Chapter 3 - graphics

- Drawing Panel
  - a window on the screen

* not apart of Java; provided by the authors

- Graphics
  - a "pen" to draw shapes and lines on a window

- Colors
  - colors in which to draw shapes

Drawing Panel → "canvas" objects that represents windows/drawing surfaces

- to create a window

```
DrawingPanel name = new DrawingPanel (width, height);
```

ex) panel

$\underbrace{\text{DrawingPanel}}_{\text{one word}}$

- The window has nothing on it
  - We can draw shapes / lines on it with another object of type Graphics

Graphics → "Pen" or "paint brush" objects to draw lines & shapes

- Access it by calling get Graphics on your Drawing Panel

```
Graphics g = panel.get Graphics ();
```

Chapter 3 cont
Graphics cont.

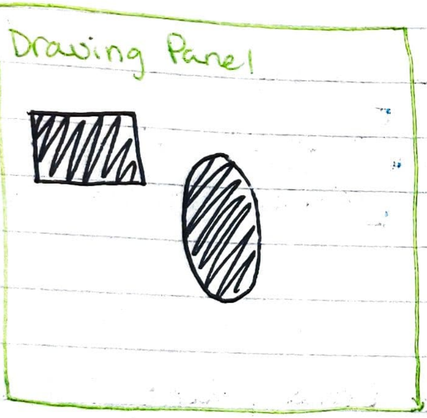
- Draw shapes by calling on methods on the Graphics object

from → Graphics g = panel.getGraphics();

to → g.fillRect(10, 30, 60, 35);
→ g.fillOval(80, 40, 50, 70);

$\underbrace{X, Y,}_{\text{starting position}}$ vs. $\underbrace{W, H}_{\text{width \& height}}$

ex)



Chapter 3 cont.

Java class libraries: classes included w/ Javas JDK

- organized into groups named packages
- to use a package, put an import declaration in your program:

```
// put this at very top of program

import packageName.*;
```

Graphics belongs to a package named java.awt

```
import java.awt.*;
```

- to use Graphics, you must place this above public class header

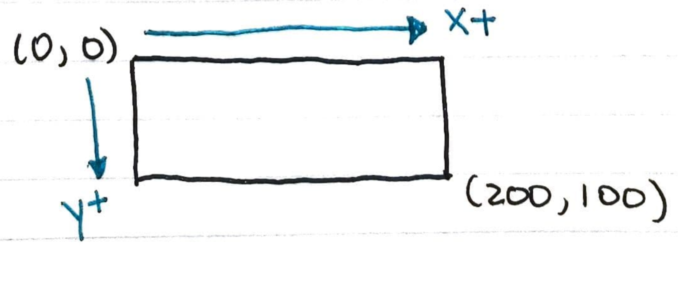
Coordinate System

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

od

- Position (0, 0) is the windows top left corner
  - x increases right ward
  - y increases downward

ex) A rectangle from (0, 0) to (200, 100)



Chapter 3 cont.

Graphics methods

g.drawLine(x1, y1, x2, y2);
- line between (x1, y1) & (x2, y2)

g.drawOval(x, y, width, height);
- outline largest oval that fits in a box size of width * height w/ top left at (x, y)

g.drawRect(x, y, width, height);
- outline of rectangle of size width * height w/ top left at (x, y)

g.drawString(text, x, y);
- text with bottom left at (x, y)

g.fillOval(x, y, width, height);
- fill largest oval that fits in a box of size width * height with top-left at (x, y)

g.fillRect(x, y, width, height);
- fill rectangle of size width * height with top-left at (x, y)

g.setColor(Color);
- set Graphics to paint any following shapes in the given color

Color

- 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 0-225

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

ex)
```
Color brown = new Color (192, 128, 64);
```

Using colors

- Pass a Color to Graphics objects set Color method
  - subsequent shapes will be drawn in a new color

ex)
```
g.set Color (Color.BLACK);       ] rectangle is black
g.fill Rect (10, 30, 100, 50);
g.draw Line (20, 0, 10, 30);
g.set Color (Color.RED);
g.fill Oval (60, 40, 40, 70);    ] Oval is filled
```

Using colors cont.

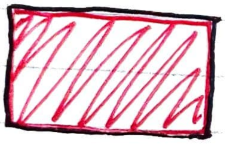
- Pass a color to DrawingPanel's set Background method

- The overall window background color will change

ex) Color brown = new Color (192, 128, 64);

panel.set Background (brown);

-- to outline a shape you must first fill it & then draw it



```
ex) import java.awt.*;

public class Outline_Example {
~~pub~~ psvm (String [] args) {

DrawingPanel panel = new DrawingPanel (150,70);
Graphics g = panel.get Graphics();
  // Red rectangle
 g.set Color (Color.RED);
 g.fill Rect (20, 10, 100, 50);
 // black outline
 g.set Color (Color.BLACK);
 g.draw Rect (20,10,100, 50);

 }
}
```