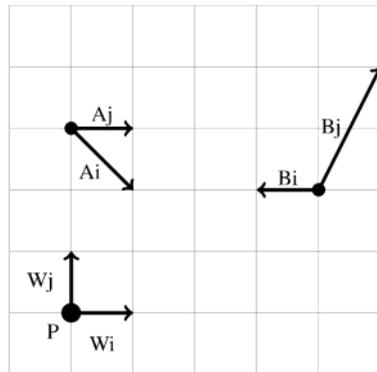




1. (8 pts) The point  $\mathbf{p}$  can be specified as  $\mathbf{p}_W = (0, 0)$  in the coordinate frame  $W$ , with orthonormal basis vectors  $\mathbf{W}_i$  and  $\mathbf{W}_j$ . Specify the coordinates of point  $\mathbf{p}$  in frames  $A$  and  $B$ .



$$P_A(3, -3)$$

$$P_B(3, -1)$$

4pts each

-1 pt sign error  
-1 pt flipped coordinates  
-2pts wrong number

2. (12 pts) True/false

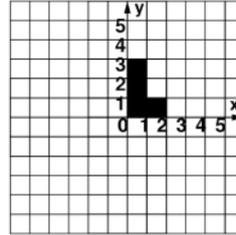
- F (a) The homogeneous points (2,4,8,2) and (2,4,8,5) map to the same Cartesian point after homogenization.  
T (b) A rotation transformation leaves the  $w$  coordinate of a homogeneous point  $(x\ y\ z\ w)$  unchanged.  
F (c) The transformation from an orthographic view volume to the display coordinate system changes the relative ordering of objects in depth.  
T (d) The transformation from a perspective view volume to the display coordinate system may change the relative horizontal and vertical spacing between objects.  
T (e) The transformation from an orthographic view volume to the display coordinate system depends on the width of the viewport.  
T (f) One use case for an asymmetric viewing frustum is stereo viewing.  
F (g) Doubling the distance of the near plane from the camera in the orthographic view frustum will result in a "zoomed-in" larger object in NDCS (normalized device coordinate system).  
F (h) Doubling the distances between the left and right plane of the perspective view frustum will result in a horizontal stretching effect in NDCS.  
T (i) Normals can be safely transformed by the modelling matrix if it contains only translation.  
F (j) Modelling transformations are always linear.  
T (k) Any point outside the orthographic viewing frustum will fall outside the range -1 to 1 in  $x$ ,  $y$ , and  $z$  in NDCS.  
F (l) The projection matrix requires an eye point, lookat point and up vector.

1 pt each

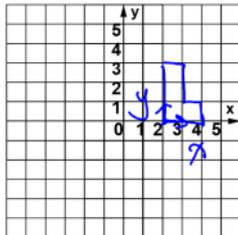
3. (20 pts) Draw shapes 2, 3, 4, and 5 transformed by the pseudocode in the left column below. The result of the first call (shape 1) is shown in the right column.  
 For reference, the pseudocode transformation is  $\text{scale}(x, y, z), \text{rotate}(\text{theta}, x, y, z), \text{translate}(x, y, z)$ . Remember that since the specification is in GL code these computations are done with a moving coordinate frame, corresponding to interpreting equations from left to right.

```

LoadIdentity();
drawShape(); // shape 1
translate(2, 0, 0);
drawShape(); // shape 2
rotate(90, 0, 0, 1);
translate(-1, 0, 0);
drawShape(); // shape 3
scale(2, 2, 1);
translate(0, -1, 0);
drawShape(); // shape 4
rotate(-90, 0, 0, 1);
scale(0.5, 1, 1);
translate(-2, 0, 0);
drawShape(); // shape 5
  
