

Name: Eli R DeWitt

Date: 4/18/22

08.07 Picture Lab Worksheet

Directions: Make note of your responses to the following questions as you work through the activities and exercise in the lesson.

Activity 5 Questions

Question	Yes	No
1. Is the method <code>getPixels2D</code> in the <code>Picture.java</code> class?		X
2. Is the method <code>getPixels2D</code> in the <code>SimplePicture.java</code> class?	X	
3. Will the following code compile? <code>DigitalPicture p = new DigitalPicture();</code>		X
4. Assuming a no-argument constructor exists for <code>SimplePicture</code> , will the following code compile? <code>DigitalPicture p = new SimplePicture();</code>	X	
5. Assuming a no-argument constructor exists for <code>Picture</code> , will the following code compile? <code>DigitalPicture p = new Picture();</code>	X	
6. Assuming a no-argument constructor exists for <code>Picture</code> , will the following code compile? <code>SimplePicture p = new Picture();</code>	X	
7. Assuming a no-argument constructor exists for <code>SimplePicture</code> , will the following code compile? <code>Picture p = new SimplePicture();</code>		X

Activity 5 Exercise Results

1. Describe your method for `keepOnly` red, blue, or green.

It is almost identical to the <code>zeroBlue()</code> method. The only difference is that instead of
--

setting blue to zero for each pixel, it sets the red and green of each pixel to zero (keepOnlyBlue())

2. For the `negate` method, paste your code related to calculating and setting the values for red, blue, and green.

```
public void negate(){
    Pixel[][] pixels = this.getPixels2D();
    for (Pixel[] rowArray : pixels)
    {
        for (Pixel pixelObj : rowArray)
        {
            pixelObj.setRed(255-pixelObj.getRed());
            pixelObj.setGreen(255-pixelObj.getGreen());
            pixelObj.setBlue(255-pixelObj.getBlue());
        }
    }
}
```

