SAMPLE FINAL
ECS 175 - Computer Graphics

Final (Spring 2013)

Use the available space on these sheets to write your answers. Read questions in their entirety before preparing your answers!

**Explain and give details.** Write down your assumptions, calculations, reasoning, and steps. Explain your results. This allows us to give partial credit if you make arithmetic mistakes.

This exam is worth a total of 100 points.

**STUDENT NAME:**

**STUDENT ID NUMBER:**
1. **TRUE** or **FALSE**:
   Specular reflection is focused along an ideal reflection direction.

2. **TRUE** or **FALSE**:
   The fragment processor cannot modify texture coordinates.

3. **TRUE** or **FALSE**:
   Z-buffering is an object-space depth-sort technique.

4. **TRUE** or **FALSE**:
   A PC mouse corresponds to a ‘locator’-type logical device.

5. **TRUE** or **FALSE**:
   The ‘varying’ attribute in GLSL is used to pass values from the fragment shader to the vertex shader.

6. **Circle the best answer**:
   The main goal of texture filtering techniques is to:
   a) Make texture-mapping more efficient
   b) Reduce the amount of texture memory used
   c) Reduce the amount of visible artifacts during texture mapping

7. **Circle every answer that is true**:
   Normal mapping
   a) Uses normals given in object space.
   b) Provides means to increase shading details.
   c) Does not change the silhouette of an object.
   d) Is performed by the rasterizer.
8. Describe the Phong lighting model (types of reflection, vectors involved, formula). Is Phong lighting a global or a local lighting technique?
9. What type of transformation is performed by the following matrix?

\[
\begin{pmatrix}
2 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & -3 & -4 \\
0 & 0 & -1 & 0
\end{pmatrix}
\]

10. Given points \( p_1 = (1, 1, 1) \) and \( p_2 = (2, 5, 2) \), where does the corresponding line segment intersect a plane given by point \( p_0 = (0, 2, 1) \) and normal \( n = (0, 1, 0) \)?
11. Compute the area and normal for triangle \((p_1, p_2, p_3)\) with \(p_1 = (0, 1, 2),\) \(p_2 = (1, 2, 2),\) \(p_3 = (2, 0, 0).\)

12. Build the kinematic chain for the shown robot arm. Compute the complete transformation matrix for the point located at the end of the arm.
13. Describe how to perform a rotation around an arbitrary point (and axis) in 3D.

14. What transformation is applied to Object 3 by the following OpenGL code? Write down the complete transformation matrix. Explain the difference between post-multiplication and pre-multiplication of transformation matrices.

```c
glTranslated(1,0,0);
glPushMatrix();
glRotated(1,1,1,1);
glScaled(10,10,1);
Object1.draw();
glScaled(0.1,0.1,1);
glPopMatrix();
glRotated(10,0,1,0);
glPushMatrix();
glScaled(1,2,1);
Object2.draw();
Object3.draw();
```
Given the GLUT functions below, fill in your code to draw a 3D tetrahedron (see example figure - make sure the local y-axis passes through your object). Implement the following functionality: If the user presses 'w', move the tetrahedron along the positive y-axis. If the user presses 's', move it along the negative y-axis. In case the user presses 'a', rotate it around the positive y-axis, in case of 'd', rotate in around the negative y-axis.

```
int main(int argc, char** argv)
{
    glutInitWindowSize(500, 500);
    glutInit(&argc, argv);
    ...
    glutKeyboardUpFunc(keyUp);
    glutDisplayFunc(draw);
    ...
}
```
void draw()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0,0,1,0,0,0,0,1,0);

    glutSwapBuffers();
}

void keyUp(unsigned char key, int, int)
{
