1. (5 pts) Explain the difference between world coordinates and model coordinates. You may use the example of a specific object such as a sphere to clarify the difference.

2. (3 pts) The three coordinate axes are oriented horizontally, vertically, and from front to back, relative to the viewpoint and view direction. Which axis is the x-axis, which is the y-axis, and which is the z-axis?

3. (3 pts) Write a `glColor3f()` function call that will set the current color to yellow.

4. (5 pts) Explain the essential geometric difference between a perspective projection and an orthogonal projection in 3D.

5. (5 pts) What is meant by “clipping?”

6. (5 pts) What is meant by the aspect ratio of the window? Be specific.

7. (8 pts) What are the default camera position and orientation? That is, if we do not call `gluLookAt()` to set the camera position and orientation, what will they be?

8. (8 pts) When constructing a surface out of polygonal facets, it is preferable to use triangles rather than polygons with more than 3 vertices. Give one good reason, and with a good explanation, why this is so.

9. (8 pts) If we choose to tessellate a polygon into a set of triangles and then pass to OpenGL the sequence of triangles, it is better to pass it as a “triangle strip.” Why is that?

10. (8 pts) Describe a situation in which you would want to use `glPushMatrix()` and `glPopMatrix()`. Explain how these functions are beneficial to you in this situation.

11. (8 pts) When defining the view frustum, we must specify the distances to the near plane and the far plane. If it were allowed, why would we not want to leave these unspecified, letting near be 0 and far be infinity?

12. (8 pts) When drawing two polygons in the same plane and in overlapping positions, there is an ambiguity as to which polygon should show. This can result in an undesirable fluctuating mixture of the two colors. An example of this is drawing a rectangular window on a rectangular wall. One solution is to draw the window a tiny distance away from the wall, but this is not always satisfactory. Another solution is to “cut” a hole in the wall the exact size and shape of the window.

Describe in sufficient detail how you would tell OpenGL to draw a rectangle with a rectangular hole in it. Keep in mind that OpenGL draws only convex
13. (12 pts) Given that the function `drawCylinder()` draws a cylinder of length 1 and radius 1, with its axis of symmetry lying along the z-axis from 0 to 1, describe exactly the size, shape, and position of the cylinder drawn by the following code.

```plaintext
glTranslatef(3.0, 0.0, 0.0);
glRotatef(-90.0, 1.0, 0.0, 0.0);
glScalef(1.0, 1.0, 2.0);
drawCylinder();
```

14. (14 pts) In 2D graphics, suppose that the viewport is from 0 to `screenWidth` horizontally and 0 to `screenHeight` vertically and that the world coordinates for that window are from `xmin` to `xmax` horizontally and `ymin` to `ymax` vertically. Write equations for the new boundaries (`newXmin`, `newXmax`, `newYmin`, `newYmax`) in world coordinates that will result from changing the window dimensions to `w` by `h` (integers), without distorting the size or shape of the scene and keeping the upper-left corner fixed.