COSC 1P02 Assignment 4

“The medium is the message.”

Due: Nov. 2, 2018 @ 4:00 pm** **(late date Nov. 5 @ 4:00 pm)

The emphasis for this assignment is processing all the pixels of a picture. In preparation for this assignment, create a folder called `Assign_4` for submission of the assignment and two subfolders: `Assign_4_A` and `Assign_4_B` for the two DrJava projects for the two parts of the assignment.

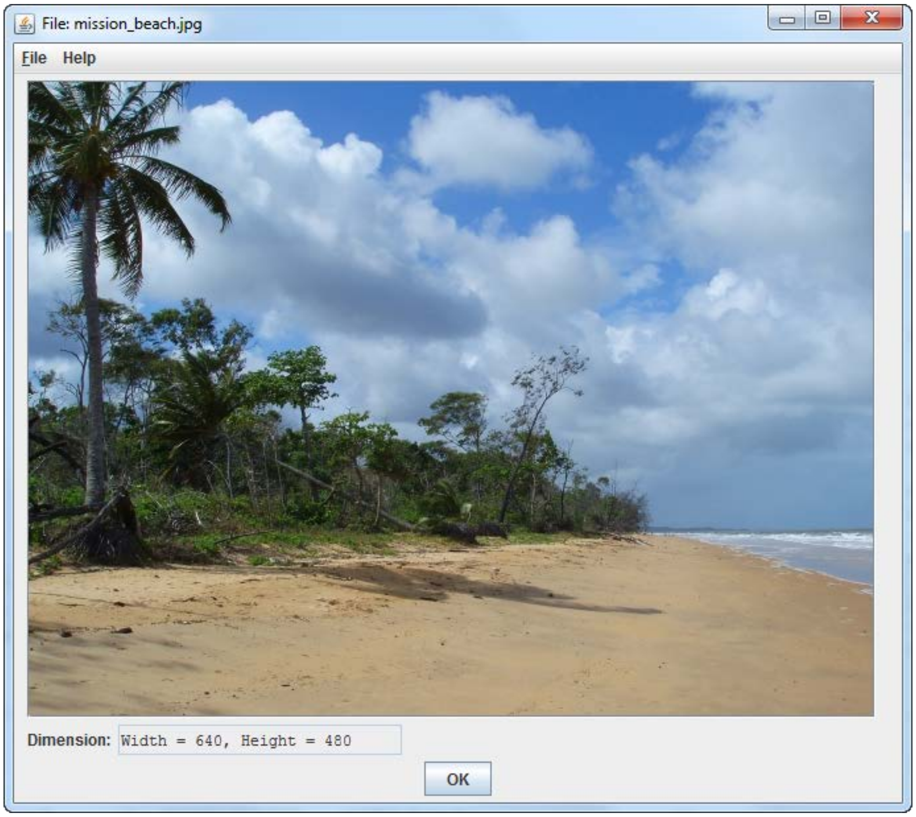
Part A – “That’s a negative”

Write a program to produce a color negative (such as could be used in a printing process) of a picture. A color negative is a picture in which the color channels for each pixel have had their value inverted. That is, if the red channel value is `r`, it is replaced by `255-r`. The same is true for the green and blue channels.

Write a method:

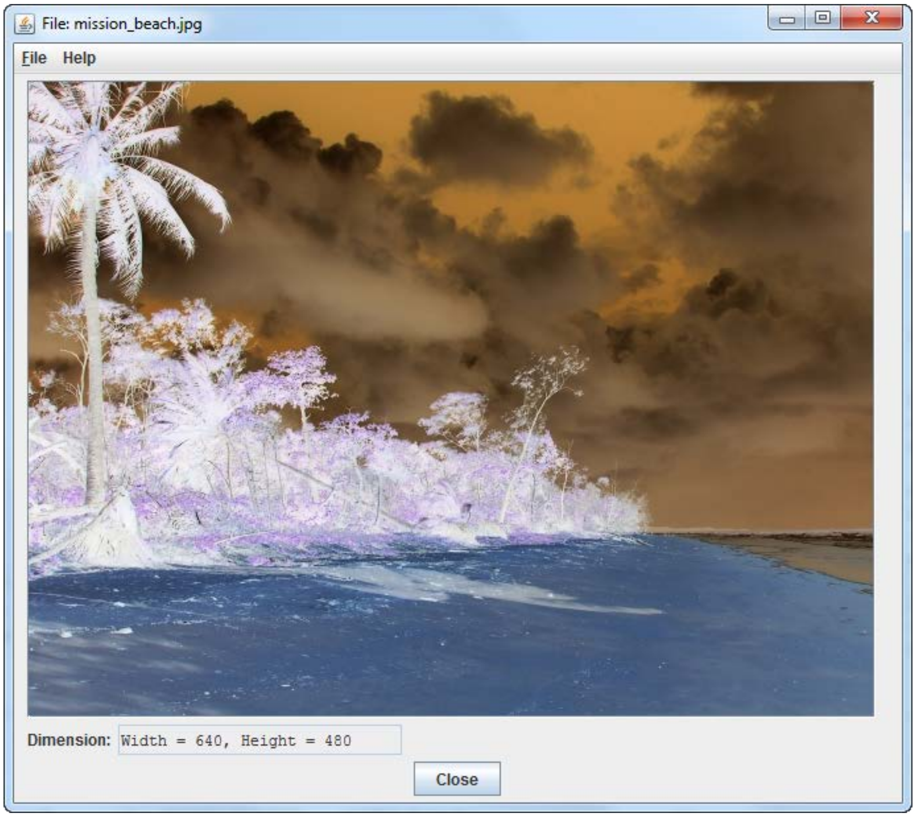
```
   private void makeNegative ( Picture aPic ) {
```

