This paper contains 6 questions each worth 12 marks. Candidates should attempt FIVE out of six questions.

1.

(a) Explain the synthetic camera model, including:

(i) How does it relate 3D space to 2D images? (2)

(ii) How is it relevant to OpenGL? (2)

(iii) How does it relate to a pinhole camera? (2)

(b) What is clipping in the context of OpenGL, and where does it fit in the steps of the OpenGL pipeline architecture? (3)

(c) What is rasterization in the context of the OpenGL pipeline architecture, and what are the inputs and outputs to the rasterization step? (3)

2.

(a) What is the difference between request input mode and event input mode? (3)

(b) What is the purpose of the idle callback in OpenGL? Include an example of what it can be used for. (3)

(c) Give three common applications of vertex shaders in OpenGL. (3)

(d) Why does modern OpenGL use shader programs written in GLSL instead of just allowing the programmer to write their whole program in a single language like C++? (3)
3.

(a) What is the difference between the Phong Model of shading and the Modified Phong/Blinn-Phong Model, and what are their relative advantages?  

(b) Suppose you want to build a program using OpenGL that draws two kinds of objects - shiny billiard balls, and dull wooden balls. How would you implement these in a way that realistically represents the difference in shininess between the two kinds of objects?  

(c) Suppose you have a situation where both the viewer and the only light source are always very distant from the objects in a scene. How could you take advantage of this to perform the lighting calculations more efficiently? Consider both per-vertex and per-fragment lighting.  

(d) Explain the difference between a directional light and a point source light in the context of OpenGL.
4.

(a) Explain clearly what each function call does in the following main program.

```c
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
    glutInitContextVersion( 3, 2 );
    glutInitContextProfile( GLUT_CORE_PROFILE );
    glutCreateWindow("Example");
    glutMouseFunc(mouse);
    glutReshapeFunc(reshape);
    glutDisplayFunc(display);
    init();
    glutMainLoop();
}
```

(b) Why is a fourth component \( w \) added to the usual three dimensional coordinates in computer graphics systems like OpenGL? Give two common uses of the extra entries in the corresponding 4x4 matrices (those in the last row and column).

5.

(a) Consider a horizontal row of fragments that form part of a triangle during rasterization. Suppose the coordinates, normals and texture coordinates of the first and last fragment in the row have already been calculated. How can we quickly calculate the coordinates of the other fragments as we scan along the row from the first fragment to the last fragment?

(b) The painter’s algorithm for hidden surface removal is an alternative to the z-buffer algorithm that instead sorts scene objects by distance and then draws them starting from the furthest away, with each overwriting (or “overpainting”) what has previously been drawn. Describe two situations where this algorithm will not work well.
(c) What is mipmapping? What advantages does it provide?  
(2)

(d) Suppose we want to assign texture coordinates to a spherical object, with the 2D coordinates mapping smoothly onto the 3D surface as much as possible. Describe two different natural ways of doing this, and compare their advantages.  
(3)

6.

(a) What kinds of objects are generally suitable for modeling using subdivision surfaces? Include an example of where using subdivision surfaces would be appropriate, and another of where it wouldn’t.  
(3)

(b) Suppose we apply subdivision surfaces starting with a box shape. What happens to the resulting surface if we add additional vertices near one corner without altering the shape of the box?  
(3)

(c) What are bone weights in the context of animation?  
(2)

(d) In a typical implementation of skinning in OpenGL that takes advantage of shader programs, like that in lectures, say what the main static data is that’s passed to the shaders (i.e., not changing between frames) and what the main dynamic data is that’s passed to the shaders (i.e., changing each frame).  
(4)