GRAPHICS HARDWARE

1. **Describe** how images are represented in binary.
2. **Define** resolution.
3. **Define** aspect ratio.
4. **Calculate** the aspect ratio from a given resolution.
5. **Describe** the classic image coordinate system.
6. **Describe** the physical structure of a cathode ray tube.
7. **Describe** the operation of a monochromatic cathode ray tube.
8. **State** how beam intensity affects brightness in a cathode ray tube.
9. **Describe** the raster process in a cathode ray tube.
10. **Describe** how a color cathode ray tube extends the monochromatic tube.
11. **Explain** the 3-bit red-green-blue (RGB) color space.
12. **List** the eight 3-bit basic colors and give the RGB tuple for each.
13. **Extend** the 3-bit RGB color space to 24-bit color.
14. **Describe** the gray-scale space in 3-bit RGB color.
15. **Define** the frame buffer.
16. **Calculate** the size of the frame buffer for 24-bit color at any resolution.
17. **Calculate** the pixel time for a 60 Hz CRT refresh with zero refresh delay.
18. **Describe** how color maps reduce storage requirements.
19. **Describe** the LCD panel and its operation as a replacement for CRT tubes.

BASIC GRAPHICS ALGORITHMS

20. **List** the four fundamental graphics primitives.
21. **Define** scan conversion.
22. **Describe** the process to scan convert a pixel.
23. **Describe** the algorithm of direct scan conversion of a line.
24. **Describe** the digital differential algorithm (DDA) for scan conversion of a line.
25. **Describe** the Bresenham algorithm for scan conversion of a line.
26. **Compare** and **contrast** line conversion algorithms in terms of storage and speed.
27. **Extend** the Bresenham algorithm for lines that have slopes between 45 and 90 degrees.
28. **Extend** the Bresenham algorithm for lines that have negative slopes.
29. **Describe** how polygons are defined and stored as a graphics primitive.
30. **Compare** and **contrast** convex and concave polygons.
31. **Compare** and **contrast** positively and negatively oriented polygons.
32. **Describe** the boundary-fill algorithm for region filling.
33. **Describe** the flood-fill algorithm for region filling.
34. **Describe** the scan-line algorithm for region filling.
35. **Compare** and **contrast** region filling algorithm in terms of storage and speed.
36. **Define** typeface.
37. **Compare** and **contrast** serif and sans-serif typefaces.
38. **Define** point and pica for typefaces.
39. **Describe** how bitmapped fonts are defined and stored as data.

40. **Describe** how bitmapped fonts scale as point size when the screen resolution changes.
41. **Describe** how outlined fonts are defined and stored as data.
42. **Describe** simple techniques to apply attributes such as bold and italic to both bitmapped fonts and outlined fonts.
43. **Describe** how grayscale could be applied to an image.
44. **State** why clipping algorithms help to optimize speed when drawing an image into a viewport.
45. **Describe** the Cohen-Sutherland clipping algorithm.
46. **Apply** Cohen-Sutherland to given lines and viewports.

GEOMETRIC TRANSFORMATIONS

47. **Write** the 2-D transformation matrix in 3x3 form.
48. **Write** the 2-D rotation matrix in 3x3 form.
49. **Write** the 2-D scaling matrix in 3x3 form.
50. **Write** the 2-D mirror matrices in 3x3 form.
51. **Extend** the transformation matrices from 2-D to 3-D in 4x4 form.
52. **Write** composite transformation matrix equations to solve given transformation problems in 2-D and 3-D.
53. **Compare** and **contrast** parallel and perspective projections.
54. **List** artifacts introduced by projections.
55. **Write** the parallel projection matrix in 4x4 form.
56. **Write** the perspective projection matrix in 4x4 form.
57. **Describe** the viewing pipeline.
58. **Describe** how world coordinates are moved to device coordinates.
59. **Describe** the final steps needed to move device coordinates onto the computer screen due to the classic placement of the screen origin.
60. **Calculate** the inverse composite transformation to move a new mouse-click from device coordinates back to world coordinates.

C++ BASICS

61. **Describe** the C++ encapsulation keywords and the implications for both parent and child.
62. **List** the five classic function sets that all C++ classes should have written as part of their creation.
63. **Describe** C++ constructors: default, parameterized, default parameter values.
64. **Describe** C++ destructors**:** default, user-provided destructors.
65. **Describe** C++ copy constructors**:** declaration, return type.
66. **Describe** the C++ assignment operator= function: declaration, return type, preventing self-assignment.
67. **Describe** how C++ passes parameters by value and by pointer reference: declaring function parameters, implications of variable value changes.
68. **Describe** the C++ used of namespaces to identify object function sets.

69. **Describe** the C++ iostream insertion and deletion operators: declaration, use in classes.
70. **Describe** the use of the **new** and **delete** operators for dynamic memory management.
71. **Describe** C++ inheritance: base classes, abstract base classes, call parent constructors, setting variables in parents and children.
72. **Describe** C++ polymorphism: virtual functions, base class pointer collections, accessing correct functions via the base class pointer collection.
73. **Describe** examples of the C++ standard template library.
74. **Describe** reading and writing to C++ text and binary files.
75. **Describe** the use of the **this** pointer in C++.