxShape {

  int n; //number of sides
  double sideLength;
  double x;
  double y;
  Color color;
  double radius;

  public xxxPolygon(double x, double y, Color color, double radius, int n) {

     super(x, y, color);

     this.radius = radius;
     this.n = n;
     this.x = x;
     this.y = y;
     this.color = color;
  }

  public double perimeter() {

     return this.n * this.sideLength;
```

```
}

public double radius() {

    double radii = sideLength / (2 * Math.sin(Math.PI / n));

    return radii;

}

public double apothem() {
    double apothem = (this.sideLength) / (2 * Math.tan(180.0 / this.n));

    return apothem;
}

public double area() {

    double area = (perimeter() * apothem()) / 2;

    return area;
}
//One of the biggest challenges was the Polygon, setting up an array that collects x and y points.
//but the hardest part was getting the correct formulas
public double[] xarray() {
    double[] xpoints = new double[n];

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

        xpoints[i] = x + radius * Math.cos(2 * Math.PI * i / n);

    }
    return xpoints;
}

public double[] yarray() {
    double[] ypoints = new double[n];

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

        ypoints[i] = y + radius * Math.sin(2 * Math.PI * i / n);

    }
    return ypoints;
}
```

```
  public double interiorAngles() {
     return (this.n - 2) / 180;
  }
  //returns a string representation of the xxxPolygon object: side length, interior angle, perimeter, and area;
  public String toString() {

     return "Side Length: " + this.n + " Area: " + area() + " Perimeter: " + perimeter() + " Interior Angles: " +
interiorAngles();

  }
  //draws a xxxPolygon object and inscribed in a circle of radius radius. The center point of the circle is defined in
class xxxShape.
  public void draw(GraphicsContext gc) {
     gc.setFill(color);
     gc.setStroke(color);
     gc.fillPolygon(xarray(), yarray(), n);
     gc.strokePolygon(xarray(), yarray(), n);

  }
}
```

This is my Main class:

```
//Chrystal Mingo
//Project One
//CSC 22100
package sample;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Group;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;

public class Main extends Application {

  @Override
  public void start(Stage primaryStage) throws Exception{
     Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
     primaryStage.setTitle("Hello, my name is Chrystal Mingo and this is my first JavaFx Project");
     primaryStage.setScene(new Scene(root, 300, 275));

     //creating my circle objects
     xxxCircle circle = new xxxCircle(800, 0 + 100, 0 + 25 , Color.PURPLE);
     xxxCircle circle2 = new xxxCircle(650, 75 + 100, 75 + 25, Color.SKYBLUE);
```

```
        xxxCircle circle3 = new xxxCircle(530, 135 + 100, 135 + 25, Color.LAVENDER);
        xxxCircle circle4 = new xxxCircle(430, 185 + 100, 185 + 25, Color.WHITE);

        //creating my polygon objects
        xxxPolygon Poly = new xxxPolygon(400 + 100, 400 + 25, Color.PINK, 400,5);
        xxxPolygon Poly2 = new xxxPolygon(400 + 100, 400 + 25, Color.YELLOW, 325,5);
        xxxPolygon Poly3 = new xxxPolygon(400 + 100, 400 + 25, Color.LIGHTBLUE, 265,5);

        //creating my line objects
        xxxline line = new xxxline(0,0, Color.BLACK, 1170,1000);
        xxxline line2 = new xxxline(0,860, Color.BLACK, 1000,0);


        Group group = new Group();
        Canvas canvas = new Canvas(1000, 1000);
        GraphicsContext gc = canvas.getGraphicsContext2D();
        circle.draw(gc);
        Poly.draw(gc);
        circle2.draw(gc);
        Poly2.draw(gc);
        circle3.draw(gc);
        Poly3.draw(gc);
        circle4.draw(gc);
        line.draw(gc);
        line2.draw(gc);


        group.getChildren().add(canvas);

        Scene circScene = new Scene(group,1000, 1000);

        primaryStage.setScene(circScene);

        primaryStage.show();
    }


    public static void main(String[] args) {
        launch(args);
    }
}
```

Conclusion:

This project was fun to me, it was challenging but still attainable. I never coded in java prior to this class, and now I feel pretty confident in programming with this language. My biggest challenge was in actually getting my code to run, JavaFx wasn't being imported on my computer. I downloaded at first Java (version 12), and was using eclipse to do my sides projects coding in java. However, eclipse was not working when I was doing the project. Students recommended me to download Java(version 8) and to not use eclipse. I downloaded the older version of Java and jGrasp, never heard of it but a student recommended it and still no progress. Then I moved onto IntelliJ, and downloaded as well the JavaFx module, and finally got some results. Even though I struggled with getting JavaFx, I feel like the experience exposed me to a lot of new IDEs I've never heard about. So it was a learning experience in the end.

The hardest part programming wise when it came to the project was the draw() function, at first I didn't include parameters which was GraphicsContent gc into the abstract, which was making it hard to overwrite and implement the draw() function in xxxLine, xxxCircle, and xxxPolygon. I found implementing the code pretty easy for line and circle, but the array of the x and y points were a challenge since there was a specific formula needed as well. However, I overcome these obstacles by using java documentation and learning about the parameters

I love how cute mine came out!