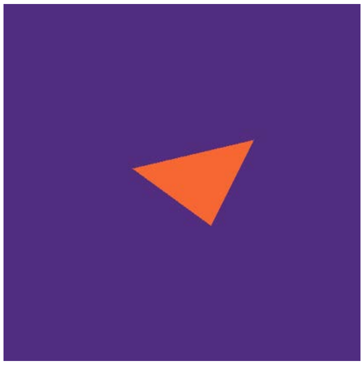
Composition, Composition, and More Composition
Programming Assignment 2
CPSC 1020, Spring 2017
Due: Sunday March 19, 2017 **THIS IS A LITTLE LESS THAN 2 WEEKS. YOU SHOULD START EARLY.**

Program Overview:
This program will read, from a file, the width and height of an image and three points that define a triangle. The program will then loop through the x and y values checking each to determine if point is within the parameter of the three points that define the triangle. If the point is within the parameter of the defined triangle the pixel will be set to a one color, otherwise the pixel will be printed as another color. An example is below.



Learning Objectives:
This assignment will give you practice working with multiple files, as well as, multiple classes. This assignment is designed to give you practice with multiple levels of composition. Below you will find a diagram of the levels of composition:

The concepts covered in the assignment are:
File i/o
Implementing basic classes
Multi-level composition
Working with multiple files
Command line arguments
And more--

