CMSC 335: Computer Graphics
Quiz #1

Name: ____________________________ Key: ____________________________

1. (1 point) List three example applications of computer graphics:
   (a) ____________________________ Video games ____________________________
   (b) ____________________________ Computer-aided Design (CAD) ______________
   (c) ____________________________ Edutainment ____________________________

2. (2 points) State the steps of the viewing pipeline in order (refer to each step in terms of its coordinate reference frame):
   (a) ____________________________ Model coordinate frame __________________
   (b) ____________________________ World coordinate frame __________________
   (c) ____________________________ Viewing/Projection coordinate frame _______
   (d) ____________________________ Normalized coordinate frame _____________
   (e) ____________________________ Device coordinate frame ________

3. (2 points) Typically, we represent models as ________ Polyhedra ________ with subcomponents (facets) as ________ Triangles ________ or ________ Quadrilaterals ________. With each vertex/facet we associate a ________ Normal vector ________ to define its “facing direction.”

4. (2 points) We construct a computer generated image in a special location in memory called the ________ Frame buffer ________. The total size of the memory is referred to as its ________ Resolution ________. Each cell of the memory is called a ________ Pixel ________. The size of each cell in memory is called its ________ Depth ________.

5. (1 point) __________ True ________ False (Circle one): Direct X is an industry standard, proprietary graphics library in C++ developed by Microsoft.

6. (2 points) Line drawing. Fill in the following table based on Bresenham’s line algorithm noting both the value of the decision parameter at each step and the pixel filled in each step of the algorithm. Assume the first point of the line is already filled. Use \((x_0, y_0) = (8, 7)\) and \((x_{\text{end}}, y_{\text{end}}) = (15, 10)\). To help, first fill in the following values:
\[
\Delta y = 3 \quad \Delta x = 7 \quad 2\Delta y = 6 \quad 2(\Delta y - \Delta x) = -8
\]

\[
\begin{array}{ccc}
k & p_k & (x_{k+1}, y_{k+1}) \\
0 & -1 & (9, 7) \\
1 & 5 & (10, 8) \\
2 & -3 & (11, 8) \\
3 & 3 & (12, 9) \\
4 & -5 & (13, 9) \\
5 & 1 & (14, 10) \\
6 & -7 & (15, 10) \\
\end{array}
\]
7. (2 points) The general approach for determining pixel coverage by polygons is the Scan-line polygon-fill algorithm, which is based on reasoning about edge intersections. To facilitate determination of edge intersections, we can use an additional data structure, referred to as the Sorted-edge table. This approach is an example of a generic algorithmic paradigm in computational geometry called plane-sweep. The approach for recoloring an area with an irregular boundary is the Flood-fill algorithm.

8. (1 point) Aliasing is the distortion of information due to low-frequency sampling. Name one common technique to address this issue: Supersampling, Area sampling, or Pixel phasing.

9. (2 points) The most basic and inefficient method of drawing a primitive in Open GL is by using Immediates. To alleviate some inefficiencies and “pre-compile” sets of instructions, Open GL offers Display lists. However, the most efficient technique is to use Vertex array objects which create buffers for Open GL to read and interpret memory efficiently with minimal copying. In these, the spatial data of the model’s points is stored in a Vertex buffer object.

10. (1 point) Bonus. There is a problem with the primitive drawing algorithms discussed in class. As a hint, think about the geometric measurements of the objects we are displaying. Describe the issue in terms of Bresenham’s line algorithm and describe how we fix it. Let’s assume the center of each pixel maps to its Cartesian coordinate. The pixel itself takes up a unit space in the Cartesian plane. So a line drawn using Bresenham’s line algorithm will not have the same length as the line in Cartesian space. So, to fix the issue, we shorten the length of our drawn line by one pixel. Described a different way, we fix the issue by drawing only pixels interior to the line itself.