

Google Doc Access Directions:

- Please click on **File** in the upper left corner.
- If you are working on a Chromebook or Google Docs, choose the **Make a copy option** and save a copy of the document to your Google Drive.
- If not, choose the **Download** as option and then the **Microsoft Word (.docx)** option to download an editable copy of the document to your computer.

Using the information you have gained so far, modify the code. Click “Submit Assignment” in the upper right corner of the screen to submit your work. Be sure and save the files as
YourNameMod13PictureLabAssignmentEight

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

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

Exercises

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. Print the value of `count` after the nested loop ends.

18,410

2. 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.



Module13LessonTwoAssignmentEight

Name _____

```
public static void testMirrorArms() {  
    Picture snowman = new Picture(fileName:"snowman.jpg");  
    snowman.explore(); // Show the original picture  
    snowman.mirrorArms();  
    snowman.explore(); // Show the picture with all four arms  
}
```

```
public void mirrorArms() {  
    Pixel[][] pixels = this.getPixels2D();  
    Pixel topPixel = null;  
    Pixel bottomPixel = null;  
    int leftArmBottomRow = Picture.mirrorArms()  
    int leftArmBottomRow = 190; // ending row for the left arm  
    int leftArmColStart = 105; // starting column for the left arm  
    int leftArmColEnd = 170; // ending column for the left arm  
  
    for (int row = leftArmTopRow; row < leftArmBottomRow; row++) {  
        for (int col = leftArmColStart, mirrorCol = leftArmColEnd; col <= (leftArmColStart + leftArmColEnd) / 2; col++, mirrorCol--) {  
            topPixel = pixels[row][col];  
            bottomPixel = pixels[row][mirrorCol];  
            bottomPixel.setColor(topPixel.getColor());  
        }  
    }  
  
    int rightArmTopRow = 155; // Similar to left arm for simplicity  
    int rightArmBottomRow = 190;  
    int rightArmColStart = 230; // starting column for the right arm  
    int rightArmColEnd = 295; // ending column for the right arm  
  
    for (int row = rightArmTopRow; row < rightArmBottomRow; row++) {  
        for (int col = rightArmColStart, mirrorCol = rightArmColEnd; col <= (rightArmColStart + rightArmColEnd) / 2; col++, mirrorCol--) {  
            topPixel = pixels[row][col];  
            bottomPixel = pixels[row][mirrorCol];  
            topPixel.setColor(bottomPixel.getColor());  
        }  
    }  
}
```

3. 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.



Module13LessonTwoAssignmentEight

Name _____

```
public static void testMirrorGull() {  
    Picture seagull = new Picture(fileName:"seagull.jpg");  
    seagull.explore(); // Show the original picture  
    seagull.mirrorGull();  
    seagull.explore(); // Show the mirrored picture  
}
```

```
public void mirrorGull() {  
    Pixel[][] pixels = this.getPixels2D();  
    int width = pixels[0].length;  
    int height = pixels.length;  
  
    // Loop over the left half of the image  
    for (int row = 0; row < height; row++) {  
        for (int col = 0; col < width / 2; col++) {  
            Pixel leftPixel = pixels[row][col];  
            Pixel rightPixel = pixels[row][width - col - 1]; // Find the corresponding pixel on the right half  
            rightPixel.setColor(leftPixel.getColor()); // Set the color of the right pixel to that of the left pixel  
        }  
    }  
}
```