In the spaces below, fill in some code for the `Emoticon` class, following guidelines given through comments. The class has attributes corresponding to the state and the x, y coordinates for positioning the `Emoticon` in a Graphics context.

For simplicity, we only represent 3 states, encoded by numbers:
0 = sad, 1 = happy, 2 = wink

```java
public class Emoticon {
    // instance variables
    private int x, y;
    private int state;

    // constructor: sets the values of x, y and state
    public Emoticon(int a, int b, int s) {
        x = a;
        y = b;
        state = s;
    }

    // toString: returns a String representing the state, e.g., :-) for happy state
    public String toString() {
        String message = "";
        if (state == 0) message = ":-(";
        else if (state == 1) message = ":-)";
        else message = ";-(";
        return message;
    }
}
```

(Do NOT write the draw() method, but assume it will also be implemented as part of this class. See back of this page for Circle.java class to use as reminder.)
public class Circle
{
    private int diameter, x, y;
    private Color color;

    // Constructor: Sets up this circle with the specified values.
    public Circle(int size, Color shade, int upperX, int upperY)
    {
        diameter = size;
        color = shade;
        x = upperX;
        y = upperY;
    }

    // Draws this circle in the specified graphics context.
    public void draw(Graphics page)
    {
        page.setColor(color);
        page.fillOval(x, y, diameter, diameter);
    }

    // Color mutator.
    public void setColor(Color shade)
    {
        color = shade;
    }

    // Color accessor.
    public Color getColor()
    {
        return color;
    }

    // toString
    public String toString()
    {
        return "O";
    }
}