Name: Paul Chafetz  
Date:

**15.07 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 getPixels2D in the Picture.java class? |  | No |
| 1. Is the method getPixels2D in the SimplePicture.java class? | Yes |  |
| 1. Will the following code compile?  DigitalPicture p = new DigitalPicture(); |  | No |
| 1. Assuming a no-argument constructor exists for SimplePicture, will the following code compile?  DigitalPicture p = new SimplePicture(); | Yes |  |
| 1. Assuming a no-argument constructor exists for Picture, will the following code compile?  DigitalPicture p = new Picture(); | Yes |  |
| 1. Assuming a no-argument constructor exists for Picture, will the following code compile?  SimplePicture p = new Picture(); | Yes |  |
| 1. Assuming a no-argument constructor exists for SimplePicture, will the following code compile?  Picture p = new SimplePicture(); |  | No |

**Activity 5 Exercise Results**

1. Describe your method for keepOnly red, blue, or green.   
   My keepOnlyBlue method sets all Red and Green to 0. Since there will be no red or green color, all that is left is blue. This is repeated for each pixel in the image.
2. For the negate method, paste your code related to calculating and setting the values for red, blue, and green.

|  |
| --- |
| pixelObj.setRed(pixelObj.getRed() - 255); |

|  |
| --- |
| pixelObj.setGreen(pixelObj.getGreen() - 255); |

pixelObj.setBlue(pixelObj.getBlue() - 255);

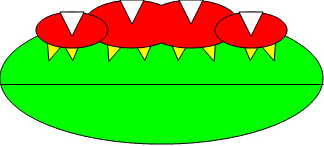
1. Paste a copy of the image that is the result of calling the grayscale.



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

The first thing I would do is find a fish by its pixels. I would find the minimum and maximum red, green, and blue values, as well as the average of the red, green, and blue across the fish. From there, I would need to see how every other pixel compares to the sample fish by finding the difference between the values of the fish and each pixel in the 2D array of pixels. If this difference is too small (the pixel color is similar to the fish) then I would set that pixel to its current blue value minus some value at my discretion. If the difference is large, I would add more blue to the pixel to add contrast from the blue color of the water.

**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.   
   This method works in the same manner as the vertical mirroring method. The height is the number of pixels in a column. For each row less than the column height (top half of the image) and for each column (total rows), set the bottom pixel to the corresponding top pixel, which is one less the height less the current row position.
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++)

* 1. How many times would the body of this nested for loop execute? \_\_90\_\_

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.  
   