1. What are the fundamental concepts in 3D viewing, and why are they important in computer graphics?
2. Describe the key stages of the 3D viewing pipeline in computer graphics.
3. Explain the significance of 3D viewing coordinate parameters in the rendering process.
4. How is the transformation from world coordinates to viewing coordinates accomplished in 3D graphics?
5. Differentiate between orthogonal and perspective projections in the context of 3D viewing.
6. What role does the viewport transformation play in the 3D viewing pipeline?
7. Can you explain the concept of 3D screen coordinates and their use in computer graphics?
8. What are some common OpenGL 3D viewing functions, and how do they facilitate 3D rendering in OpenGL?
9. How do transformations and projections impact the final appearance of 3D objects on a 2D screen in computer graphics?
10. Could you provide an example of a real-world application where understanding 3D viewing concepts and OpenGL 3D viewing functions is crucial?

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

11. What are the key categories used to classify visible surface detection algorithms, and how do they differ in their approach to solving the visibility problem in computer graphics?
12. Describe the depth buffer method as a visible surface detection technique. How does it work, and what are its advantages and limitations?
13. How does the depth buffer method handle situations where multiple objects intersect or share the same screen space? What challenges may arise?
14. Can you explain the concept of a "Z-buffer" and its role in implementing the depth buffer method for visible surface determination?
15. What are some practical applications or scenarios where the depth buffer method is particularly effective for achieving realistic 3D rendering?
16. In the context of OpenGL, what are some of the primary visibility detection functions or mechanisms that developers can leverage to manage visible surfaces in a 3D scene?
17. How does OpenGL handle complex scenes with multiple objects, and what built-in mechanisms can be used to optimize visible surface detection and rendering performance?
18. Discuss the trade-offs between different visible surface detection methods and their suitability for specific types of 3D graphics applications.

19. In the context of computer game development, how might the choice of visible surface detection method impact gameplay and user experience?

20. Can you provide examples of real-world scenarios where choosing the right visible surface detection method is critical for achieving visual realism and interactive 3D experiences in applications like architectural visualization or virtual reality?