

## Sceneview Algorithm Questions

Name and Account:

### Question 1

Suppose you want to apply a transformation matrix to some vertices. In what order should you use the following five OpenGL commands?

`glEnd()`, `glLoadMatrix()`, `glMatrixMode(GL_MODELVIEW)`, `glVertex4fv()`, `glBegin()`

### Question 2

Consider the following excerpt from a scenefile:

```
<transblock>
  <translate x="0" y=".5" z="0"/>
  <scale x=".05" y="1.0" z=".05"/>
  <rotate x="1" y="0" z="0" angle="45"/>
  <object type="primitive" name="cylinder">
    <diffuse r="1" g="1" b="1"/>
  </object>
</transblock>
```

To transform the cylinder:  $C$   
into the desired cylinder:  $C'$

In which order would you multiply the three transformations:  
translate (T), rotate (R), and scale (S) to achieve the desired effect?

$C' =$                        $* C$

### Question 3

In question 2 you described how to compose transformations within a single transformation block. When coding Sceneview, you will also have to compose transformations whenever there is an `object tree` block contained within a `trans` block. Consider the following (contrived) excerpt from a scene file:

```
<transblock>
  <rotate x="0" y="1" z="0" angle="60"/>
  <scale x=".5" y=".5" z=".5"/>
  <object type="tree">
    <transblock>
      <translate x="0" y="2" z="0"/>
      <scale x="1" y=".5" z="1"/>
      <object type="primitive" name="sphere">
        <diffuse r="1" g="1" b="0"/>
      </object>
    </transblock>
  </object>
</transblock>
```

Suppose you composed the two transformations in the outer `trans` block, calling the result `CTM1`, and composed the transformations in the inner `trans` block, calling the result `CTM2`. Show the order in which you must multiply these matrices to obtain a single composite matrix with the desired effect on the sphere.

COMPOSITE =

## Question 4

Being sure of the order in which matrices must be multiplied puts you well on your way to completing Sceneview. The other principle hurdle is deciding how you will traverse the parse tree provided by `CS123ISceneParser`.

a) In your most efficient program design, when and how many times should you traverse the original parse tree?

b) Flattening the parse tree makes it quicker and easier to traverse when drawing the scene. What type of data structure will you use for this flattened tree?

c) What information will you store at each of the nodes in the flattened tree? Please give valid types, and descriptions of any types you are defining yourself.