that changes all the pixels in `aPic` to their negative, producing a color negative. The program should allow the user to select a picture, present it on the display, modify it to a color negative and save the result. For example, if the original image is:



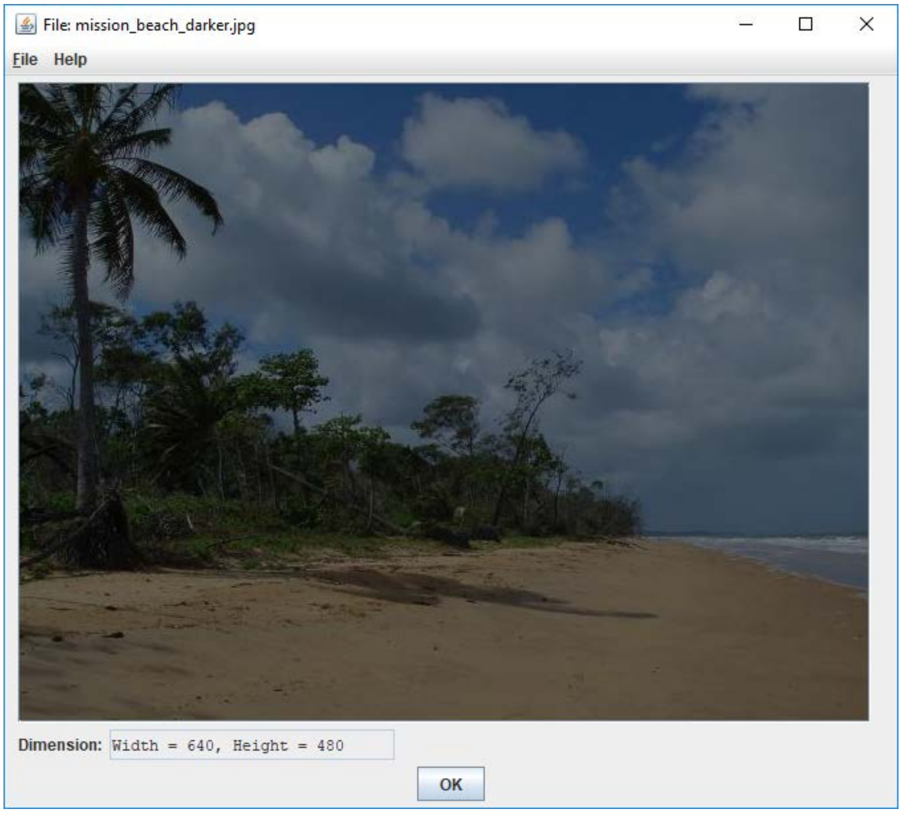
Dimension: Width = 640, Height = 480

the negative image is:



Part B – Shedding a little light on the subject.

Sometimes when a picture is taken there wasn't enough light, and if the flash wasn't used the result is a dark image such as:



In lab we developed a program called `Brighten` that brightened an image by multiplying the three color channel values of each pixel by some factor (e.g. 2.0). As we discovered, if you set a channel value to greater than 255, the result was a “random” color. So we wrote a method called `clip` that clipped the value we were setting the channel value to at 255. When the intensity of the channels of a pixel are so great that the color bleeds out (i.e. tends toward white), this is called saturation.

Saturation is similar to clipping that we encountered when increasing the amplitude of a sound. We can deal with saturation in pictures in the same way we dealt with clipping in sounds—normalization—increasing the intensities as much as possible without exceeding the limit. With sounds there was one value in question for each sample—the amplitude—and we scanned the sound to find the largest magnitude of the amplitude of all samples. With pictures there are three values in question for each pixel—the R, G and B channel values—none of which we want to saturate, so we need to determine the largest intensity of any of the channels of all the pixels.

