Student Number: **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. (5 pts) Which of the PyOpenGL function calls below is wrong?

   1) glColor3ub(255, 0, 0)

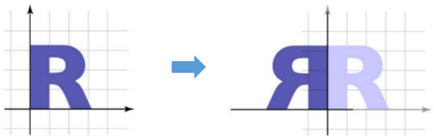
   2) glVertex2f((0.0, 1.0))

   3) glVertex2fv([-1.0,-1.0])

   4) glVertex2fv(np.array([1.0,-1.0]))

2. (5 pts) A ___________ is the portion of memory to hold the bitmapped image that is sent to the display device. Fill in the blank.

3. (6 pts) Write down the 2 x 2 matrix to perform the following transformation (reflection about the y-axis).

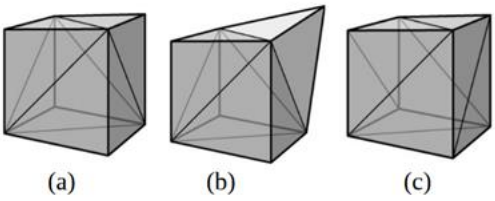


4. (6 pts) Write down the 4 x 4 transformation matrix to translate by (1, 2, 3).

5. (6 pts) The following table summarize various transformations. Fill in the blank (a), (b), (c) with correct transformation types.

__(a)__ transformation : combination of scale, rotation, reflection, and shear
__(b)__ transformation : combination of __(a)__ transformation and translation
__(c)__ transformation : combination of rotation and translation

6. (6 pts) See the following figure, and write down answers for following questions.



1) Which of these share the same topology?

2) Which of these share the same geometry?

7. (6 pts) Let's say the modeling, viewing, projection, viewport transformation matrices are $\mathbf{M}_{m}$, $\mathbf{M}_{v}$, $\mathbf{M}_{pj}$, $\mathbf{M}_{vp}$, respectively. Write down an equation describing the relationship between the position of a vertex in the object space, $\mathbf{p}_{o}$, and the position of the corresponding vertex in the screen space, $\mathbf{p}_{s}$ using given transformation matrices.

8. (6 pts) Let's say drawTriangle() draws a certain triangle. Now you want rotate the triangle about the y-axis by 30 degrees and then translate it by 2 along the x-axis w.r.t. a global coordinate system, and draw the triangle. Fill in the blank (a), (b) to do this (You have to use glTranslatef() and glRotatef() functions)

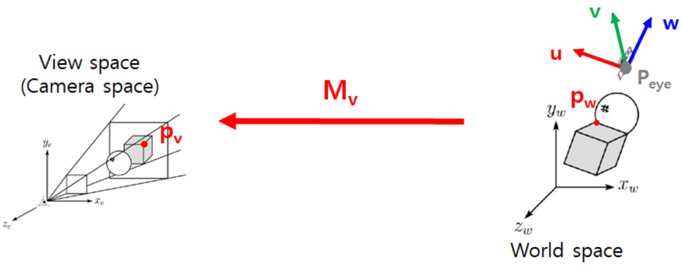
```
def render():
    glClear(GL_COLOR_BUFFER_BIT)
    glLoadIdentity()

    _____(a)_____
    _____(b)_____
    drawTriangle()
```

9. (8 pts) The order of composite affine transformations does matter. For example, the results of moving and then rotating a certain geometry, and rotating and then moving the same geometry are different. Briefly explain why this happens.

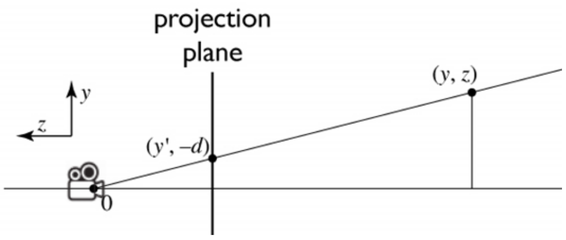
10. (8 pts) Explain what this 2D transformation matrix does. You have to describe the order of transformations and which reference frame (local or global) your explanation is based on.

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & -1 & 4 \\ 0 & 0 & 1 \end{bmatrix}$$

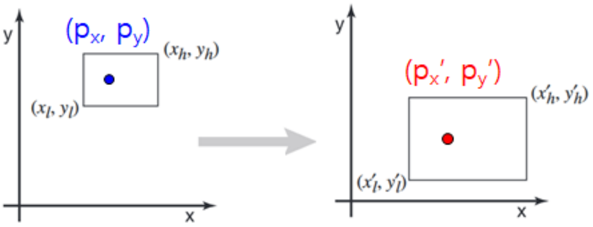
11. (8 pts) Let's say the local frame of a camera is defined by three unit vectors **u, v, w** and one point $\mathbf{P}_{\text{eye}}$ as shown in the following figure. Write down viewing transformation matrix $\mathbf{M}_{\text{v}}$ that transforms the position of a vertex in the world space, $\mathbf{p}_{\text{w}}$, to the position of the corresponding vertex in the camera space, $\mathbf{p}_{\text{v}}$ (You can write the answer in the form of an inverse matrix of a matrix).



12. (8 pts) Write down the 3 x 4 projection matrix that maps a 3D point (x, y, z) to a 2D point on the projection plane (x', y') (Hint: use similar triangles, homogeneous coordinates).



13. (8 pts) Write down the 3 x 3 windowing transformation matrix that maps a point (p_x, p_y) in a rectangular space from (x_l, y_l) to (x_h, y_h) to a point (p_x', p_y') in a rectangular space from (x_l', y_l') to (x_h', y_h') as shown in the following figure (You can write the answer in the form of a product of matrices).



14. (14 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. There are two ways of implementing render(), so fill in the blank (a), (b), (c), (d), (e), (f), (g) to complete two different implementations of render(). (You can access the numpy module using the name 'np'.)

```
def render(M):       # implementation 1
    glClear(GL_COLOR_BUFFER_BIT)
    glLoadIdentity()

    glMultMatrixf(__(a)__)
    glBegin(GL_TRIANGLES)
    glVertex3fv(__(b)__)
    glVertex3fv(__(c)__)
    glVertex3fv(__(d)__)
    glEnd()

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

    glBegin(GL_TRIANGLES)
    glVertex3fv(__(e)__)
    glVertex3fv(__(f)__)
    glVertex3fv(__(g)__)
    glEnd()
```