

## CS4202: Test

Name: \_\_\_\_\_

1. Write the letter corresponding to the library name next to the statement or statements that describe library. (4 points)
  - A. GLUT \_\_\_\_\_ contains routines that use lower level OpenGL commands to perform tasks
  - B. GLU \_\_\_\_\_ routines that provide the MS Windows to OpenGL interface
  - C. WGL \_\_\_\_\_ window system independent toolkit
  - D. GLX \_\_\_\_\_ Extension to OpenGL for use only on the Unix operating system.
2. List three characteristics a polygon should have to insure proper rendering by OpenGL. (3 points)  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
3. State the purpose of Normal Vectors in a single sentence. Indicate the proper length of a normal vector for use in OpenGL lighting calculations. (2 points)
4. Write an equation, which gives the definition of aspect ratio. (1 point)

5. Below are descriptions of several objects, which would require numerous polygons to model in a realistic manner. Next to each description, write whether you would assign “averaged” normals to co-located vertices of adjacent polygons or use “distinct” normals for each polygon face. (5 points)

A. An Icosahedron \_\_\_\_\_

B. A ball modeled using an icosahedron \_\_\_\_\_

C. A Geodesic dome \_\_\_\_\_

D. A cereal bowl \_\_\_\_\_

E. A cut diamond \_\_\_\_\_

6. List two types of shading available in OpenGL and describe in one sentence how the color of a polygon is determined under each type. (4 points)

1. \_\_\_\_\_

2. \_\_\_\_\_

7. List the transformations which must be carried out on a vertex to place it in *window coordinates* in the order in which they must appear as OpenGL commands in a graphics application, then list the transformations in the order in which they are actually carried out during execution. (6 points)

8. Complete the below table by checking the characteristics that apply to the two principle types of projection transformations. (6 points)

Projection Type	Preserves Angles	Parallel lines remain parallel	Relative Dimensions remain unchanged	Center of projection at infinite distance	Frustum shaped view volume	Fore-shortening effect
Orthogonal						
Perspective						

9. The function `drawPyramid()` renders a pyramid centered on the current origin. Below are several OpenGL commands that generate transformation matrices. Number the commands to place them in an order that will cause the pyramid to be rendered with the following characteristics. Cross out any unnecessary commands. (6 points)

Global Position:  $X = 5, Y = -3, Z = 12$

Global Orientation: Upside down, pointing in the  $-Y$  direction

```
glMatrixMode( GL_MODELVIEW );
glLoadIdentity();
_____ glRotated(180.0, 1.0, 0.0, 0.0);
_____ glTranslated(5.0, -1.0, 0.0);
_____ glRotated(90.0, 1.0, 0.0, 1.0);
_____ glTranslated(-3.0, 5.0, 12.0);
_____ glTranslated(0.0, -2.0, 12.0);
drawPryramid();
```

10. Complete the below table by check marking the characteristics which are applicable to true type and bitmap fonts. (4 points)

Font Type	Geometric Representation	Fixed Size	Can be rotated, scaled, & translated	Relatively Fast to Render
Bit Map				
True Type				

11. The function `drawEagle()` renders a bird centered on the origin with its “beak” pointing down the x-axis. Below are four OpenGL commands that generate transformation matrices. Number the commands to place them in an order that will cause the bird to be rendered with the following characteristics. Cross out unnecessary commands. (8 points)

Size: bird to be drawn twice normal length, width and height unchanged

Global Orientation: 30 degrees of roll to the left, heading of 075

(+z axis is north)

Global Position: X = 33, Y = 53, Z = -78

```
glMatrixMode( GL_MODELVIEW );
glLoadIdentity();

_____ glRotated(165.0, 0.0, 1.0, 0.0);
_____ glRotated(-165.0, 0.0, 1.0, 0.0);
_____ glRotated(-195.0, 0.0, -1.0, 0.0);
_____ glTranslated(33.0, 53.0, -78.0);
_____ glScaled(2.0, 1.0, 1.0);
_____ glRotated(180.0, 1.0, 1.0, 0.0);
_____ glScaled(2.0, 0.0, 0.0);
_____ glScaled(1.0, 2.0, 1.0);
_____ glRotated(-30.0, 1.0, 0.0, 0.0);
_____ glRotated(30.0, 1.0, 0.0, 0.0);
_____ glRotated(30.0, 0.0, 1.0, 0.0);
_____ glTranslated(33.0, -78.0, 53.0);

drawEagle();
```

12. List the three factors with which the OpenGL lighting model is concerned. (3 points)

---



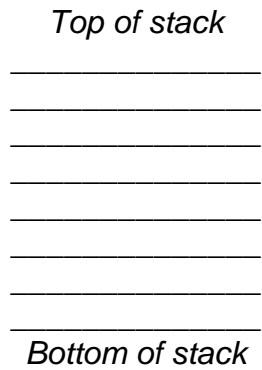
---



---

13. Complete the given picture that illustrates the contents of the modelview stack following completion of the given sequence of OpenGL commands. The symbol for the matrix generated by each command is given as a comment. If a stack matrix is the result of multiplying several matrices together, write a mathematical expression for the matrix i.e. If a matrix on the stack is the result of multiplying matrix  $A$  times  $B$  write  $[A][B]$  (remember order is important). Assume the stack is initially empty. (8 points)

```
glMatrixMode( GL_MODELVIEW );
glLoadIdentity(); // [I1]
glPushMatrix();
glRotated(. . .); //[R1]
glTranslated ( . . . ); //[T1]
glPopMatrix ();
glPushMatrix();
glRotated(. . .); // [R2]
glTranslatef(. . .); // [T2]
glTranslatef(. . .); // [T3]
glPushMatrix();
glRotatef(); // [R3]
glPushMatrix();
glScalef( . . . ); // [S1]
glPopMatrix();
```



14. Complete the following table by check marking the factors that influence the intensity of the various types of “illumination” for a given vertex. (8 points)

Type of illumination	Ambient Material Color	Diffuse Material Color	Specular Material Color	Ambient Light Intensity	Diffuse Light Intensity	Specular Light Intensity	Object Orientation relative to light source	Object Orientation relative to the view point
Ambient								
Specular								
Diffuse								

15. Indicate where you would set the position the following light sources in the code below by writing the name of the light at the appropriate position in the code. (8 points)

GL\_LIGHT0 // positional light which orbits the objects in the scene

GL\_LIGHT1 // positional light 15 units above the center of the scene

GL\_LIGHT2 // positional light fixed to the view point

GL\_LIGHT3 // directional light fixed relative to the entire scene

```

glMatrixMode( GL_MODELVIEW );
glLoadIdentity();
_____
executeViewingTransforms(); // Complete all viewing transformations
_____
glPushMatrix();
glRotatef(. . .);
glTranslate(. . .);
drawObjectA();
glPopMatrix();
_____
glPushMatrix();
glRotatef(. . .);
_____
glPopMatrix();

```

16. Write the basic steps of the depth buffer algorithm. (6 points)

17. Name and describe the primary limitation of the Z buffer algorithm. (2 points)