Write a method:

```
   private int maxChannel ( Pixel p ) {
```

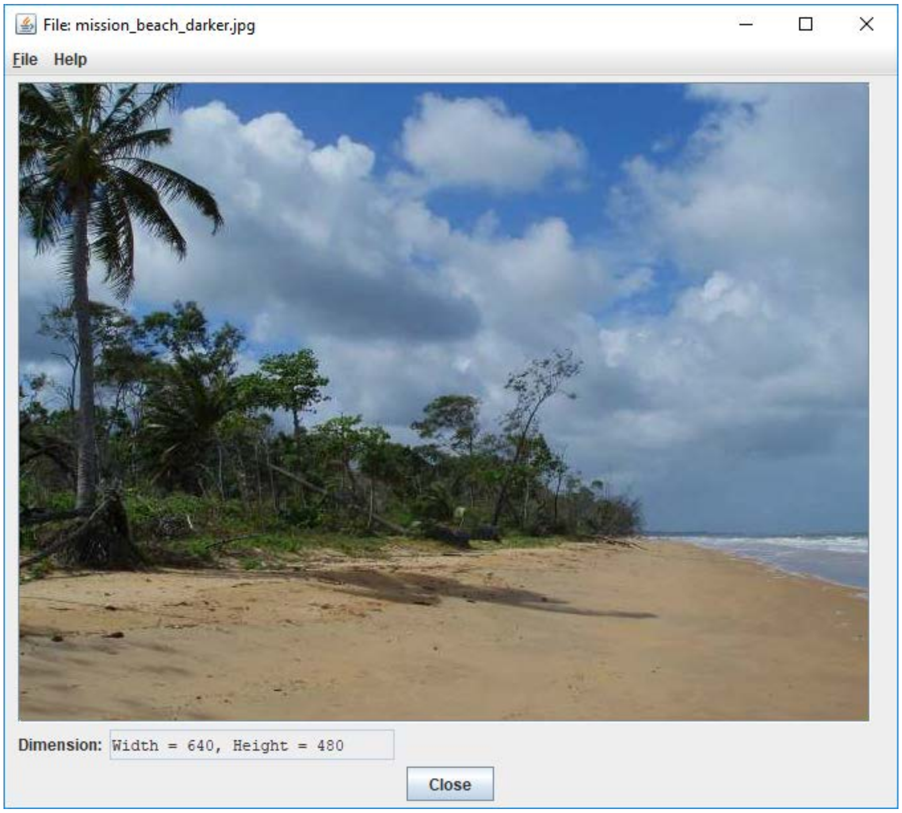
that determines the largest channel value of the three channels (R, G and B) of the pixel `p` and returns that value.

Write a method

```
   private void normIntensity ( Picture aPic ) {
```

which normalizes the intensities of the channels of all pixels by finding the maximum intensity of any channel, determining the factor necessary to raise this intensity to 255, and then multiplying each channel value of every pixel by this value.

The result of applying this program to the picture above is:



Dimension: Width = 640, Height = 480

Submission:

Details regarding preparation and submission of assignments in COSC 1P02 are found on the COSC 1P02 Sakai Site as `Assignment Guidelines` under `Course Documents`. This document includes a discussion of assignment preparation, programming standards, evaluation criteria and academic conduct (including styles for citation) in addition to the detailed assignment submission process copied below.

To prepare and submit the assignment electronically, follow the procedure below:

1. Ensure your submission folder (`Assign_4`) has two subfolders (`Assign_4_A` and `Assign_4_B`) that contain the two DrJava projects for the assignment.
2. Using DrJava, print (as a pdf file, e.g. using “printer” `Microsoft Print to PDF` or similar) the `.java` files of each of the parts for your assignment using the name *`ClassName`*`.pdf` where *`ClassName`* is the class name (i.e. same name as the `.java` file) and save the `.pdf` files at the **top level** of the submission folder (i.e. directly within `Assign_4`).
3. Run each program. For Part A use the file: `mission_beach.jpg` as input. When the display is finished (i.e. `Close` button visible), select `Print Image of Window...` from the `File` menu on the `PictureDisplayer`, direct the output to `Microsoft Print to PDF` and save the `.pdf` file at the **top level** of the submission folder (i.e. directly within `Assign_4`) using an appropriate name (e.g. `Part_A.pdf`). Similarly, for Part B, use the file `mission_beach_darker.jpg` as input. When the display is finished (i.e. `Close` button visible), select `Print Image of Window...` from the `File` menu on the `PictureDisplayer`, direct the output to `Microsoft Print to PDF` and save the `.pdf` file at the **top level** of the submission folder (i.e. directly within `Assign_4`) using an appropriate name (e.g. `Part_B.pdf`).
4. Create a `.zip` file of your submission by right-clicking on the top level folder (i.e. `Assign_4`) and selecting `Send to/Compressed (zipped) folder`. A zipped version of the folder will be created. Use the default name (`Assign_4.zip`).
5. Log on to Sakai and select the COSC 1P02 site.
6. On the `Assignments` page select `Assignment 4`. Attach your `.zip` file (e.g. `Assign_4.zip`) to the assignment submission (use the `Add/Remove Attachments` button and select `Browse`). Navigate to where you stored your assignment and select the `.zip` file (`Assign_4.zip`). The file will be added to your submission. Be sure to check the `Honor Pledge` checkbox. Press `Submit` to submit the assignment. You should receive a confirmation email.

DrJava

The `.zip` folder you submit should contain the project folders for the two parts, including all files relevant to the project—the `.java` and `.class` files for the assignment—and the `.pdf` files for program listings and output at the top level.

Other Platforms

If you are using an IDE other than DrJava to prepare your assignment, you must include the `.java` source files and the `.pdf` files described above for each part as well as an executable file (likely `.class` or`.jar`) that will execute on the lab machines.