Drawing in C#
I want to draw a line

- To draw a line, you use a pen
- Pen pen = new Pen(Color.SeaShell, 5)
  - This is the typical method used to create any object
  - Pen takes two arguments
    - Color
    - Width of line in pixels
I want to draw a line (cont)

- `g.DrawLine(pen, 130, 220, 130, 280)`
- It takes 5 arguments
  - What you will be drawing with (pen name)
  - x/y coordinates of starting point
  - x/y coordinates of ending point
  - Lines can be connected to create shapes.
I want to draw a filled shape

- To draw a filled shape, you use a brush
  - `SolidBrush yellowBrush = new SolidBrush(Color.Yellow)`
    - Again, we create an object in the standard way
    - The brush takes one argument....Color
g.FillRectangle(myBrush, ClientRectangle)

Again, we create the object

It takes 3 arguments, shape, starting color, ending color.

Other Brush options are:

- SolidColorBrush – Paints an area with a solid color
- RadialGradientBrush – Paints an area with a radial gradient
- ImageBrush – Paints an area with an image (requires image source)
- DrawingBrush – Paints an area with a drawing including vector and bitmap
- VisualBrush – Paints an area with another portion of your application
How to draw a rectangle

- Create a brush if it is going to be a filled rectangle.
- `g.FillRectangle(blackBrush, 175, 320, 35, 70)`
  - It takes 5 arguments
    - The pen or brush
    - The coordinates of the 4 corners
Point[] points2 = {new Point(70, 190), new Point(200, 50), new Point(320, 190)}

You create the object and between curly braces, give it the coordinates of the ends of each line (2 arguments), then create new points for each line.

E.Graphics.FillPolygon(new SolidBrush(Color.Red), points2)

To fill the object, make certain object is closed, then use the FillPolygon method.
Clean Up

- Calling **Dispose** allows the resources used by a **Brush** object to be reallocated for other purposes, thus freeing up memory.
- You should dispose a brush if you are no longer going to use it.