```
#pragma once


#include<stdio.h>
#include<string>
#include<iostream>
#include<fstream>

#include<GL/glew.h>

class Shader
{
public:
    Shader();

    //this reads file and pass to compileShader();
    void createFromString(const char* vertexCode, const char* fragmentCode);
    //little bit code in shader here 40:21 vid clearn up code
    void createFromFiles(const char* vertexLocation, const char* fragmentlocation);

    std::string ReadFile(const char* fileLocation);

    GLuint GetProjectionLocation();
    GLuint GetModelLocation();


    void useShader();
    void clearShader();

    ~Shader();

private:
        GLuint shaderID, uniformProjection, uniformModel;

    void compileShader(const char* vertexCode, const char* fragmentCode);
    void AddShader(GLuint theProgram, const char* shaderCode, GLenum shaderType);
};


```