Lab Instructors: Beth Simon (A), Sanjoy Dasgupta (B), Keith Marzullo (C).

OVERVIEW
1) Part A: Compound Boolean expressions – ANDs (&&) and ORs (||)
2) Part B: Explore || versus && and boundary checking
3) Quiz: Each person will individually take a short quiz.

Lab 5 Part A

Copy Code:
1) Copy the starter code in for Lab 6. In the public folder Lab6
   a. Open Lab6FIRST.txt (use textpad so you can copy and paste)
   b. In Dr. Java, open your copy of Picture.java from your LABS/bookClasses folder.
      Copy and paste TWO methods (but one is commented out) into your Picture.java
      file. (Note: we won’t uncomment and use the second method until Part B)
   c. EXPLANATION: We are providing you some code that is a “Picture method”;
      that is, it is an action you can call on a Picture object.
2) Copy the testing applications for Lab 6. Copy and paste the files Lab6A.java and Lab6B.java
   from the public/Lab6 folder into your LABS/bookClasses folder.

Read and Understand
1) Part A will have a main application that calls a method on a Picture object. You will be
   exploring if statements in modifying the Picture with a compound Boolean expression.
   The goal is to make a bird picture with the bird wearing a “Black and White” convict suit
   for Halloween.
   a. Open Lab6A.java – look at the code, it calls pickAFile and you should open
      bird1.jpg (if you don’t have bird1.jpg in your mediasources directory from last
      week’s lab, copy it there from the public/Labs/Lab4).
   b. Open Picture.java – you should have already copied in the makeConvict
      method.
2) Read and understand the makeConvict method. It is not yet complete. What do you
   predict that it currently does?

Run the Lab6A.java application, select bird1.jpg as your picture and see what happens.

3) Currently, there is an if statement which replaces green pixels with black lines” on the
   bird – only on rows in the picture whose indexes are even. That is it changes the pixel
   color to black if the y index is even AND pixel’s level of Green is > 200.
4) Copy and paste the four lines of the if statement below the current if statement, but still
   inside the loops.
   a. Modify the Boolean expression of the second if statement to evaluate to true if the
      y index is odd AND the color of Green of the pixel is > 200
   b. Modify the line of code controlled by the second if statement to set the pixel color
      to white.
5) Compile, Run and check that your bird now wears a convict’s striped black and white
   outfit.
**CAN YOU DO: Do this after lab.**

a. The bird’s head is getting the striped effect too. Use Picture explorer to find the y height of the head. Modify your code so that the head doesn’t get changed from green.

The two if statements we have are inefficient and perhaps bad coding style. Re work them to be an if statement that checks for Green > 200, then inside the “true” portion of that statement, have another if statement that will either assign black or white stripes based on the value of the y index.

**Lab 6 Part B**

**Read and Understand Code:**

6) In Picture.java uncomment the `blur` method.

7) Before you get started, read and understand the code for part B – the `blur` method.
   
   This is a simple blur – it is supposed to blur pixels only along the x dimension. That is, each pixel is replaced with the average of three pixels – itself, the one to its left and the one to its right.

![Diagram of pixel averaging](image)

However, we know that we might have problems “at the edge” of the picture boundary. That is for all pixels with x offset 0, we should not do an average, since there is no pixel to the left to average in. Same with pixels with x offset `this.getWidth()-1`: there is no pixel to the right to average in. So we’ve included an if statement in the loop that should only do the averaging as long as it’s not a “first column” or “last column” pixel. That is, we’ll only calculate averages for the middle part of the picture below:

![Diagram of pixel averaging at edges](image)

**Compile, Run and Test:**

1) Compile the code, run it and test on a picture (we recommend flower1.jpg)

2) AHA! Look in the interactions pane and you will see an error! What is the error? (write the first line of the error message below)
3) Look through the various lines of the error message for the error reported from the Picture file in your blur method. What line is the error in Picture.java in the blur method? (if you can’t find this information in the error message, ask)

4) Go to that line in Picture.java. This should be the line assigning a value to redValue. To assign to redValue we need to “look at” three pixels: (x-1,y), (x,y) and (x+1,y). Why do you think we have an index out of bounds error?

5) Fix the if statement so that we only calculate an average and update the pixel color when x is bigger than the first column and smaller than the last column. You will need to make two changes to the if statement. Compile and Run. Did it work? Ask if you are not sure exactly why your if statement is now keeping you from getting an error.

Can You Do?: DO THIS AFTER LAB
1) Modify this code so that it replaces each pixel with the average of the pixel above and below it. (blur on the y-axis)
1) (4pts) Write the correct if statement (on the provided line) to make sure this code doesn’t have any `ArrayIndexOutOfBoundsException` when it runs. HINT IT NEEDS TO CHECK 4 CONDITIONS – 2 for x and 2 for y. Yes, this code is dumb and doesn’t change the picture at all.

```java
for (int x = 0; x < this.getWidth(); x++)
{
    for (int y = 0; y < this.getHeight(); y++)
    {
        int foo = this.getPixel(x-3, y-2).getRed() +
                this.getPixel(x+2,y+3).getRed();
    }
}
```

1) (1pt) What does this code do? Explain in one short English sentence.
```java
Pixel[] pixelArray = this.getPixels();
for (int i = 0; i < pixelArray.length; i++)
{
    if ( (i > pixelArray.length/2) && pixelArray[i].getBlue() == 255 &&
        pixelArray[i].getGreen() == 0 && pixelArray[i].getRed() == 0)
    {
        pixelArray[i].setColor(Color.BLACK);
    }
}
```
/* Lab6A.java -- Use compound if statements in a Picture method
to make a picture of a bird have a convict suite on (black and
white striped).
*/

class Lab6A
{
    public static void main (String[] args)
    {
        Picture p = new Picture(FileChooser.pickAFile());
        p.show();
        p.makeConvict();
        p.repaint();
    }
}

public void makeConvict()
{
    for (int x = 0; x < this.getWidth(); x++)
    {
        for (int y = 0; y < this.getHeight(); y++)
        {
            Pixel currentPix = this.getPixel(x,y);
            if ( ((y%2) == 0) && (currentPix.getGreen() > 200))
            {
                currentPix.setColor(Color.BLACK);
            }
        }
    }
}
public class Lab6B
{
    public static void main(String[] args)
    {
        Picture p = new Picture(chooser.pickAFile());
        Picture copyP = new Picture(p);
        copyP.blur();
        p.show();
        copyP.show();
    }
}

void blur()
{
    int redValue, blueValue, greenValue;
    Pixel updatePixel;
    Color c;
    for (int x = 0; x < this.getWidth(); x++)
    {
        for (int y = 0; y < this.getHeight(); y++)
        {
            updatePixel = this.getPixel(x,y);
            if ((x > 0) || (x < (this.getWidth()+1)) )
            {
                redValue = this.getPixel(x-1,y).getRed() +
                            this.getPixel(x,y).getRed() +
                            this.getPixel(x+1,y).getRed();
                blueValue = this.getPixel(x-1,y).getBlue() +
                            this.getPixel(x,y).getBlue() +
                            this.getPixel(x+1,y).getBlue();
                greenValue = this.getPixel(x-1,y).getGreen() +
                            this.getPixel(x,y).getGreen() +
                            this.getPixel(x+1,y).getGreen();
                c = new Color(redValue/3, greenValue/3, blueValue/3);  
                updatePixel.setColor(c);
            }
        }
    }
}