NAME (print): ____________________________________

STUDENT NUMBER: ________________________________

There are 6 questions totaling 174 marks.
A calculator without a memory bank may be used. No other aids are permitted.
Use or possession of unauthorized materials will automatically result in a grade of zero for this examination.
Please answer all questions on the exam paper. Use the back of pages if necessary.
Keep written answers brief and to the point. Write neatly and legibly.
A grade of 40% is required on this exam to pass the course.

<table>
<thead>
<tr>
<th>Question</th>
<th>Total</th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>174</td>
<td></td>
</tr>
</tbody>
</table>
Question 1 [24] Multiple choice: select the best answer to each question
[+2 correct, 0 nothing, -0.5 wrong guess; one free wrong guess without penalty]

1. You give your function to GLUT to execute when certain events are seen. This is called:
   a. display lists
   b. GLUT programming
   c. immediate-mode execution
   d. callback registration

2. In ray tracing, the nearest intersection of a ray and an object is called a/an:
   a. hit point
   b. nearest intersection
   c. reflection vector
   d. HSV entity

3. A viewport is:
   a. colour palette used in index colour mode
   b. an area of graphics window to draw into
   c. an old vector-graphics monitor
   d. camera’s view of scene

4. An advantage of the Cohen-Sutherland line clipping algorithm is:
   a. minimizes clipping of lines
   b. optimizes trivial accept and rejection of lines
   c. reduces aliasing
   d. fast integer arithmetic when drawing lines

5. Which of the following is not a property of an affine transformation:
   a. parallelism is preserved
   b. proportional distances are preserved
   c. angles between vertices are preserved
   d. lines are preserved

6. A cylindrical projection would be used during:
   a. perspective projection calculations
   b. Phong lighting
   c. ray tracing intersection optimizations
   d. bitmap texture mapping
Question 1. (cont)

7. Let gval be float, cvf be a float array of size 3, and cvub be an unsigned byte array of size 3. Which of the following is correct:

a. glVertex3fv(2.5, gval, 0.0);
b. glVertex4fv(2.5, gval, 0.0, 1.0);
c. glVertex3fv(cvub);
d. glVertex3fv(cvf);

8. A 2D triangulation can be used to:

a. create a random-sampled 3D surface
b. verify the correctness of a 2D convex hull
c. implement Gouraud shading
d. create traumatic assignment questions

9. In ray tracing, refraction refers to:

a. perfect mirror reflection off of surfaces
b. light bending through semi-transparent solids
c. computing shadows from lights
d. optimizing intersections by organizing the scene into data structures

10. Which of the following is a divide-and-conquer algorithm:

a. Area subdivision
b. floodfill
c. Phong shading
d. ray tracing

11. Orthographic projection is useful for:

a. modeling and scene building
b. creating perspective foreshortening
c. inverting the universe through a unit circle
d. giving the illusion of smooth surfaces

12. Aliasing in graphics is caused by:

a. a sampling error caused by representing a continuous quantity with discrete signals
b. integer arithmetic
c. polygonal models
d. non-photorealistic rendering
Question 2 [20] Define and briefly discuss the following terms or OpenGL /GLUT commands:

a) minimum-weight triangulation

b) frustum

c) glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);

d) colour map mode

e) key frame
Question 3 [20]

(a) [10] Using general homogeneous 3D transformation matrices, show that the application of a scale and a translate transformation is or is not associative (order-dependent).
Question 3 (cont)

(b) [10] Explain how OpenGL’s stack of transformation matrices is used to invert transformations. Include a diagram with your discussion. Identify and discuss the major OpenGL commands that implement this facility. Also discuss the advantage it has over using sequences of inverted transformations.
Question 4 [25]

(a) [15] Identify the 3 types of lighting effects found in OpenGL’s basic lighting model. Discuss how each effect contributes to the overall lighting of an object. Include equations for the monochrome versions of these effects, and discuss the equations and their parameters.
4 (b) [10] Explain how Gouraud shading works. Discuss all the steps for implementing Gouraud shading. Mention where the lighting model in 5(a) comes into consideration. Identify a weakness of Gouraud shading, and why it arises.
Question 5 [20]  Select one of the following:

(a) Give pseudo-code for the Quickhull algorithm. Discuss what the algorithm steps are doing. Include a diagram to help your discussion. (Note: you do not need to include the algebraic formula used; however, do describe the general calculations required at various steps).

(b) Discuss the idea of back face culling. Include a diagram to help your discussion. What is its main advantage? What overhead is involved with it, if any? How do you use it with OpenGL?
**Question 6 [65]**

Write C/ OpenGL/GLUT code that does the following. Read the entire question before beginning. Some parts rely on the use of others. You can assume that GLUT callbacks may invoke your functions. Please do not implement anything that is not requested (ie. user commands, lighting, textures, menus, etc.). Use the backs of pages if necessary.

This question involves implementing a cube model for an animation. The cube can be in one of two states – intact, or exploding.

(a) [10] Define some suitable data structures that define an intact cube, centered on the origin (0,0,0). Each side of the cube should have a length of 2. Each vertex has a random RGB colour. Include code that initializes the cube.
(b) [15] Write a loop in C that renders the intact cube defined in (a).
(c) [15] Next, you are to define an exploding cube that works as follows. In the main application, the cube can undergo an explosion before it is removed from the scene. Each of the 6 sides of the cube will move away from the origin, along one of the positive or negative X, Y, or Z directions, according to the side’s orientation around the origin. In other words, the cube explodes so that each side flies straight out, away from the cube center. Each side will also spin randomly as it flies away.

Define some new data structures (if different from (a)) that will let you implement this explosion effect (perhaps the main program replaces the intact cube in (a) with one that uses these new data structures). Also include some code that initializes the data structures so that the explosion effect can proceed in (d) below.
(d) [25] Now write a callback that will update and draw the cube explosion as a single updated frame of an ongoing animation. The callback should update the cube as necessary, draw it, and release control back to GLUT.

```cpp
    glEnd();
```