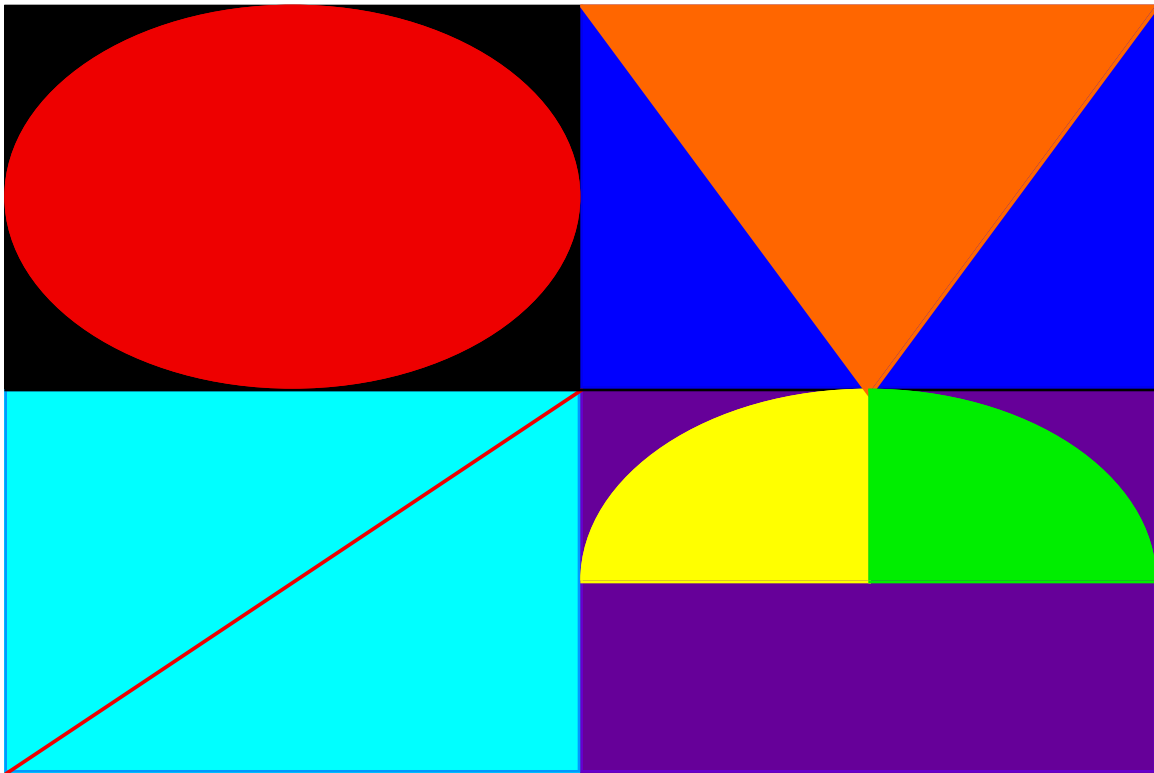


Goal:



Description:

This picture is made out of 4 equal size rectangles.

- The upper left hand rectangle is black, with a red filled circle on the inside.
- The lower left hand rectangle is cyan, with a red line cutting across one of the diagonals.
- The upper right hand rectangle is blue, with an orange triangle (hint: polygon) on the inside.
- The lower right hand rectangle is magenta with two quarter ovals (think filled arcs) - one that is yellow, and another that is green.

Suggestions:

This lab tests your ability to break down the different components that I want to be drawn in very precise mathematics. Essentially, don't just guess and check. Keep in mind that your picture will need to **STRETCH** as your program stretches - that means you will need to make use of `getWidth()` and `getHeight()` (don't use number like 500 and 300 - they'll change when your program's window is stretched). As a reminder, here are all the Graphics commands that you can use:

- `int width = getWidth();` gets the total width of the drawing area
- `int height = getHeight();` gets the total height of the drawing area

MORE ON BACK

- **g.setColor(Color.x)**; changes the paint brush color to x
- **g.drawLine(x1, y1, x2, y2)**; draws a line from (x1, y1) to (x2, y2)
- **g.fillRect(x, y, w, h)**; draw a rectangle that has an upper left hand corner of (x,y), a width of w, and a height of h
- **g.fillOval(x, y, w, h)**; fills an oval that is bounded by a rectangle given by g.drawRect(x, y, w, h) or (fills an ellipse that has a center at (x + w/2, y + h/2), a horizontal axis of w, and a vertical axis of h)
- **Polygon p = new Polygon()**; creates a polygon (no drawing)
- **p.addPoint(x, y)**; adds a point with coordinates (x,y) to the polygon (no drawing)
- **g.fillPolygon(p)**; fills the polygon
- **g.fillArc(x, y, w, h, startAngle, arcMeasure)**; fills an arc representing a portion of an oval represented by g.drawOval(x, y, w, h) - the portion is from angle (degrees) startAngle and goes arcMeasure amount of degrees

Remember also that you can draw on top of old drawing and it will be covered up (just like real paint). Also, each of these commands only work on integer data - **no decimals!** Many of these commands also have draw versions - like drawRect, drawOval, drawArc, drawPolygon, etc.

Getting Started:

Create a new java project in Eclipse called **DrawingLab**. Download the two files called DrawingLabFrame.java and DrawingLabPanel.java and put the java files in your src folder. All of your code needs to go inside the DrawingLabPanel, in the paint block of code. Attack the lab step by step, and comment your code as you go - do not wait till the end!

```
//Example:

public void paint(Graphics g)
{
    /*YOUR CODE STARTS HERE*/

    g.setColor(Color.BLUE);

    //...

    /* YOUR CODE STOPS HERE */
}
```