Learning Outcomes

Explain how conditional logic can manipulate the output of a computer program
Apply mathematical knowledge to **write** computer programs that manipulate pixels in a surface
Trace existing computer programs

We can create such a mapping using IF statements
The end result is that a bunch of different colours, get set to a few colours
Beware of naive solutions with a large number of 'if' statements

plain] [remember picture, overlay] [at=(current page.center)] [width=]field$_{n}ormal$;

plain] [remember picture, overlay] [at=(current page.center)] [width=]field$_{s}epia$;

Sepia Tone

First, we're calling greyScaleNew (the one with weights).
We then manipulate the red (increasing) and the blue (decreasing) channels to bring out more yellows and oranges.
It's perfectly okay to have one function calling another.
Why are we doing the comparisons on the red? Why not? After greyscale conversion, all channels are the same!
The end result is that a bunch of different colours, get set to a few colours
Why these values? Trial-and-error: Tinker the values!

fragile] Source Code: Sepia (1)

public void sepiaTint(picture) //Convert image to greyscale makeGreyscale(picture); for (int y = 0; y ¡ height; y+

int r = p.R; int g = p.G; int b = p.B;

if (r ¡ 63) r = r*1.1; b = b*0.9; ...

fragile] Source Code: Sepia (2)

... if (r ¿ 62 and r ¡ 192) bmp.SetPixel(x, y, Color.FromArgb(255, 255, 255, 255)); if (r ¿ 191) r = r*1.08; if (r ¿

Note: This requires that you incorporate a greyscale function as well.

Activity #6: Sepia Tone

In pairs:

Setup a Windows form project in Visual Studio
Refer to the following documentation
Refactor the function: `sepiaTint(picture) to use constants rather than literals`
Tinker with the values of the constants to test your solution
Then, post your solution on Teams