“When the computer crashed during the execution of your program, there was no hiding. Lights would be flashing, bells would be ringing and everyone… would come running to find out whose program was doing something bad to the system.”

Margaret Hamilton
Director of Apollo Flight Computer Programming
To create an empty window:

```java
DrawingPanel panel = new DrawingPanel(300, 200);
```
DrawingPanel panel = new DrawingPanel(300, 200);

Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);

Draws a rectangle, but.... huh?
Graphics

DrawingPanel panel = new DrawingPanel(300, 200);

Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);

Top left corner is (0, 0)
DrawingPanel panel = new DrawingPanel(300, 200);

Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);
DrawingPanel panel = new DrawingPanel(300, 200);

Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);

Top left corner of rectangle is... 10 pixels to the right, and...
Graphics

DrawingPanel panel = new DrawingPanel(300, 200);

Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);

Top left corner of rectangle is...

…10 pixels to the right, and...

…30 pixels down.
Graphics

DrawingPanel panel = new DrawingPanel(300, 200);

Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);

(0, 0) 30
10 60 pixels wide
Graphics

DrawingPanel panel = new DrawingPanel(300, 200);

Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);
Coordinate system

- Each (x, y) position is a *pixel* ("picture element").

- Position (0, 0) is at the window's top-left corner.
  - x increases rightward and the y increases *downward*.

- The rectangle from (0, 0) to (200, 100) looks like this:
Graphics

DrawingPanel panel = new DrawingPanel(300, 200);
Graphics g = panel.getGraphics();
g.fillRect(10, 30, 60, 35);
g.fillOval(80, 40, 50, 70);
Java class libraries, import

- **Packages**: code that Java developers already wrote

- To use a package, add an “import” line to your Java code:
  - import ......;

- For example, `Graphics` belongs to a package named `java.awt`

- To use `Graphics`, put this at the top of your program:
  ```java
  import java.awt.Graphics;
  ```
# Graphics methods

<table>
<thead>
<tr>
<th>Method name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>g.drawLine(x1, y1, x2, y2)</code>;</td>
<td>line between points ((x1, y1), (x2, y2))</td>
</tr>
<tr>
<td><code>g.drawOval(x, y, width, height)</code>;</td>
<td>outline largest oval that fits in a box of size (width \times height) with top-left at ((x, y))</td>
</tr>
<tr>
<td><code>g.drawRect(x, y, width, height)</code>;</td>
<td>outline of rectangle of size (width \times height) with top-left at ((x, y))</td>
</tr>
<tr>
<td><code>g.drawString(text, x, y)</code>;</td>
<td>text with bottom-left at ((x, y))</td>
</tr>
<tr>
<td><code>g.fillOval(x, y, width, height)</code>;</td>
<td>fill largest oval that fits in a box of size (width \times height) with top-left at ((x, y))</td>
</tr>
<tr>
<td><code>g.fillRect(x, y, width, height)</code>;</td>
<td>fill rectangle of size (width \times height) with top-left at ((x, y))</td>
</tr>
<tr>
<td><code>g.setColor(Color)</code>;</td>
<td>set Graphics to paint any following shapes in the given color</td>
</tr>
</tbody>
</table>
Clicker Question

- How many rectangles appear on the DrawingPanel when the following code is run?

```java
DrawingPanel p1 = new DrawingPanel(200, 200);
Graphics gr = new Graphics();
for(int i = 0; i < 5; i++) {
    gr.drawRect(i * 25, i * 20, 20, 50);
}
```

A. 5       B. 6       C. 20
D. None due to syntax error
E. None due to runtime error
Using colors

g.setColor(Color.BLACK);
g.fillRect(10, 30, 100, 50);
g.drawLine(20, 0, 10, 30);
g.setColor(Color.RED);
g.fillOval(60, 40, 40, 70);

Color brown = new Color(192, 128, 64);
panel.setBackground(brown);
Specified as predefined Color class constants:

```
Color.CONSTANT_NAME
```

where CONSTANT_NAME is one of:

- BLACK, BLUE, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW

Or create one using Red-Green-Blue (RGB) values of 0-255

```
Color name = new Color(red, green, blue);
```

- Example:
  
  ```
  Color brown = new Color(192, 128, 64);
  Color burntOrange = new Color(191, 87, 0);
  ```

List of Colors
What named color is closest to the Color object created by this code?

```java
Color mc = new Color(255, 255, 255);
```

A. Black
B. Brown
C. Gray
D. Orange
E. White
Outlined shapes

To draw a colored shape with an outline, first fill it, then draw the same shape in the outline color.

```java
import java.awt.Graphics;  // so I can use Graphics
import java.awt.Color;

public class OutlineExample {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(150, 70);
        Graphics g = panel.getGraphics();

        // inner red fill
        g.setColor(Color.RED);
        g.fillRect(20, 10, 100, 50);

        // black outline
        g.setColor(Color.BLACK);
        g.drawRect(20, 10, 100, 50);
    }
}
```
Superimposing shapes

- When $\geq 2$ shapes occupy the same pixels, the last drawn "wins."

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

public class Car {
    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, 50);

        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);
    }
}
```
Drawing with loops

- The $x, y, w, h$ expressions can use the loop counter variable:

```java
panel.setBackground(Color.YELLOW);
g.setColor(Color.RED);
for (int i = 1; i <= 10; i++) {
    // $x \ y \ w \ h$
    g.fillOval(100 + 20 * i, 5 + 20 * i, 50, 50);
}
```

- Nested loops can be used with graphics:

```java
g.setColor(Color.BLUE);
for (int x = 1; x <= 4; x++) {
    for (int y = 1; y <= 9; y++) {
        g.drawString("Java", x * 40, y * 25);
    }
}