**Picture Lab – Activity 7: Mirroring part of a picture**

**Questions**

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

for (int row = 7; row < 17; row++)

for (int col = 6; col < 15; col++)

90 times

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

for (int row = 5; row <= 11; row++)

for (int col = 3; col <= 18; col++)

112 Times

**Exercises -** Copy and paste the methods from the Picture class below once you have them working.

1. Check the calculation of the number of times the body of the nested loop executes by adding an integer count variable to the mirrorTemple method that starts out at 0 and increments inside the body of the loop. Copy and Paste the mirrorTemple method below. Print the value of count after the nested loop ends.

*/\*\* Mirror just part of a picture of a temple \*/*public void mirrorTemple()  
{  
 int mirrorPoint = 276;  
 Pixel leftPixel = null;  
 Pixel rightPixel = null;  
 int count = 0;  
 Pixel[][] pixels = this.getPixels2D();  
   
 // loop through the rows  
 for (int row = 27; row < 97; row++)  
 {  
 // loop from 13 to just before the mirror point  
 for (int col = 13; col < mirrorPoint; col++)  
 {  
 leftPixel = pixels[row][col];   
 rightPixel = pixels[row][mirrorPoint - col + mirrorPoint];  
 rightPixel.setColor(leftPixel.getColor());  
 count++;  
 }  
 }  
  
 System.out.println(count);  
}

1. Write the method mirrorArms to mirror the arms on the snowman (“snowman.jpg”) to make a snowman with 4 arms. Write a class (static) test method in PictureTester to test this new method and call it in the main method.

*/\*\*  
 \* Mirrors arms of the snowman vertically  
 \*/*public void mirrorArms()  
{  
 Pixel[][] pixels = this.getPixels2D();  
  
 Pixel topPixel = null;  
 Pixel bottomPixel = null;  
  
 for (int row = 160; row < 200; row++)  
 {  
 for (int col = 105; col < 170; col++)  
 {  
 topPixel = pixels[row][col];  
 bottomPixel = pixels[195 - row + 195][col];  
 bottomPixel.setColor(topPixel.getColor());  
 }  
 }  
  
 for (int row = 170; row < 200; row++)  
 {  
 for (int col = 239; col < 295; col++)  
 {  
 topPixel = pixels[row][col];  
 bottomPixel = pixels[200 - row + 200][col];  
 bottomPixel.setColor(topPixel.getColor());  
 }  
 }  
}

1. Write the method mirrorGull to mirror the seagull (“seagull.jpg”) to the right so that there are two seagulls on the beach near each other. Write a class (static) test method in PictureTester to test this new method and call it in the main method.

*/\*\*  
 \* Mirrors the seagual horizontally  
 \*/*public void mirrorGull()  
{  
 Pixel[][] pixels = this.getPixels2D();  
  
 Pixel rightPixel = null;  
 Pixel leftPixel = null;  
  
 for (int row = 235; row < 325; row++)  
 {  
 for (int col = 240; col < 350; col++)  
 {  
 rightPixel = pixels[row][col];  
 leftPixel = pixels[row][350 - col + 350/3];  
 leftPixel.setColor(rightPixel.getColor());  
 }  
 }  
}