

Graphics methods review

Method name	Description
<code>drawLine (x1, y1, x2, y2)</code>	line between points $(x1, y1)$, $(x2, y2)$
<code>drawOval (x, y, width, height)</code>	draws outline of largest oval that fits in a box of size $width * height$ with top-left corner at (x, y)
<code>drawRect (x, y, width, height)</code>	draws outline of rectangle of size $width * height$ with top-left corner at (x, y)
<code>drawString (text, x, y)</code>	writes text with bottom-left corner at (x, y)
<code>fillOval (x, y, width, height)</code>	fills largest oval that fits in a box of size $width * height$ with top-left corner at (x, y)
<code>fillRect (x, y, width, height)</code>	fills rectangle of size $width * height$ with top-left corner at (x, y)
<code>setColor (Color)</code>	Sets <code>Graphics</code> to paint subsequent shapes in the given color

Superimposing shapes

- Drawing one shape on top of another causes the last shape to appear on top of the previous one(s).

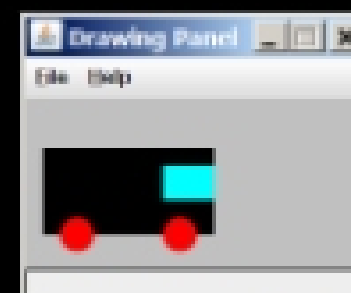
```
import java.awt.*;

public class DrawCar {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(200, 100);
        panel.setBackground(Color.LIGHT_GRAY);
        Graphics g = panel.getGraphics();

        g.setColor(Color.BLACK);
        g.fillRect(10, 30, 100, 30);

        g.setColor(Color.RED);
        g.fillOval(20, 70, 20, 20);
        g.fillOval(80, 70, 20, 20);

        g.setColor(Color.CYAN);
        g.fillRect(80, 40, 30, 20);
    }
}
```

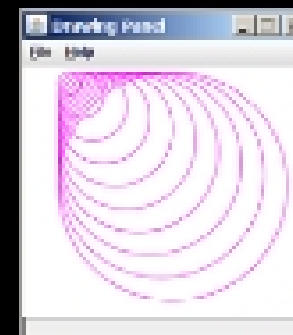


Loops to change shape's size

A `for` loop can also vary a shape's size:

```
import java.awt.*;

public class DrawCircles {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(250, 220);
        Graphics g = panel.getGraphics();
        g.setColor(Color.MAGENTA);
        for (int i = 1; i <= 10; i++) {
            g.drawOval(30, 5, 20 * i, 20 * i);
        }
    }
}
```



CSD1 191

Adapted from: "Building Java Programs": Riggs & Stepp

Dr. Nazli Haridi

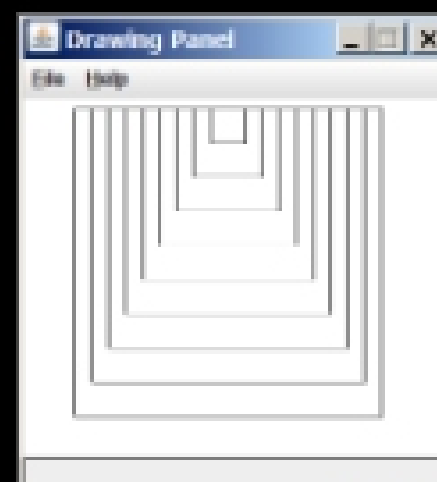
A loop that varies both

The loop in this program affects both the size and shape of the figures being drawn.

Each pass of the loop, the square drawn becomes 20 pixels smaller in size, and shifts 10 pixels to the right.

```
import java.awt.Color;
import java.awt.Graphics;

public class DrawSquares {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(250, 220);
        Graphics g = panel.getGraphics();
        g.setColor(Color.ORANGE);
        for (int i = 1; i <= 10; i++) {
            g.drawRect(20 + 10 * i, 5,
                200 - 20 * i, 200 - 20 * i);
        }
    }
}
```



CSD1 191

Adapted from: "Building Java Programs": Riggs & Stepp

Dr. Nazli Haridi

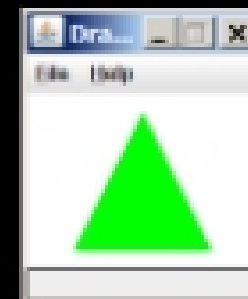
Drawing polygons

- Polygon objects represent arbitrary shapes.
 - Add points to a Polygon using its `addPoint(x, y)` method.

```
import java.awt.*;

public class DrawTriangle {
    public static void main(String[] args) {
        DrawingPanel p = new DrawingPanel(100, 100);
        Graphics g = p.getGraphics();

        Polygon poly = new Polygon();
        poly.addPoint(10, 80);
        poly.addPoint(50, 10);
        poly.addPoint(90, 80);
        g.setColor(Color.GREEN);
        g.fillPolygon(poly);
    }
}
```



Drawing parameter solution

```
import java.awt.*;

public class DrawingWithParameters {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(210, 100);
        panel.setBackground(Color.BLACK);

        Graphics g = panel.getGraphics();
        drawCar(g, 10, 30, 100);
        drawCar(g, 130, 10, 50);
    }

    public static void drawCar(Graphics g, int x, int y, int size) {
        g.setColor(Color.ORANGE);
        g.fillRect(x, y, size, size / 2);

        g.setColor(Color.GREEN);
        g.fillOval(x + size / 10, y + 2 * size / 3,
                 size / 3, size / 3);
        g.fillOval(x + 7 * size / 10, y + 2 * size / 3,
                 size / 3, size / 3);

        g.setColor(Color.CYAN);
        g.fillRect(x + 7 * size / 10, y + size / 10,
                 2 * size / 10, size / 3);
    }
}
```