```

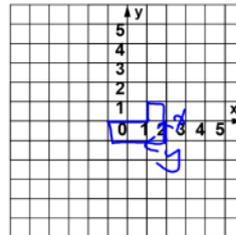


a) shape 2



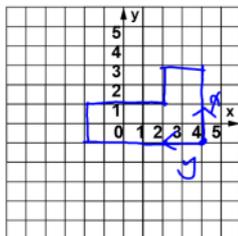
4 pts

b) shape 3



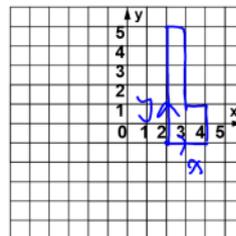
5 pts

c) shape 4



5 pts

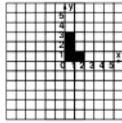
d) shape 5



6 pts

(partial credits if cascading error)

4. (24 pts) For each equation below, sketch the new location  $L'$  of the L shape on the grid, whose original position is shown below. For each equation, on the first row of grids show the incremental computations reading from right to left (moving object). Then on the next line, show the computations considered left to right (moving coordinate frame). You should get different intermediate answers, but check your work to ensure that the final position of the L is be the same in both.



3 pts each. partial credits if cascading error.

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$L' = CBDA L$ .

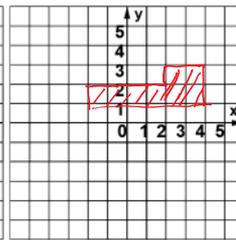
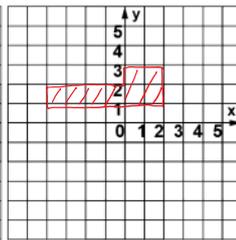
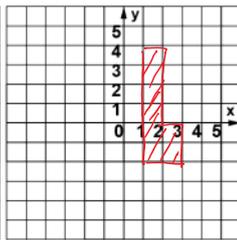
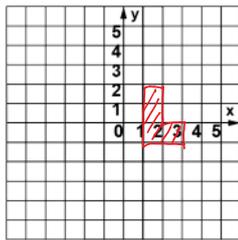
First row: right to left, moving object.

$L' = A L$

$L' = DA L$

$L' = BDA L$

$L' = CBDA L$



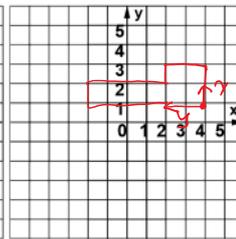
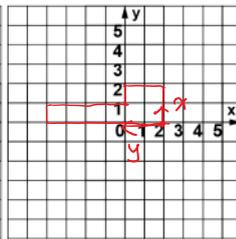
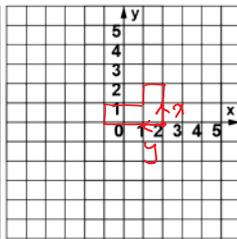
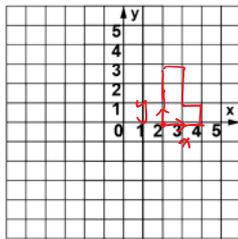
Second row: left to right, moving coordinate frame.

$L' = C L$

$L' = CB L$

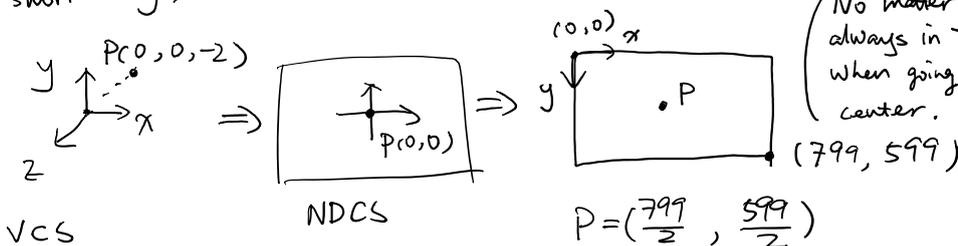
$L' = CBD L$

$L' = CBDA L$



5. (16 pts) A point  $p$  is at location  $(0,0,-2)$  in VCS (view coordinate system), with near = 1, far = 5, bot = -1, top = 1, right = 1, left = -1. Find its  $x$  and  $y$  location in DCS (display coordinate system) with a viewport of width 800 and height 600. Assume  $(0,0)$  is the origin in all coordinates. Show your work/reasons.

