CIS 441/541: Project #1F
Due February 19, 2019 (which means 6am February 20)
Worth 8% of your grade

Instructions:

You will add shading to your program and also generate a movie.

1) You can use the same reader routine from 1E. HOWEVER: you must add "#define NORMALS" at the top of your project1F.cxx file. This will enable "#ifdef NORMALS" commands in the GetTriangles function.
2) NOTE: there is a new data member, normal, for the Triangle class.

```
class Triangle
{
 public:
    double      X[3];
    double      Y[3];
    double      Z[3];
    double      colors[3][3];
    double      normals[3][3];
};
```

   Normals is indexed by the vertex first and the dimension second.

```
 int vertexId = 0;
 int x = 0, y = 1, z =2;
 normals[vertexId][y] = …;
```

   Note: I also added a "double shading[3];" data member to Triangle. I found this to be a helpful location to store per-vertex shading information.

3) Download the file shading.cxx. This file defines a data structure that contains the parameters for shading. I pasted the contents of this file into my code, and encourage you all to do the same.
4) Extend your code to do Phong shading. Use two-sided lighting for the diffuse component, but only one-sided lighting for the specular component.
5) The correct image for GetCamera(0,1000) is posted to the website.

When you are done upload the following to Canvas:

- A tarball with 3 files:
    - your code (named project1F.cxx). This code should only produce the image for GetCamera(0,1000). It should produce an output image called "frame000.png".
    - a screen shot of the differencer program tell you that frame000.png had 100 pixels or less difference.

    - a text file containing a link to a movie you generate. This movie must be posted to a website (YouTube, ix.cs.uoregon.edu/~<yourname>, or something else).

Note: incorrect images are likely to earn less than half credit. I'd rather have correct submissions late than incorrect submissions on time.

My implementation notes:

- my first step was to add shading as a data member to Triangle.
- I added a fake function that would calculate the shading for a vertex. The function returned 0.5.
- I then added code to LERP the per-vertex shading as I did scanlines, and to modify the output color for a fragment using the shading info.
- I then tested and confirmed it looked right.
- After all of that worked, I implemented the shading equations.

Movie encoders: I imagine most will use ffmpeg. I used mpeg2encode, since I can access it easily through other software I use.

Grading rubric:

- Tarball with everything correct: 5.5 points
- Movie on a website: 2.5 points

(Part of this assignment is learning a movie encoder … install software, learn to use it, etc. If you want to skip that, you will lose 2.5 points.)