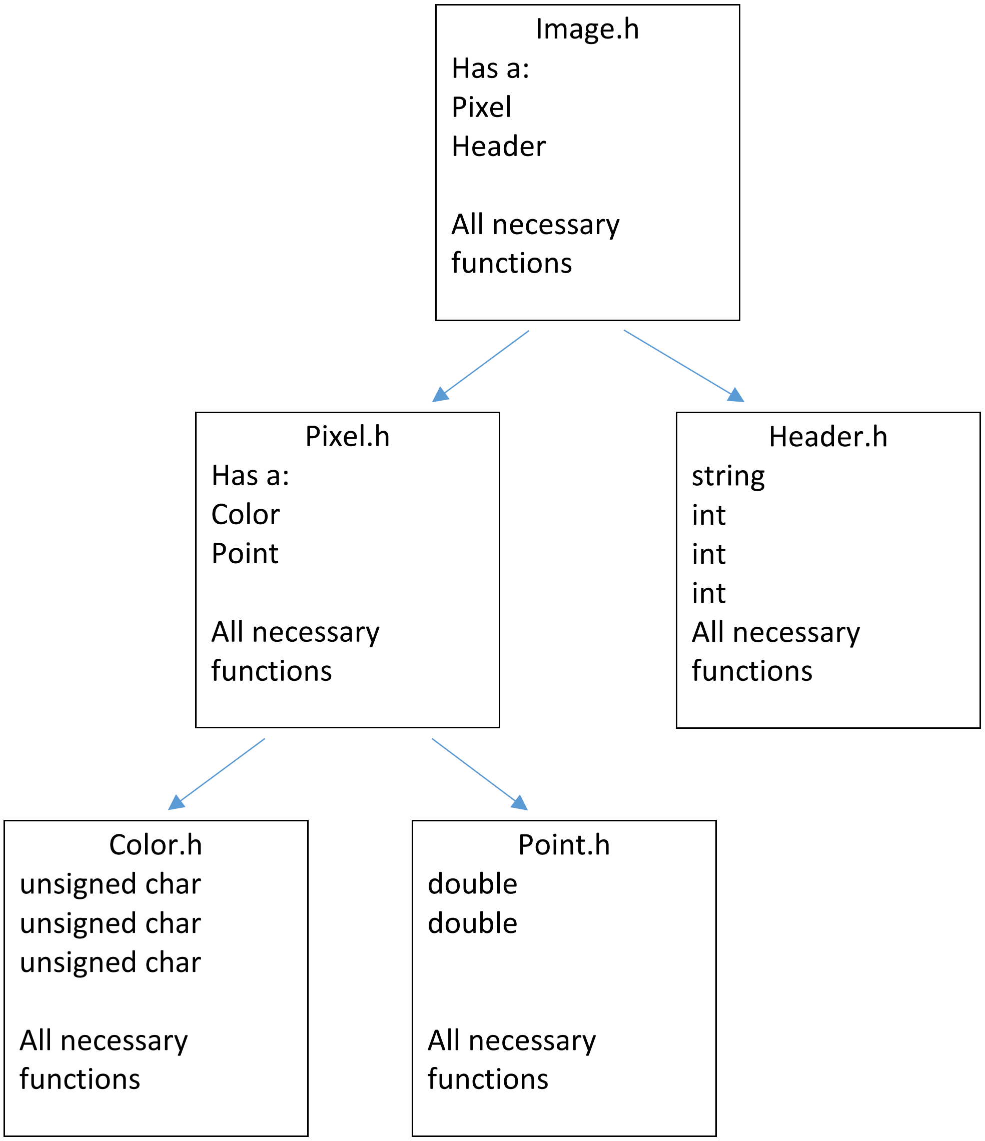


Image.h
Has a:
Pixel
Header

All necessary
functions

Pixel.h
Has a:
Color
Point

All necessary
functions

Header.h
string
int
int
int
All necessary
functions

Color.h
unsigned char
unsigned char
unsigned char

All necessary
functions

Point.h
double
double

All necessary
functions

I will provide you with the basics of my driver. I will delete parts of the driver. However, I will leave detailed comments to help you fill in the deleted code from the driver. This is necessary because you may name your functions differently than I did. As well as your design may differ.

Below you will find a description of how I implement my classes. You are not required to use this outline. You may choose to implement the classes the way you wish. However, you may not change the outline of how the classes are composed (shown above).

Image.h

This class has two private variables:

1. A Pixel
2. A Header

The class has the following public functions:

1. A function to set the header
   This function used the instance of header provided as a private data member to call the headers set functions.

2. A function to print the header
   This function should use the instance of header, provided as a private data member to call the header function that prints the header to the output file.

3. A function to print the color of the Pixel
   This function uses the instance of Pixel to call the Pixel class's function that prints the RGB values of the pixel to the output file.

4. A function to set the RGB values of a pixel
   This function uses the instance of Pixel to call a function in the Pixel class that sets the RGB values. It passes in 3 unsigned chars

5. An overloaded function used to set the RGB values of a pixel
   This function uses the instance of Pixel to call a function in Pixel's class that sets the RGB values. The color depends on a bool that is passed to the function. The bool is a value returned from a client function in the driver that returns true if the current point is within the parameters of the defined triangle. (this is the function you wrote in the lab)

Header.h

This class has 4 private variables:

1. A string to represent the magic number in the ppm header
2. Three integers that represent
    a. Width of the ppm image
    b. Height of the ppm image
    c. Maximum value of the RGB channels

The class has the following public functions:

1. A function to set the value for the magic number of the ppm image.
2. A function to set the value of the width of the ppm image.
3. A function to set the value of the height of the ppm image.
4. A function to set the maximum value of the RGB channels of the ppm image.
5. A function that prints the header for the ppm image to the ouput file.

Pixel.h

This class has two private variables:

1. A Color
2. A Point

The class has the following public functions:

1. A function to uses the instance of Color to call a function of the Color class that sets the color of the pixel. This function has one parameter a bool. If the bool is true, the color is set to one color and if false the color is set to another color.

2. Another function that uses the instance of Color to set the color of the pixel. This function has three parameters that represent the RGB values.

3. A function that uses the instance of Color to print the RGB values to the output file.

Point.h

This class has two private variables:

1. A double to represent the x coordinate of the point.
2. A double to represent the y coordinate of the point.

This class has the following public functions:

1. A function to set the x coordinate and y coordinate of the point.
2. A function that can be used to access the x coordinate.
3. A function that can be used to access the y coordinate.

