With the OpenGL examples for this class we use one GLU function (gluErrorString) which returns an error string corresponding to the error code produced by OpenGL. GLU is simply some software that was written to help make some things in OpenGL easier to do. Many functions in it are legacy but it is still a part of the OpenGL distribution. If you want to remove the dependency on GLU you can replace our `printOpenGLErrors()` function with the following two methods. You will, of course, need to add the `gl_error_string` header to the declaration for the GraphicsEngine.

```
void GraphicsEngine::printOpenGLErrors()
{
    GLenum errCode;

    while ((errCode = glGetError()) != GL_NO_ERROR)
        fprintf(stderr, "OpenGL Error: %s\n", gl_error_string(errCode));
}


char const* GraphicsEngine::gl_error_string(GLenum const err)
{
    switch (err)
    {
    case GL_NO_ERROR:
        return "GL_NO_ERROR";

    case GL_INVALID_ENUM:
        return "GL_INVALID_ENUM";

    case GL_INVALID_VALUE:
        return "GL_INVALID_VALUE";

    case GL_INVALID_OPERATION:
        return "GL_INVALID_OPERATION";

    case GL_STACK_OVERFLOW:
        return "GL_STACK_OVERFLOW";

    case GL_STACK_UNDERFLOW:
        return "GL_STACK_UNDERFLOW";

    case GL_OUT_OF_MEMORY:
        return "GL_OUT_OF_MEMORY";

    case GL_TABLE_TOO_LARGE:
        return "GL_TABLE_TOO_LARGE";

    case GL_INVALID_FRAMEBUFFER_OPERATION:
        return "GL_INVALID_FRAMEBUFFER_OPERATION";

    case GL_CONTEXT_LOST:
        return "GL_CONTEXT_LOST";

    default:
        return "Unknown Error";
    }
}
```