# Draw a rainbow

1. Use the ideas in the Bulleye program (below, you can copy/paste) to draw a rainbow. Unlike the Bullseye, the rainbow should drac arcs instead of ovals. The method to draw an arc is described in the box on the next page; instead of fillOval, it’s fillArc.
2. The rainbow should have at least 20 arcs. Instead of only using two colors, use many colors. This is left for a problem for you to solve: how to change the color as the arcs progress. Here’s the way to make a color:

Color c = new Color(red, green, blue);

(where each component of color is between 0 and 255)

**import** java.applet.Applet;

**import** java.awt.\*;

**public** **class** Bullseye **extends** Applet {

**public** **void** paint(Graphics page) {

**final** **int** MAX\_WIDTH = 300, NUM\_RINGS = 5, RING\_WIDTH = 25;

**int** x = 0, y = 0, diameter;

setBackground(Color.*CYAN*);

diameter = MAX\_WIDTH;

page.setColor(Color.*WHITE*);

**for** (**int** i = 0; i < NUM\_RINGS; i++) {

**if** (page.getColor() == Color.*BLACK*)

page.setColor(Color.*WHITE*);

**else**

page.setColor(Color.*BLACK*);

page.fillOval(x, y, diameter, diameter);

diameter -= (2 \* RING\_WIDTH);

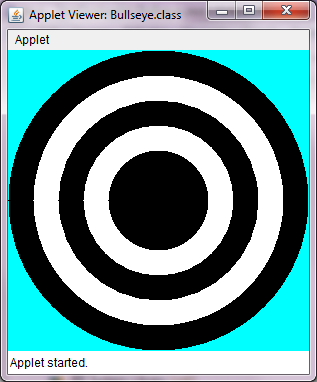
x += RING\_WIDTH;

y += RING\_WIDTH;

}

}

}



### fillArc

public abstract void **fillArc**(int x, int y, int width, int height, int startAngle,int arcAngle)

Fills a circular or elliptical arc covering the specified rectangle.

The resulting arc begins at startAngle and extends for arcAngle degrees. Angles are interpreted such that 0 degrees is at the 3 o'clock position. A positive value indicates a counter-clockwise rotation while a negative value indicates a clockwise rotation.

The center of the arc is the center of the rectangle whose origin is (*x*, *y*) and whose size is specified by the width and height arguments.

The resulting arc covers an area width + 1 pixels wide by height + 1 pixels tall.

The angles are specified relative to the non-square extents of the bounding rectangle such that 45 degrees always falls on the line from the center of the ellipse to the upper right corner of the bounding rectangle. As a result, if the bounding rectangle is noticeably longer in one axis than the other, the angles to the start and end of the arc segment will be skewed farther along the longer axis of the bounds.

**Parameters:**

x - the *x* coordinate of the upper-left corner of the arc to be filled.

y - the *y* coordinate of the upper-left corner of the arc to be filled.

width - the width of the arc to be filled.

height - the height of the arc to be filled.

startAngle - the beginning angle.

arcAngle - the angular extent of the arc, relative to the start angle