Coverage: Everything we’ve covered in the PowerPoints, ch. 1 through projections in ch. 5. Homework.
Format: short answer (e.g. “Explain the Depth Buffer algorithm for hidden surface removal”), problem solving, e.g. “use the dot product method to determine the cosine of the angle between <1,2,3> and <1,1,1>”

1. Chapter 1.
   a. Terms: scan line, resolution, horizontal and vertical retrace, refresh, refresh rate, interlace, NTSC
   c. Color. R, G, B, A. Why do we use three colors?
   d. Ray tracing: algorithm to ray-trace diffuse illumination, specular, shadows, transparency, refraction, fog, reflections, anti-aliasing. Drawbacks of ray tracing?
   e. Ambient, diffuse, specular types of illumination
   f. Textures. Procedural textures. Using textures to make clouds, mountains, etc.
   g. Difference in the way dielectrics and conductors reflect light. How do you make a material look like metal?
   h. Radiosity: algorithm? What are form factors? What are the benefits/drawbacks to radiosity?
   i. Graphics pipeline: transformations, clipping, projection, rasterization. What does each step do? What goes into and out of each step? How are stages of the pipeline implemented nowadays?

2. Chapter 2:
   a. Sierpinski algorithm—point and polygon versions

3. Chapter 3: Input and interaction
   a. Input devices: event mode vs. sample mode
   b. Event-loop programming
   c. State tables for UI events
   d. Rubber banding algorithm
   e. Double buffering and Canvas
   f. Convolutions. Kernels for blurring, sharpening, edge detection
   g. JPEG compression: basic steps

4. Chapter 4: Geometric objects and transformations
   b. Affine sum of points or vectors.
   c. Dot and cross products. Know how to compute a dot or cross product of given two vectors such as <1,2,3> and <2,4,0>.
   d. Computing the angle between two vectors using a dot product or cross product. When should I use which?
   e. How to compute face outward normals?
   f. How to tell if two line segments intersect using dot and cross products?
   g. What are homogeneous coords? Why do we use them?
   h. Show a homogeneous transformation matrix that would translate by (dx, dy, dz) or scale by 2 in X dimension.
   i. Why does it appear that transformation function calls are made in reverse order in some cases?
   j. How to rotate around a point (1,2) in the plane using translations and a rotation around the origin.
   k. Given a bunch of points read in from a file, how would I normalize them to the positive unit cube?
   l. Given two mouse clicks, how would I construct the corresponding rotation using the virtual trackball metaphor?
   m. How could I draw a bunch of faces with hidden surfaces removed (basic algorithm studied in class)?

5. Chapter 5: Viewing
   a. Projections: parallel vs. perspective. COP or DOP. Projectors. Orthographic projections.
   b. View frustum; front and back clipping planes
   c. Orthographic projection. One, two, three-point perspective.
   d. Camera API: Eye point, LookAt point, up vector
   e. Projection API: fov, aspect, near, far; uvn frame (VRP, VUP, VPN)
   f. Perspective projections: similar triangles to compute coordinates.
   g. Perspective matrices, divide by w step
   h. How can you use projections to fake shadows?