Due: Thursday, March 2, 2017, 11:59:59 Midnight

Introduction

Today's lab is designed to give you practice using some of the C++ skills we have discussed in class. It will also help strengthen your problems solving and/or critical thinking skills.

Lab Objectives

- Practice defining and implementing a class
- Practice using c++ style file pointers
- Practice using getters, setters, and constructors.
- Practice creating a ppm image
- Practice problem solving skills
- Practice creating and using client functions.
- Practice passing an instance of a class to a client function.
- Practice using command line arguments
- Practice using multiple files.

Prior to Lab

- Review the notes we have gone over the last several days.

Instructions

You are going to create a class called Color. Color will be designed to represent the RGB values for of a color. You will create three files. Color.h, Color.cpp, and driver.cpp.

Color.h will contain the declaration of the Color class. You are required to provide constructors, setters, and getters for the class. (I will provide this for you.)

Color.cpp will contain the implementation of the functions declared in the Color.h file.

driver.cpp

To test your class, you will write a driver that will:

1. Create **two** OUTPUT file pointers. Open the file pointers, taking care to make sure they both open successfully. Both files will be provided by command line arguments.

2. Provide **two** instances of **Color**.
   a. One should call the default constructor.
   b. One should request the RGB values from the user and then call the overloaded constructor.

3. Create a client function that will produce a square image like the following. You may choose the size of the width and height.

4. Create a second function that will produce a rectangle like the second image below. Again, the width and height is up to you.

The colors in the above images are the official Clemson purple and orange. Feel free to look up the color pallet for the Clemson and use them if you wish, otherwise use two other colors.

FORMATTING

1. Your program should be well documented
2. Each file should include a header (example below)
3. Your program should consist of proper and consistent indention   Ex. You should choose a specific number of spaces to indent – 3, 4, or 5 – and be consistent
4. No lines of code should be more than 80 characters
5. Variable names should be meaningfull
6. You should provide a header in each of the files submitted.

5 – 10 points will be deducted for each of the above formatting infractions.

```
/******************
  your name
  username
  Lab 4
  Lab Section:
  Name of TA
*****************/
```

Submission Instructions
Use handin (http://handin.cs.clemson.edu) to submit a tarred file called Lab7.tar.gz containing driver.cpp, Color.cpp, Color.h