Question #A.1

Which of the two Java classes can be used to make objects? What are the first things that your group looks for and notices in this class?

Answer: Image.java is used to actually assign a value to the object, where the testImage.java file is allocating the memory necessary to construct the object. We look to the main method first and notice the java packages as well the keywords ‘throw’ and ‘catch’ which we need to understand to further decipher the program. We also are looking for the creation of the objects within the main, and since the ‘new’ keyword is used within the method, we discern that it is creating an object and calling the class, which assigns the parameters of the object.

Question #A.2

The constructor calls the read method. Find the read method. How many parameters does the read method take and what are their types? What is the return type?

Answer: It takes 1 string parameter. It does not return anything.

Question #A.3

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?

Answer: So this warns users that there may be some risky behavior which may result in not working properly and not achieve the desired results. We asked the peer leader Emett who gave us a brief overview.
alternate Throws exception is added to the read method to indicate that at some point the code is not going to work properly. Additionally the throws methods classifies these exceptions in three categories. There can be an exception due to input error, there can also be an exception due to hardware issues such as the device being used does not support the program, and finally the last exception can be caused by a bug in the code itself. Answer was found in java docs

Question #A.4

The first line of the read method creates a File object. In what Java package does the File class live? Hint: Use the import statements and Java docs to help determine this! What is a File object?

Answer: It lives in the java.io.*. A file object represents the actual file or directory on the disk.

Alternate: The file Object lives in the java.io.* package. This object is an abstract representation of file and directory pathnames. Essentially java.io.file allows to import and then read and write image files.

*Question #A.5(type of method??)

The second line of the read method creates a BufferedImage object by passing in a reference to the File object to another read method. However, this is a different read method! Which class is this read method found in? Hint: Look at who is calling the method. What type of method is this read method?

Additional: The read method used by BufferedImage object is found in the javax.imageio.* package. This is a subclass that describes an Image with method that takes a object type parameter. It is used to load a new bufImage in by using the information from original image file.

Question #A.6

A BufferedImage is a representation of an Image that is composed of two parts: aColorModel and a Raster. In what Java package does the BufferedImage class live? Hint: Use the import statements and Java docs to help determine this!

Answer: java.awt.image.*

Question #A.7

The Raster class is an underlying component of BufferedImages. What type of object is created from the Raster class (explain in detail)?

Answer: A Raster class represents a rectangular array of pixels. It assigns pixel values for the rectangle in a plane. This rectangle is also knowns as the Rasters bounding rectangle and it is called by using the getBounds methods. It has a minX and minY, width, and height values. For the purpose of our research lab the raster class would create an rectangular object which has the values corresponding to the original image.

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. An image is essentially a 2D array of pixels where each element is an RGB value. How do the width and height instance variables relate to the number of rows and columns in an image?

Answer: The width will equal the amount of columns and the height will equal the amount of rows in the array created by the BufferedImage raster 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 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 already existing in the Image.java code that makes you think of a grid? What is it?

Answer: The first, can be seen in the read method in Image.java. File class imports a image file. Then this imagefile(reference) is used to retain the object and get its width and height. Second, in the method createbufferedImage(). Here we see bufImage creates an object with columns and rows and as pixel values to each space inside the object array.

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.

The size of the array will be this.width in columns and this.height in rows

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's RGB value 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 RGB value for each element of the image so that you can copy it over to your instance variable? Use the method and assign it to the correct element in your instance variable. Compile your code.

Answer: Yes its the getRGB() method.

```
public void read(String filename) throws Exception
{
  File fileImage = new File(filename);

  BufferedImage bufImage = ImageIO.read(fileImage);
  this.width = bufImage.getWidth();
  this.height = bufImage.getHeight();

  // Complete the remainder of this method
  pixels = new int[this.height][this.width];
  for( int i = 0; i < height; i++ )
    for( int j = 0; j < width; j++ )
      pixels[i][j] = bufImage.getRGB( j, i );
}
```

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?

Answer: It is a BufferedImage return type which would classify it as an object. This method recreates the image as a private object that can be manipulated without altering the original bufferedImage. This is what allows the pixels to be assigned which allows for the manipulation of said pixels.

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?

Answer: The return type is void with one only one parameter which is of String type but would actually be an image. The function of this method is to render the images on different memory space than the original and generate a file.

