2.5 Marking Scheme – Base Assignment (up to 100 marks max)

Max 90 marks:

Step 1 – An Empty Window	Max	Mark
Code submitted compiles without errors	4	4
Correctly set the clear colour	2	2
Correctly enable back-face culling	2	2
Correctly clear the depth and colour buffers	2	2
Step 2 – A Red Triangle		
Correctly pass vertex positions to OpenGL	5	5
Correctly issue draw commands in Triangle	2	2
Implement SimpleShader.vert to pass positions through	2	2
Implement SimpleShader.frag to set fragment colours red	3	3
Step 3 – Vertex Colours		
Correctly pass vertex colours of the Triangle to OpenGL	3	3
Correctly read colours in VertexColour.vert and pass to fragment shader	5	5
Correctly use vertex colours in fragment shader	4	4
Step 4 – Implement Box		
Correctly pass vertex positions, normals, and colour to OpenGL	4	4
Correctly issue draw commands for Box	3	3
Step 5 –View Transformation		
Generate projection matrix from scene	2	2
Generate view matrix from scene	2	2
Correctly pass matrices to shader as uniform variables	2	2
Correctly transform vertex positions using matrix	3	2
Correctly implement reshape to recalculate the projection	2	2
Step 6 – Model Transformation		
Pass transform data to shader as uniform variable	2	0
Correctly transform vertex position using matrix	2	0
Toon Shading		
Calculate intensity based on angle	4	0
Apply intensity to scene with defined colour modifiers	4	0
Textures		
Specify texture coordinates for box	2	0
Load texture and display Textured Cube	2	0
Basic Lighting		
Pass ambient lighting from scene and use in fragment shader	2	0
Pass point light data to fragment shader as uniform variables	4	0
Implement diffuse phong shading in fragment shader	3	0
SELF ASSESSMENT		
CONPULSORY – Provide a self assessment (complete this table and provide as an additional word file)	5	5

GIT – commit at each step	5	5

We have discussed each of the following concepts at a high level in class. However it is expected you will employ the textbook and online materials to complete these steps.

Specular Lighting	Max	Mark
Add specular lighting to point light	5	0
Multiple Lights	**Max**	**Mark**
Implement spotlight	2	0
Correctly pass lighting data for all lights to shader	5	0
Iterate over lights in fragment shader to shade objects	5	0
Shadow Mapping		
Generate shadow maps by rendering to a Framebuffer object	5	0
Use shadow map during rendering to create shadows	5	0
3D Models		
Implement init and display in Geometry to render 3D models	10	0
Use materials specified in Geometry in shader	5	0
Anti-aliasing		
Implement anti-aliasing	3	0
Texturing		
Implement Texture's init method	5	0
Use texture images for material diffuse colour	5	0
Specify texture coordinates for triangle	2	0
Specify texture coordinates for other shapes (1 per shape)	4	0
Additional Shapes		
Implement Cone	4	0
Implement Cylinder	4	0
Implement Sphere	4	0
Interaction		
Click and Drag	2	0
Keyboard Interactions		
Capture keyboard interactions	2	0
Camera control	4	0
Scene		
Create your own custom scene file	4	0