Color.h

This class has three private variables:

1. An unsigned char to represent the red channel of the pixel.
2. An unsigned char to represent the green channel of the pixel.
3. An unsigned char to represent the blue channel of the pixel.

This class has four public functions:

1. A function that sets the red, green, and blue channels of the pixel.
2. A function that can be used to access the red channel of the pixel.
3. A function that can be used to access the green channel of the pixel.
4. A function that can be used to access the blue channel of the pixel.

driver.cpp

The driver creates and opens two files, one to read and one to write.   It tests whether the files opened successfully or not.

The driver has a client function that determines if a given test point is within the parameters of three given points. NOTE: This is the function you wrote in a lab.
The driver has an array of Points to hold the three points that define the triangle and one Point that represents the x and y coordinates being tested.

The driver has an instance of Image.

The driver uses the the input file pointer to read in the width and height of the image. Using the instance of image call the function created to set the width and height of the header and print the header information to the output image. The driver also reads in the three points used to define the parameter of the triangle and calls the appropriate function to set the x and y coordinates of the points.

Because two files were opened these files must be closed.

I have provided a basic outline of my files. Again feel free to use this outline to create your files or develop your own functions. However, you **MUST** use the above described composed classes. Let me remind you this assignment is designed to give you a better understanding of how classes and functions interact using the “has a” relationship.

REMINDER: You should have an instance of Image in your driver that should be used to basically creates the image. In the driver you are allowed to create instances of a Point to define the parameter of the triangle, but there is no reason to have any other class instances in your driver.cpp.

Other Instructions:

I will provide a makefile. Type **make** and your program should compile.

You should provide a README that consist of the following.

1. A short description of any problems you encountered when writing this program.
2. How you solved the the problems you encountered.
3. Your thoughts on the assignment. This is your opportunity to tell me if you like the assignment or not. What you did or did not like about the assignment. Anything you want to tell me.

Formatting:
You will need to add a header to each of your files similar to the following:

```
/*************************
 *Your name
 *CPSC 1020 your section, Sp17
 *Your user name
 *************************/
```

Your program should compile with no warnings and no errors. There will be a deduction up to 40 points if your program does not compile. If your program compiles but has warnings, there will be a deduction up to 10 points.

- Your code should be well documented. (comments)
- There should be no lines of code longer than 80 characters.
- You should use proper and consistent indention.

- Variable names should be meaningful.

There will be a 5 point deduction for each infraction of the above formatting rules.

WARNING: YOU HAVE A LITTLE LESS THAN 2 WEEKS TO COMPLETE THIS ASSIGNMENT. I WILL HAVE LIMITED INTERNET ACCESS ON MARCH 18 AND 19. SO DON'T EXPECT A LOT OF HELP FROM ME ON THOSE DAYS. PLEASE START EARLY.   WE THE EXECPTION OF THOSE TWO DAYS I WILL BE AVIALABLE IN MY OFFICE AND BY EMAIL.

Handin:
Use handin.cs.clemson.edu to submit your files. I have created a bucket named PA2.
Things to do prior to handing in your files:

1. Test your program on the SoC servers. I will not accept the excuse "It compiled on my computer." I test programming assignments on the SoC servers.
2. Run "make clean" to remove the executable and all .o files
3. Tar zip your files naming the tarred file PA2.tar.gz if you are doing the Extra Credit name your file PA2EC.tar.gz   Your tarred file should include the following:

driver.cpp, Image.h, Image.cpp, Pixel.h, Pixel.cpp, Point.h, Point.cpp, Color.h, Color.cpp, Header.h, Header.cpp, makefile, READMe

5 POINTS EXTRA CREDIT:

Create an image that has two triangles one that over lays the other. The triangles may not have a common side.   An example is shown below. Be creative. You may rotate one of the triangles as long as they overlap and do not share a side.