

```

public void paint(Graphics g)           // Permits "public" access to this part of the program.
                                        // "void" is the return type for this part of the program.
                                        // Until chapter 3, all return types are "void".
                                        // Names this part of the program "paint".
                                        // Declares that "paint" requires a "Graphics" object "g".

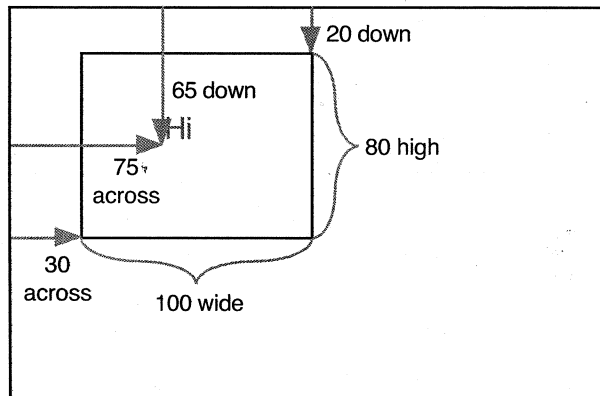
g.setColor(Color.black);                // These statements provide graphics instructions
g.drawRect(30, 20, 100, 80);            // At this point, you should consider modifying only
g.setColor(Color.red);                  // graphics instructions.
g.drawString("Hi", 75, 65);

```

Specifics of the graphics commands above

Colors are set first to black (for the rectangle) and then later to red (for the word "Hi").

Each number counts pixels (dots) on the screen measured from left to right or top to bottom.



The rectangle has its left side 30 pixels from the left of the screen; its top is 20 pixels from the top of the screen. The rectangle is 100 pixels wide and 80 pixels high.

The word "Hi" starts 75 pixels from the left of the screen and 65 pixels from the top of the screen. The positions of words are measured from the bottom of the letters because this is the traditional way typographers do it. The measurement is to the bottom of common letters — letters such as 'y' extend lower. Typographers call the position of the bottom of the common letters the *baseline*.

Important note:

Computers are extremely fussy about how things are spelled and punctuated. The program above has 10 periods, 5 semi-colons and 5 commas. The quotation marks around the word "Hi" are the double quotes generated by holding down the shift key and pressing the key to the right of the semi-colon key. Get one symbol wrong and the program will not work. Furthermore, the Java language is "case sensitive". This means that the proper use of upper case and lower case letters is critical.

Graphics commands

a. Commands

```

g.setColor(Color.red); // other colors are listed as constants below
g.setColor(new Color(0, 120, 0)); // red, green, blue (0...255 each)

g.setFont(new Font("SansSerif", Font.BOLD, 24)); // font, style (constants below), size in points.

g.drawString("Any Text", 50, 100); // place text in window at over 50, down 100

g.drawLine(30, 20, 120, 100); // xstart ystart xend yend

g.drawRect(30, 20, 100, 80); // left, top, width, height - outlined
g.fillRect(30, 20, 100, 80); // left, top, width, height - solid

g.drawRoundRect(30, 20, 100, 80, 10, 10); // left, top, width, height,
// arcWidth, arcHeight* - outlined
g.fillRoundRect(30, 20, 100, 80, 10, 10); // left, top, width, height,
// arcWidth, arcHeight* - solid

g.drawOval(30, 20, 100, 80); // left, top, width, height - outlined
g.fillOval(30, 20, 100, 80); // left, top, width, height - solid

g.drawArc(30, 20, 50, 80, 30, 60); // left, top, width, height,
// startAngle, extentAngle
// - outlines part of the oval described by
// -- left, top, width, height
// - startAngle is measured in degrees
// -- counter clockwise from the right (east)
// - extentAngle is measured in degrees
// -- counter clockwise from the startAngle

g.fillArc(30, 20, 50, 80, 30, 60); // left, top, width, height,
// startAngle, extentAngle
// - fills part of the oval (a 'pie piece')

Polygon p; // A polygon, any number of sides
p = new Polygon(); // To draw a second polygon
p.addPoint(20, 20); // - repeat all lines except
p.addPoint(20, 120); // - "Polygon p;"
p.addPoint(120, 120);
g.drawPolygon(p); or g.fillPolygon(p);

```

* *arcWidth* and *arcHeight* are the horizontal and vertical diameter of an oval; the corners of the round rect are quarters of the oval.

b. Constants

```

Color.red, Color.green, Color.blue, Color.yellow, Color.magenta, Color.cyan,
Color.white, Color.black, Color.lightGray, Color.gray, Color.darkGray, Color.pink,
Color.orange

```

```

Font.PLAIN, Font.BOLD, Font.ITALIC, Font.BOLD+Font.ITALIC
"SansSerif" (SansSerif), "Serif" (Serif), "MonoSpaced" (MonoSpaced)

```

Any font on your computer can be used, however the standard fonts on the line above should be available on all computers.