DHA Suffa University
Object-Oriented Programming (CS 103)
Fall 2015
Submission Date: Wednesday, 21st October 2015 – 11:59 pm sharp
Submission Details: Programming Assignment Submission Instructions.pdf

Assignment #03

Note: Make sure you submit your own work. Any form of collaboration will not be tolerated and case will be forwarded directly to the Disciplinary Committee or in the best case, -100 policy (Remember?). You may discuss the algorithm logic with your friends/classmates verbally OR visually, but not share your code.

Giggle is hiring young bright programmers like you for its digital image processing tool Picasso. They want you to demonstrate your programming skills by designing an image processing tool.

Your program takes as input images; if you open these images from Notepad++ (https://notepad-plus-plus.org/download/) you see some integer values. These values correspond to the some integer value between 0 and 255; where 0 means black, and 255 means white.

You are given an image file, **apollo.pgm**. To view the image use **Irfanview viewer**. Download it from http://www.irfanview.com/

The detail of the file format of the pgm images is in **Pgm.txt** (also here: http://netpbm.sourceforge.net/doc/pgm.html); so read them to understand the format of pgm files.

Define the functions declared in the file **imageproc.h**. You must define these in a separate file **imageproc.cpp**. Write your own driver file to test the functions. However you are only required to submit the file **imageproc.cpp.**

You can assume the file is in the current folder of your code. The modified files are to be saved in the current folder with appropriate prefix e.g. after **shrink2x** is called for **apollo.pgm** it is saved as **apollo-shrink2x.pgm**.

Useful Tips:

- You MAY NOT hard code the size of the array; it should be dynamic – read from the image file
- You can convert images from other formats to the pgm format by opening them in Irfanview and then saving it with pgm extension

- **invertColors:** changes the low intensity pixel to high, and vice versa. Hence, black becomes white, and vice versa.
- **shrink2x:** reduces the size of the image to half i.e. now the given image will become of size 300 by 300
- **enlarge2x:** doubles the size of the image i.e. now the image will become of size 1200 by 1200 (from 600 by 600)
- **brighten:** increases the brightness of the image by 10%. (Note: a pixel with value 150 is more brighter than a pixel having value 100)
- **darken:** decreases the brightness of the image by 10%. (Note: a pixel with value 150 is more brighter than a pixel having value 100)

- **crop:** takes 4 extra values: x1, y1, x2, y2.
    - x1 and y1: are the row and column number from where the image has to be cut.
    - x2 and y2: are the row and column number till where the image has to be cut. The cut image is not stored in an array but is stored to a file. For clarification see the image below

Original Image	Image After crop
	

Best of luck!