Name:

Write down answers in-between questions. Please answer using short sentences.

The back of each page can be used for practice, but DO NOT write down the answer on the back.

Be sure to write your student number and name on each page.

1. (4 pts) Write down the 2 x 2 matrix to perform the following transformation (reflection about the y-axis).



2. (4 pts) What is the y-axis vector of the affine frame represented by the following matrix (w.r.t. global frame)?

```
[[1,2,3,4],
[5,6,7,8],
[9,10,11,12],
[0,0,0,1]]
```

3. (10 pts) Match each transformation diagram on the right with the corresponding matrix on the left.



Student Number:

Name:

4. (7 pts) What is the printed result of the following code?

5. (5 pts) Write down the 4 x 4 affine transformation matrix to rotate by an angle θ about the y-axis.

- 6. (7 pts) There are two affine transformations **T** and **S** on n-dimensional spaces. Their linear parts are represented by n x n matrix \mathbf{M}_t and \mathbf{M}_s , and their translational parts are represented by n x 1 column vectors \mathbf{t}_t and \mathbf{t}_s . The question is:
 - 1) Let's say you have a n-dimensional point \mathbf{p} . What is the new point \mathbf{p}' from which the point \mathbf{p} is transformed by the transformation \mathbf{T} compose \mathbf{S} ? Write down your answer as a formula only using \mathbf{M}_t , \mathbf{M}_s , \mathbf{t}_t , \mathbf{t}_s , and \mathbf{p} . Do not use matrix operations in your answer.
- 7. (9 pts) Let's say you want a triangle rendered by drawTriangle() to be transformed in the following order **w.r.t. global frame**. Fill in the blank (a), (b), and (c) to do this (You have to use glTranslatef(), glScalef(), and glRotatef() functions)
 - 1) First, translated by (1,0,2)
 - 2) Then, rotated 90 degrees about y axis
 - 3) Lastly, scaled by (-2, 2, 4)

Student Number:

Name:



8. (5pts) What is the OpenGL current transformation matrix at line (a) in the following code?

```
def render():
    global M1, M2; // 4x4 transformation matrices (numpy n-dim array)
    glClear(GL_COLOR_BUFFER_BIT)
    glLoadIdentity()
    glPushMatrix()
    glMultMatrixf(M1.T)
    glPushMatrix()
    glMultMatrixf(M2.T)
    drawSomething1()
    glPopMatrix()
    drawSomething2() // (a)
...
```

9. (7 pts) The position of point **p** is (5,-10,21) w.r.t. an affine frame T_1 , whose x-axis=(-1,0,0), y-axis=(0,1,0), z-axis=(0,0,1), and origin=(1,20,-5) w.r.t. global frame. Write down the position of **p** w.r.t. global frame.

10. (6 pts) Let's say the modeling, viewing, projection, viewport transformation matrices are M_m, M_v , M_{pj}, M_{vp} , respectively. Write down an equation describing the relationship between the position of a vertex in the object space, p_0 , and the position of the corresponding vertex in the screen space, p_s using given transformation matrices.

Student Number:

Name:

11. (7 pts) What should go to the blank (a) to render the following image?



- 2) glOrtho(-.5,.5, -1,1, -1,1)
- 3) glortho(-1,1, -1,1, -1,1)
- 4) glortho(-2,2, -2,2, -1,1)
- 5) glOrtho(-1,1, -2,2, -1,1)
- 6) glOrtho(-2,2, -.5,.5, -1,1)

Student Number:

Name:

12. (12 pts) The following figure and equation show the process of getting a transformation matrix that maps a point (p_x, p_y) in a rectangular space from (x_1, y_1) to (x_h, y_h) to a point (p_x', p_y') in a rectangular space from (x_1', y_1') to (x_h', y_h') . Fill in the blank (a) – (f) to complete the equation.



13. (5 pts) Choose one false(incorrect) statement about triangle mesh representations.

- 1) Separate triangle representation stores all vertex positions for each triangle.
- 2) You can use glDrawArray() to render a triangle mesh written in separate triangle representation.
- 3) Separate triangle representation is more efficient than indexed triangle set representation in terms of memory space.
- 4) You can use glDrawElements() to render a triangle mesh written in indexed triangle set representation.

Student Number:

Name:

14. (6 pts) The following figure shows vertex indices of a cube object which is a triangle mesh. The following code creates an index array to draw this cube object with the glDrawElements () function. Fill in the blanks (a) and (b) below to complete the index array (use "counterclockwise" for the vertex winding order to indicate "front" face).



15. (6 pts) You have an affine transformation matrices **M** which is expressed as a 4 x 4 2-dim numpy ndarray object in your code. Now you want to transform a triangle with vertices (0,1,0), (0,0,0), and (1,0,0) by **M** and draw it using the following render() function, which takes **M** as a parameter. Fill in the blank (a), (b), (c) to complete render(). (You can access the numpy module using the name 'np'.)

```
def render(M):  # implementation 2
  glClear(GL_COLOR_BUFFER_BIT)
  glLoadIdentity()

  glBegin(GL_TRIANGLES)
  glVertex3fv(_(a)_)
  glVertex3fv(_(b)_)
  glVertex3fv(_(c)_)
  glEnd()
```