Use constants
COMP 102

Victoria University of Wellington

Move or resize the Lollipop.

```java
import ecs100.*;
import java.awt.Color ;

/** Draws little pictures on the graphics pane */
public class  Drawer {

    /** Draw a lollipop */
    public void drawLollipop() {
        UI.setColor(Color.black);         // set color to black
        UI.setLineWidth(10);              // set line width to 10
        UI.drawLine(300, 200, 300, 400);  // draw line
        UI.setLineWidth(1);               // set line width back to 1
        UI.setColor(Color.red);           // set color to red
        UI.fillOval(260, 160, 80, 80);    // draw blob
    }
}
```
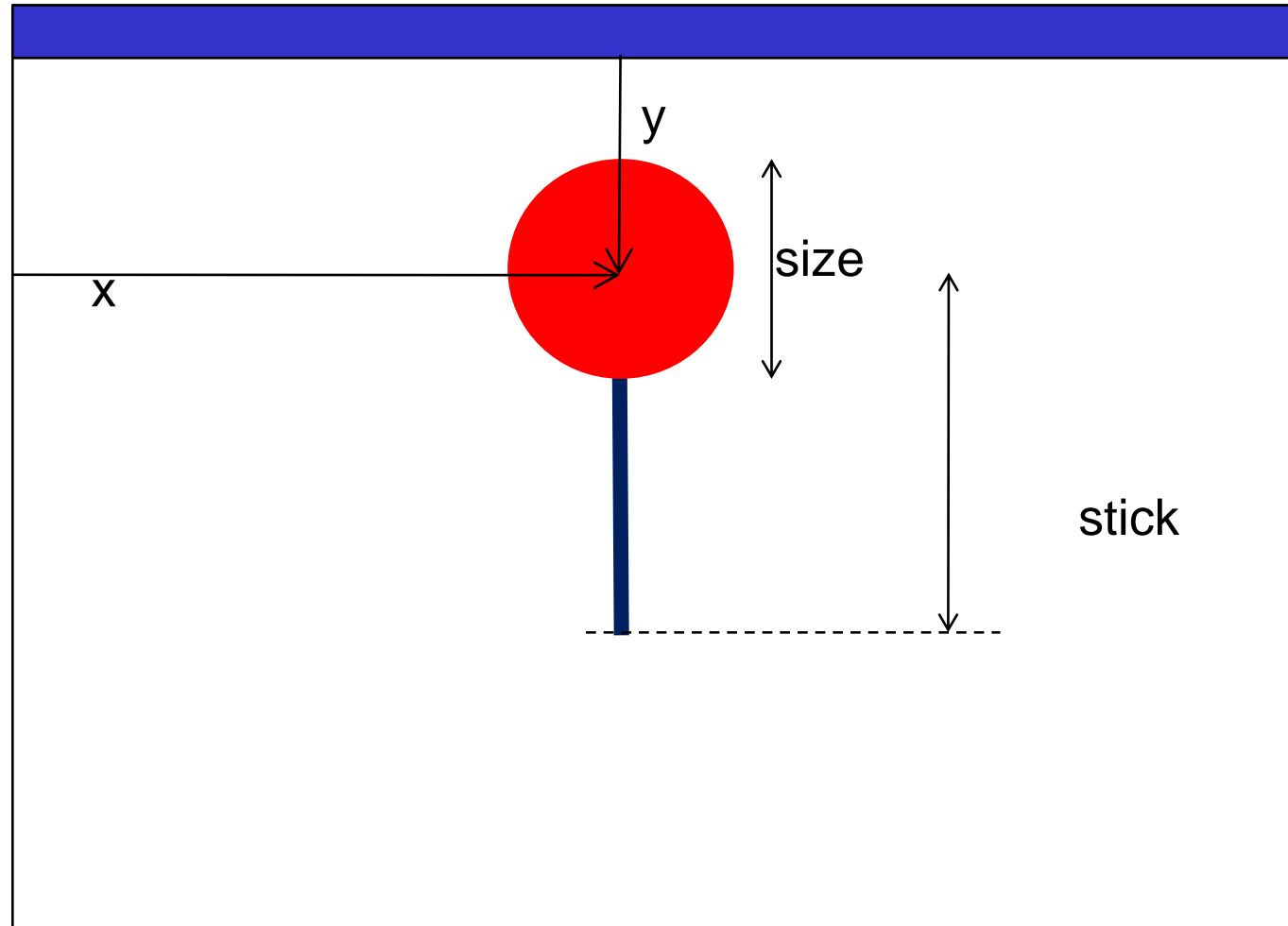
Move it left

Move it down

Change blob size

Improving the design

- Better design: Use named constants and variables
    - ⇒ easier to write and easier to change
    - ⇒ get the computer to do the arithmetic

- Use named constants for values that won't change while the program is running.

Values to specify lollipop & stick

Improving the program: constants

```java
import  ecs100.*;     import  java.awt.Color;

/** Draw a lollipop with a stick */
public class  Drawer {

    public static final double  X = 300.0;          // horizontal center of lollipop
    public static final double  Y = 180.0;          // vertical center of lollipop
    public static final double  SIZE = 80.0;        // diameter of lollipop
    public static final double  STICK =  200.0;     // length of lollipop stick


    /** Draw a lollipop */
    public void drawLollipop() {
        UI.setLineWidth(SIZE/8.0);
        UI.drawLine(X, Y, X, Y+STICK);
        UI.setLineWidth(1);
        UI.setColor(Color.red);
        UI.fillOval(X-SIZE/2.0,  Y-SIZE/2.0,  SIZE, SIZE);
    }
}
```
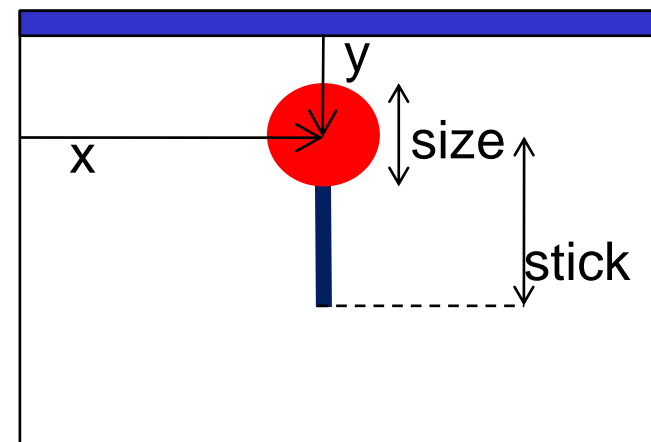
Easy to change: one place!

Syntax rules: Program structure

- 2nd version

⟨**import** statements⟩

public → **class** → ⟨classname⟩ → **{**

⟨constant definition⟩

⟨method description⟩

}

Constant definition

public → **static** → **final**

⟨type⟩ → ⟨name⟩ → **=** → ⟨value⟩ → **;**