Additional : the write method does a couple things. It uses the File class to import and write the image file. Then its uses a string to get destination file extension. It also creates bufImage object by using the method createBufferedImage(its a new object so we don't alter the original). Finally the it uses ImageIo.write to write and save the image by using the file pathname and destination file extension.

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?

Answer: The return type is void. This method takes 3 parameters, a Graphics object and two integers. A graphics object is a window in which our images will be displayed, the main method will then rendered a picture through the draw method. It is useful because it allows for us to specify which image we would like to display and or alter through the methods in the Image class.

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?

Answer: The first four lines of the main method are creating frame object which will be used to contain the graphics being passed. This is essentially what makes the window frame we will be viewing our altered and original image in.

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.

Answer: A try block contains lines of code that could throw an exception. These exceptions then are caught and handled by the catch block. A try block is always followed by a catch block.

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.

Answer: The parameter of the Image class constructor is a String type which will be the filename of the image aka "animals.jpg"

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, in the TestImage class, call the flipY method on the Image object. Make sure to draw it using the draw method (place it next to the original image!). Compile and run your TestImage class.

Answer: This method does not take any parameters, this is due to the fact that the image must remain static, and therefore only altered through the 2D instance variable we have created. Since this is the case, the best method to altering the picture is to simply call a method that deals with our instance variable directly and doesn't need any parameters or returns.

```
public void flipY()
{
  int[][] a = new int[this.height][this.width];
  for(int i = 0; i < this.height; i++)
  {
    for(int j = this.width - 1; j >= 0; j--)
    {
      a[i][this.width-j-1] = this.pixels[i][j];
    }
  }
  this.pixels = a;
}
```

Question #E.2 + coding

The second method that your boss wants you to create in the Image class is named grayscale. This method should change the image from a color image into grayscale. Should your method take any parameters? Why or why not? Should your method return anything? Why or why not? In order to do this, you will need to use the Color class. Which constructor should you use from the Color class to create a Color object? (Hint: Remember that each element of your pixels array is an RGB value.) In order to convert to grayscale, you will need to get the red, blue and green values from the Color object. Which methods should you use to do this? A simple way to create a grayscale color is to add the red, blue, and green values and divide the sum by 3. Use the resulting average for each of the red, green, and blue values of the Color constructor that takes 3 parameters and create a new Color object. This Color object represents a gray color for that pixel. Get the total integer RGB value using the proper getter from the Color class for this gray color and assign it to the correct location in your pixels array. Then, in the TestImage class, call the grayscale method on the Image object. Don't forget to comment out the code for flipping the image horizontally, otherwise your image will end up reversed and in gray! Compile and run your TestImage class.

Answer: The method grayscale should not take any parameters, because it is only called to alter our originally constructed instance array. There is also no method return because it is simply taking the original image input and making it into a different array which turns it gray. We should use the Color constructor to get a new color object and the getBlue, getRed, and getGreen methods to get the values necessary to create the required gray color object.

```
public void greyscale()
  {
    int[][] a = new int[this.height][this.width];
    for(int i = 0; i < this.height; i++)
    {
      for(int j = 0; j < this.width; j++)
      {
        Color o = new Color(this.pixels[i][j]);
        int b = o.getBlue();
        int r = o.getRed();
        int g = o.getGreen();
        int grey = (r + b + g)/3;
        Color o1 = new Color(grey, grey, grey);
        int total = o1.getRGB();
        this.pixels[i][j] = total;
      }
    }
  }
```

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!

```
public void flipX()
{
  int[][] a = new int[this.height][this.width];
  for(int i = 0; i < this.height; i++)
  {
    for(int j = this.width - 1; j >= 0; j--)
    {
      a[this.height - i - 1][j] = this.pixels[i][j];
    }
  }
  this.pixels = a;
}
```

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?

Answer: The coding was difficult and not having explicit directions on what to do within each class package with images. It seemed like there was not enough time since there was a considerable amount of data to go over and try understand.

Question #F.2

What did your group learn/find the most useful by doing this research lab?

Answer: The usage of image packages/javadocs.

Question #F.3

What was the most fun aspect of doing this research lab?

Answer: Finally understanding the prompt and the code necessary to achieve the predicted results.