3. Paste a copy of the image that is the result of calling the `grayscale`.

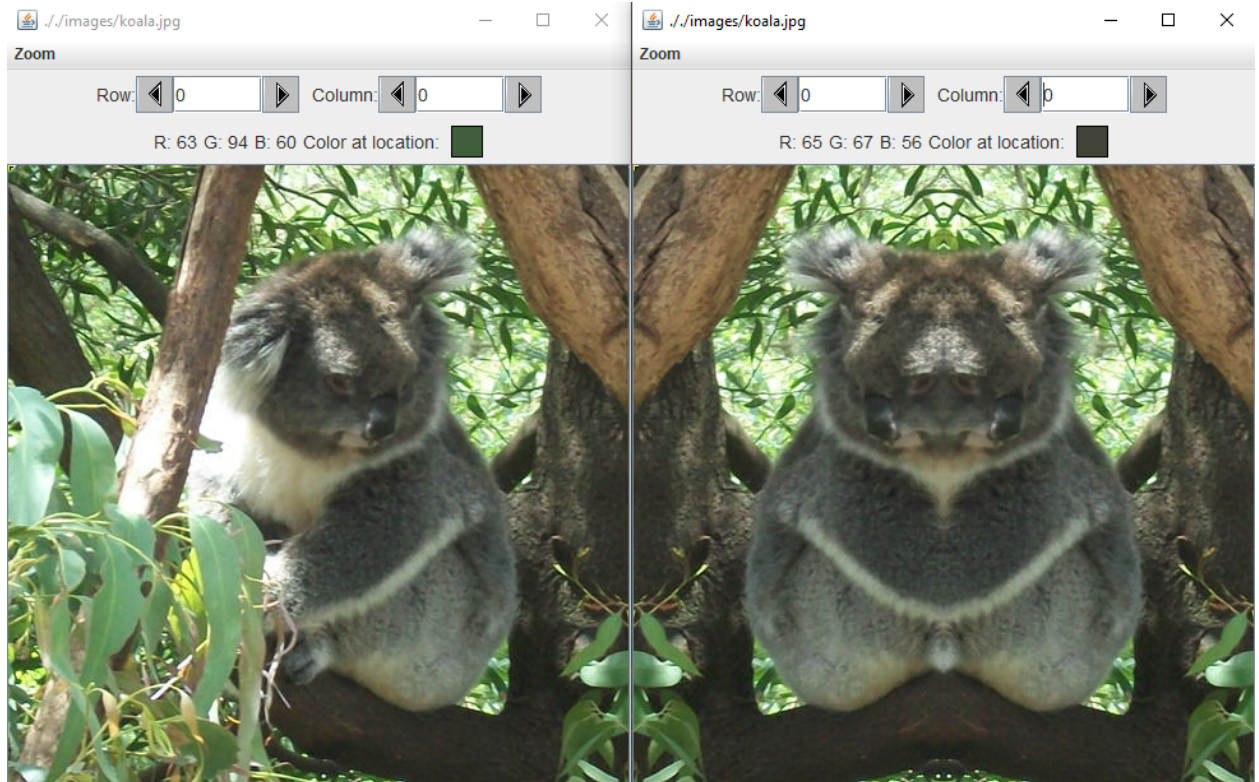


4. For the method `fixUnderwater`, describe the algorithm you'd propose to accomplish the task.

The algorithm just loops through every pixel and increases red by 50, decreases green by 50 and decreases blue by 20

Activity 6 Exercise Results

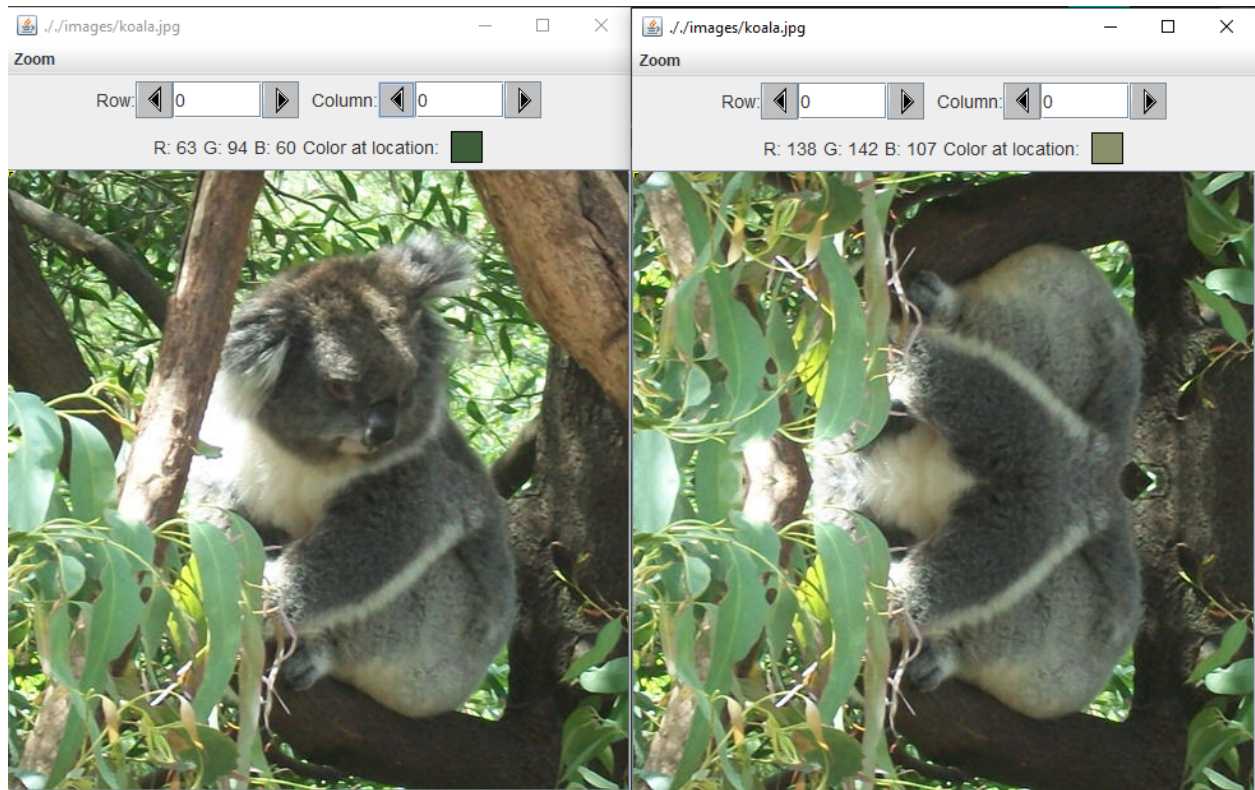
1. Paste the image that is the result of calling the method `mirrorVerticalRightToLeft`.