<short way>



No matter what projection you use,  $P$  is always in the center of view plane. So when going to NDCS, it will remain in the center.

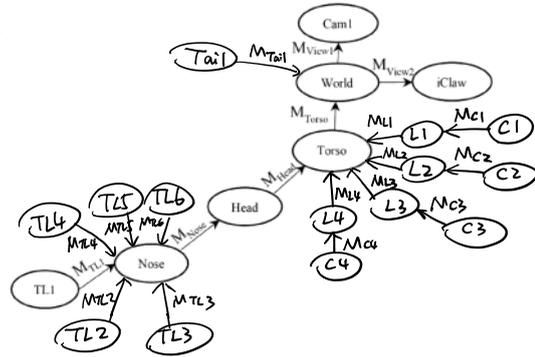
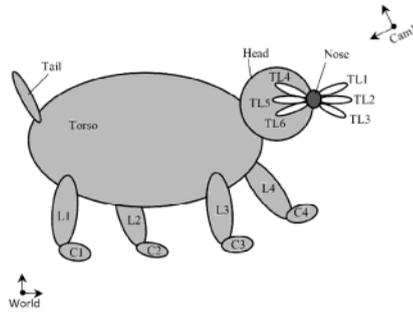
<Long way>

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -3/2 & -5/2 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -2 \\ 1 \end{bmatrix}_{VCS} \xrightarrow{M_p} \begin{bmatrix} 0 \\ 0 \\ 1/2 \\ 2 \end{bmatrix} \xrightarrow{1/W} \begin{bmatrix} 0 \\ 0 \\ 1/4 \\ 1 \end{bmatrix} \xrightarrow{M_{NDCS}} \begin{bmatrix} 399.5 \\ 299.5 \\ 5/8 \end{bmatrix}$$

-1 pt (400, 300) -1 pt per mismatch in long way answer, even if not influencing the final answer. Up to 4 pts.

6. (20 pts) Scene Graphs

6. (20 pts) Scene Graphs



1 pt per branch

-3 pts no transformation labels

- (10 pts) For the mole shown above, complete the scene graph shown on the right. Label the transformation involved, using  $M_A$  to designate the matrix that takes point from coordinate frame A to its parent coordinate frame, etc. Assume that the torso is defined with respect to the world coordinate, and all other body parts are defined with respect to the torso.
- (5 pts) Give the expression for the composite transformation that should be in the modelview matrix to get from viewing to the first tentacle coordinates: that is, the transformation to get from the Cam1 coordinate system to the TL1 coordinate system, or equivalently the matrix that takes a point specified in TL1 coordinates and transforms it to Cam1 coordinates.

$$M_{View1} \cdot M_{Torso} \cdot M_{Head} \cdot M_{Nose} \cdot M_{TL1}$$

-1 pt missing View x Model

-2 pts inverse

-3 pts backward

- (5 pts) Your mole wants to have an iClaw to log its fitness goals while swimming and digging. The iClaw has a camera that lets you see what things would look like from a point of view of the left front claw, C4. (You will feel dizzy when your mole is digging!) Express  $M_{View2}$  in terms of the composite transformation in the scene graph that you complete in the first question. Notice the iClaw frame superimposes on C4 frame. ( $M_{View2}$  is the W2V matrix from the point/object point of view.)

-1 pt include  $M_{World}$

-1 pt include  $M_{View2}$

-2 pts no inverse

-3 pts backward

$$(M_{Torso} \cdot M_{L4} \cdot M_{C4})^{-1} = M_{C4}^{-1} \cdot M_{L4}^{-1} \cdot M_{Torso}^{-1}$$