# Computer Graphics Coursework – Self Assessment Document

**Name:** *Your Name* **ID number:** *12345678*

Complete the self-assessment grid below by writing a short explanation of how you have satisfied the requirement and how it has implemented in your code. For example:

|  |  |  |
| --- | --- | --- |
| **Grade Band** | **Requirement** | **Explanation** |
| **40-49%** | Create a window in OpenGL. | *A GLFW window object is created using OpenGL which terminates on the press of the escape key. The code for this can be found inside the main() function in the main.cpp source file (lines 19 – 40).* |

## Self-assessment Grid

|  |  |  |
| --- | --- | --- |
| **Grade Band** | **Requirement** | **Explanation** |
| **70 -100%** | LO1: Creation and implementation of your own library for the model, view and projection calculations (you may use glm::mat and glm::vec functions). |  |
| LO2: Models imported using external object files. |  |
| LO2: The user can change viewing modalities, e.g., control the camera using different navigation modes. |  |
| LO3: Use of shaders to apply parameter driven affects within the scene, e.g., night and day transition. |  |
| LO3: Use of normal maps. |  |
| **60 – 69%** | LO1: Use of quaternions for axis-angle rotation calculations. |  |
| LO2: Interactive dynamic aspects of the virtual world controllable by the user using keyboard and mouse (e.g., position of objects, location of light sources etc.). |  |
| LO3: Use of multiple light sources with differing attenuation. |  |
| **50 – 50%** | LO1: Use of Euler angles for axis-angle rotation calculations. |  |
| LO2: Basic user input driven interaction using keyboard and mouse. |  |
| LO3: Use of shaders to apply dynamic lighting. |  |
| **40 – 49%** | LO1: Use of a math library (e.g., glm) to calculate view, model and projection matrices.  *Not required if you have written your own maths library.* |  |
| LO2: Models are hardcoded within the application.  *Not required if you have imported the models.* |  |
| LO3: Use of shaders to apply appropriate textures to objects. |  |