CS-207: Programming II
Fall 2016
Northeastern Illinois University
Research Lab: Image Processing
Due: Thursday, 09/29 at 9:00 a.m.

Goal:
The goal of this research lab is to use and modify the provided Java files to investigate how to manipulate 2-dimensional arrays in image processing. As this is a "research" lab, you will need to investigate and analyze the code that has been provided for you in order to be able to modify it using the materials presented in class.

The Problem:
You have been provided with an image file named `"animals.jpg"` and an `Image` class that contains code needed to read in an image file, display the image, and write an image file to the computer. You will need to create the methods necessary to turn the image upside-down and reverse it.

Instructions:

- You should work in groups of 2-3 individuals. Groups of more than 3 are **not** permitted.
- Each group should submit ONE lab write-up. It is the responsibility of each group member to ensure that their name is on the write-up.
- The lab write-up should be typed! Type each question (and the question number) followed by your group's answer. **Convert your lab write-up to a .pdf.**
- You should use complete sentences and proper grammar in your write-up. Use spell-check! This counts as part of your grade.
- You should not copy/paste directly from your sources for your answers (this is called plagiarism). Instead, you should re-word the information in your own words.
- Submit the pdf and .java files (both the Image.java and TestImage.java files) to D2L by the specified due date.
- Each member of the group must turn in a hard-copy of the peer assessment on the day the research lab is due. The peer assessment counts as a significant part of your grade and you will receive a **zero** for that portion of the research lab grade if you do not turn it in.

Part A: Getting Started
Download the files from D2L. Research is often a very ambiguous process and it can be difficult to figure out how to get started on a particular problem. In this lab, we are going to imitate a scenario that happens frequently in software development - working with code that you did not write! Your boss wants you to manipulate the `animals.jpg` image in several ways (as described

above). Your boss is not a programmer, but he received two .java files from a friend (who is a programmer) and he wants you to use them to get started on the image manipulation.

Question #A.1
What is the first file your group looks at and why? What are the first (and most important) things that your group takes notice of in the code in this file?

Question #A.2
What happens in the constructor?

Question #A.3
Find the `read` method (note that the `read` method is not complete). How many parameters does the `read` method take and what are their types? What is its return type? How many objects have been created in this method?

Question #A.4
Notice that the `read` method has an additional part appended to the end of the method signature:
`throws Exception`
Use Google to find out the role of this additional part - what does it do? Where did you find your answer?

Question #A.5
The first line of the `read` method creates a `File` object. What is a `File` object? In what Java package does the `File` class live? Where did you find your answers?

Question #A.6
The second line of the `read` method creates a `BufferedImage` object by reading in the `File` object using the `ImageIO` class. What is a `BufferedImage` object? More specifically - what is a `BufferedImage` object composed of? In what Java package does the `BufferedImage` class live? Where did you find your answers?

Question #A.7
In Java, how are images represented? What are they composed of? This is a very important question! In order to be able to manipulate the images (flip them around, etc), you have to be able manipulate the pieces of the image in Java.

Question #A.8
After the `BufferedImage` object is created, the `width` and `height` instance variables are set using the `getWidth` and `getHeight` methods from the `BufferedImage` class. What do these methods do? Hint: If you wrote down that they give you the width and height of the image, you are `wrong`! How do they relate to the "pieces" of an image?

Part B: Completing the `read` method
Your boss did not give you a lot of instructions from his programmer friend, but you do know that the `read` method is not complete. You now know that the image has been read in and stored using the `BufferedImage` class/object, but it is still not in a format (think grids!!) that

you can work with. In order to manipulate the image, you need it to be in a grid-like format.

Question #B.1
Is there anything in the `Image.java` code that makes you think of a grid? What is it?

Question #B.2 + coding
The `pixels` instance variable has been declared, but not created. How big is this array instance variable? Create it and compile your code.

