```glsl
#version 330 core

// Cook-Torrance Shader


// uniform data
uniform mat4 uProjectionMatrix;
uniform mat4 uModelViewMatrix;
uniform vec3 uColor;

// I created these---------------------
uniform float u_ambient_coeff;
uniform float u_diffuse_coeff;
uniform float u_specular_coeff;
uniform float u_power_specular_coeff;
uniform vec3 u_light_position_coeff;
//-------------------------------------

// mesh data
layout(location = 0) in vec3 aPosition;
layout(location = 1) in vec3 aNormal;
layout(location = 2) in vec2 aTexCoord;

// model data (this must match the input of the vertex shader)
out VertexData {
    vec3 position;
    vec3 normal;
    vec2 textureCoord;
} v_out;

void main() {
    // transform vertex data to viewspace
    v_out.position = (uModelViewMatrix * vec4(aPosition, 1)).xyz;
    v_out.normal = normalize((uModelViewMatrix * vec4(aNormal, 0)).xyz);
    v_out.textureCoord = aTexCoord;

    // set the screenspace position (needed for converting to fragment data)
    gl_Position = uProjectionMatrix * uModelViewMatrix * vec4(aPosition, 1);
}
```