2. Describe the algorithm for the method `mirrorHorizontal` works.

The algorithm looks at the top half of the image, and copies the top row to the bottom row, 2nd top row to 2nd bottom row ect. It does this by looping through the first half of the rows all the columns

3. Paste the image that is the result of calling the method `mirrorHorizontalBotToTop`.



Activity 7 Questions

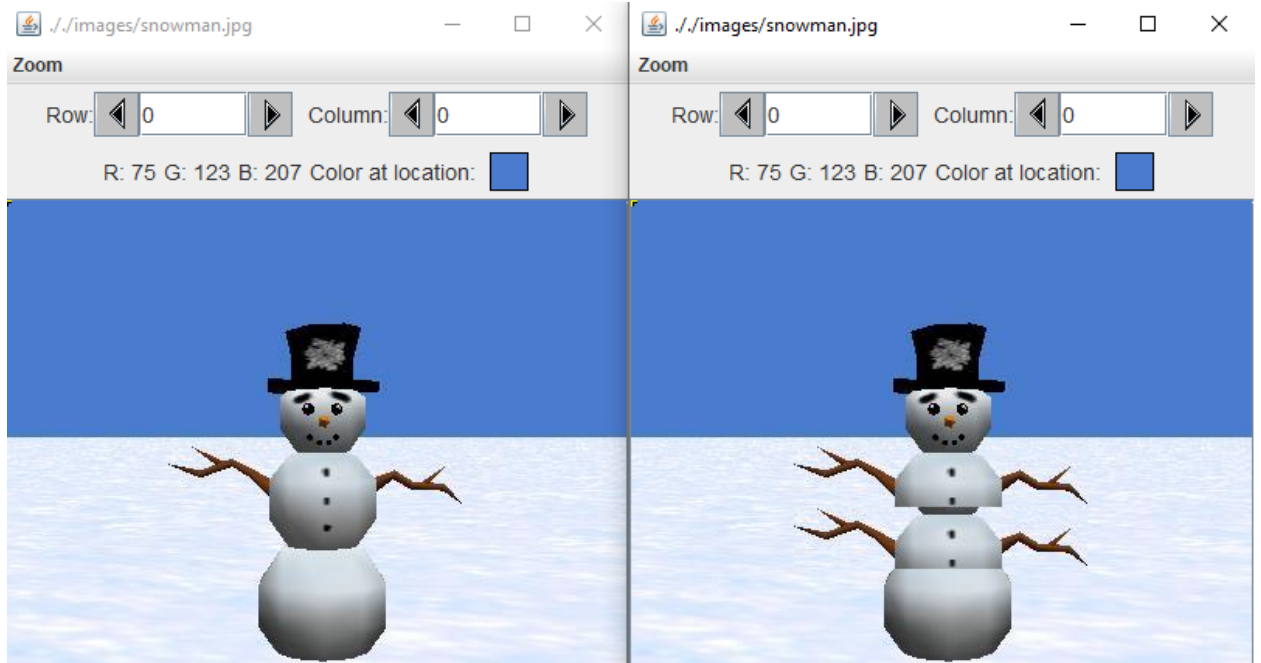
1. How many times would the body of this nested `for` loop execute? **90**

```
for(int row = 7; row < 17; row++)
    for(int col = 6; col < 15; col++)
```
2. How many times would the body of this nested `for` loop execute? **112**

```
for(int row = 5; row <= 11; row++)
    for(int col = 3; col <= 18; col++)
```

Activity 7 Exercise Results

1. What value is displayed for `count` after the nested loop ends in the `mirrorTemple` method? **18410**
2. Paste the image that is the result of calling the method `mirrorArms`.



3. Paste the image that is the result of calling the method `mirrorGull`.

