**CSCI 3302 Pair Programming Assignment 01 (100 Points)**

**Due: Sept 8, 8:00 AM**

**GitHub Link:** [**Pair Program 01**](https://classroom.github.com/a/9YnD0BSE)

**Objectives:**

* Demonstrate understanding of prerequisite Java topics.
* Demonstrate how to implement operations on double arrays.

**Assignment Assistance:**

* This homework assignment is due before the date and time specified above.
* You and your partner must work on this TOGETHER. One person is the DRIVER, the other is the NAVIGATOR. If you are unfamiliar with paired programming, please Google and get acquainted. Here is a quick video with an overview: <https://youtu.be/q7d_JtyCq1A>
* This assignment is restricted to you and your partner. You may not receive help from any other person except the instructor or the AARC (help from the AARC must be well documented!).
* Any resource used (other than Dr. Becnel or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or in whole from any source.
* If you need help, visit your instructor during his posted office hours. If your schedule cannot accommodate any of these times, then email your instructor to schedule a different time.

**Problem Description:**

1. After getting access to the repository, you will notice there are a lot of files. This is common for a junior developer starting with a company. But don’t worry….everything you need to do for this assignment can be found in one file.
   1. **ImageManipulator** – a class that acts on images (double arrays of integers) to perform various operations such as rotations, filtering, flipping, etc.
   2. ImageDisplay – this file contains the main method which runs the entire program. Use this when you are ready to see the result of your work. *You should not edit this file (outside of possible printlns to help with testing).*
   3. TestIM – this is a simple test program that you can use to test methods in the ImageManipulator file. Use (and change if you like) this file when testing.
2. You should start by understanding the code you were given. In particular, notice a few things in ImageManipulator:
   1. ImageDisplay has one field, a double array of integers. These are interpreted as colors to display the image.
   2. There is a method setImage which takes the pixels of the image and reads them into the array.
   3. We implemented a couple of the methods in class: filter and grayscale
3. In VS Code, if you look under Problems, you will see four TODOs for ImageMainpulator.java  
   A screenshot of a computer

   Description automatically generated with medium confidence   
   Here are examples of each operation:
   1. rotate – rotates the pixels 90 degrees to the right.   
      Starting Image Pixels: Rotated Image Pixels:

1 5 9 4 3 2 1

2 6 10 8 7 6 5

3 7 11 12 11 10 9

4 8 12

* 1. flipHorizontal – flips the pixel values of the image horizontally  
     Starting Image Pixels: After Flip Pixels:

1 5 9 9 5 1

2 6 10 10 6 2

3 7 11 11 7 3

4 8 12 12 8 4

* 1. flipVertical – flips the pixel values of the image vertically  
     Starting Image Pixels: After Flip Pixels:

1 5 9 4 8 12

2 6 10 3 7 11

3 7 11 2 6 10

4 8 12 1 5 9

* 1. invert – inverts the colors of the image. That is, if a pixel has RGB values given by (r,g,b), then the new value will be (255-r, 255-g, 255-b).   
     Starting Image Pixels: After Flip Pixels (in Hex for ease of comparison):

1 5 9 fffffe fffffa fffff6

2 6 10 fffffd fffff9 fffff5

3 7 11 fffffc fffff8 fffff4

4 8 12 fffffb fffff7 fffff3

1. Add appropriate comments to all your methods. You may consider using /\*\* for the Java doc comment. You should also add an appropriate file heading to the file which includes both your name and your partner's name.
2. Your submission should compile and run with errors.
3. You may write any private helper methods if needed.

**Hints:**

* While testing your code, change the data or create new data files to ensure you fully test your methods.
* To start testing your methods use TestIM.java file. As you implement methods, uncomment the corresponding portions of the file to test. Note: This file is not meant to be a complete test. After this file is working well, try your methods by running ImageDisplay.java

**Submission:**

* Review the Evaluation below to ensure you have met all the requirements.
* Only one person needs to turn in/commit the assignment. However, both members of the team are expected to understand the solutions and be able to answer questions about the solution.
* Commit all files to GitHub. Upload a backup copy to D2L. You may include testing files in your repository; however, these will not be considered when grading. If you wish to include non-working code for insight into your thought process, make sure to contain it within comment blocks and ensure that submission successfully compiles/runs.

**Evaluation**

* 1. Project is late or not submitted at all. -100
  2. Project does not run/compile. -50
  3. Project compiles with warnings. -30
  4. Project does not correctly implement the interface. -30
  5. Calculations/output are for requirements incorrect. -10 each
  6. Code is not well organized or properly indented. -5
  7. Code is inadequately commented for readability. -5
  8. Code does not contain the student’s name, course section, -5

and date of submission.

* 1. Code is not submitted to the Github -15