Question #B.3 + coding
Now that you have a 2D integer array of all 0s that is the same size as your image, you need to assign each pixel from your `BufferedImage` to its corresponding value in the array. Find the Java documentation for the `BufferedImage` class. Is there a method that returns an integer value for each piece of the image? Hint: Remember that each position in an image can be thought of as a grid where the point (0, 0) is in the upper left-hand corner. Use the method and assign it to the correct element in your 2D array. Compile your code.

Part C: Understanding the `write`, `draw` and `createBufferedImage` methods
It turns out that your boss' programmer friend created several additional methods that you are able to use - but you need to understand how they work in order to use them!

Question #C.1
Find the `createBufferedImage` method. What is the return type of this method? Using the information about the `BufferedImage` class from your previous research, what does this method do? Why would this method be useful for manipulating an image?

Question #C.2
Find the `write` method. What is the return type of this method? How many parameters does this method take and what are their types? What does this method do (be specific!)? Why would you want to use this method?

Question #C.3
Find the `draw` method. What is the return type of this method? How many parameters does this method take and what are their types? What are the second and third parameters used for? What is a `Graphics` object and what does its `drawImage` method do? How will this method (the `draw` method of the `Image` class) be useful?

Part D: The `TestImage` class
Your boss' friend made a test class that you can use to test your `Image` class. Again, it's important to understand the code that has already been written for you.

Question #D.1
What do the first 4 lines in the `main` method do?

Question #D.2
Notice that there is some code that looks like a block of code - with the words `try` and `catch`, but with nothing inside the braces that follow the word `try`. This is called a try-catch block.

You will learn about this in detail later in the class. Using Google, find out what a try-catch block should do and explain it.

Question #D.3 + coding
You want to make a new `Image` object using the `Image` class that you finished. What type of parameter does your `Image` class constructor require? Inside of the try-block, create a new `Image` object using the animals image provided. Make sure that the `animals.jpg` file, the `Image.java` class, and the `TestImage.java` class are all in the same folder! Then, call the `draw` method on the `Image` object that you just created and pass in the `Graphics` object, 10 for the parameter `x`, and `40` for the parameter `y`. Compile and run the `TestImage` class.

Part E: Manipulating the image
After all that work trying to understand the other programmer's code, you are ready to write your own code to manipulate the image.

Question #E.1 + coding
The first method that your boss wants you to create in the `Image` class is named `flipY`. This method should take the image and flip it around the y-axis. Think of this as what you would see if you looked at the image in a mirror. Everything would be reversed horizontally, but not vertically (i.e. not upside-down). Should your method take any parameters? Why or why not? Should your method return anything? Why or why not? Create the code for this method and then compile. Then, call your method on the `Image` object in the `TestImage` class and make sure to draw it (place it next to the original image!). Compile and run your `TestImage` class.

Question #E.2 + coding
The second method that your boss wants you to create in the `Image` class is named `flipX`. This method should take the image and flip it around the x-axis, meaning that the modified image should be upside-down. Should your method take any parameters? Why or why not? Should your method return anything? Why or why not? Create the code for this method and then compile. Then, call your method on the `Image` object in the `TestImage` class and make sure to draw it (place it next to the original image!). Don't forget to comment out the code for flipping the image horizontally, otherwise your image will end up reversed and upside down! Compile and run your `TestImage` class

Question #E.3 - only coding
Your boss is so happy with you! But there's just one more thing. Your boss wants saved copies of the modified images. You're ready for this - no hints on this one! Write the code to save copies of the modified image (one saved copy of the image reversed and one saved copy of the image upside-down) in .jpg formats. Do not overwrite your original image!

Part F: Final Summary
Whenever you complete a project, it is important to assess what you think went well and what you need to improve on.

Question #F.1
What was the most challenging part of this research lab for your group?

Question #F.2
What did your group learn/find the most useful by doing this research lab?

Question #F.3
What was the most fun aspect of doing this research lab?