Exam Instructions (Read Carefully)

1. Sign the first page of the exam with your **Signature** in the space provided on the upper left immediately.

2. Continue reading the instructions, but do **not open the exam booklet** until you are told to do so by a proctor.

3. Print your **Name** and **Student Identification Number** on every page in the space provided at the top of each page before you start the exam.

4. Cheating is an academic offense. Your signature on the exam indicates that you understand and agree to the University’s policies regarding cheating on exams.

5. Please read the entire exam before answering any of the questions.

6. There are **four** questions on this exam, each worth the indicated number of marks. **Answer as many questions as you can.**

7. Write all of your answers on these pages. If you need more space, there is blank space at the end of the exam. Be sure to indicate when a question is continued, **both** on the page for that question and on the continuation page.

8. Interpret the exam questions as written. **No questions** will be answered by the proctor(s) during the exam period.

9. The exam is **closed book**. There are **no aids permitted**, except for a calculator.

10. You have **70 minutes** in which to work. **Budget your time wisely.**

11. In the event of a **fire alarm** during the exam, enter the four-character code provided by the proctor(s) in the space on the upper right, then gather your belongings and exit the room, handing your exam to a proctor as you exit.

12. No one will be permitted to leave the exam room during the **last ten minutes** of the exam.

<table>
<thead>
<tr>
<th>Question</th>
<th>Mean</th>
<th>SD</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.5</td>
<td>3.7</td>
<td>12</td>
</tr>
<tr>
<td>2(a-b)</td>
<td>7.0</td>
<td>1.9</td>
<td>8</td>
</tr>
<tr>
<td>2(c-e)</td>
<td>9.7</td>
<td>3.1</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>16.7</td>
<td>8.1</td>
<td>28</td>
</tr>
<tr>
<td>4(a-j)</td>
<td>16.6</td>
<td>3.6</td>
<td>20</td>
</tr>
<tr>
<td>4(k-t)</td>
<td>16.2</td>
<td>3.6</td>
<td>20</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>74.7</strong></td>
<td><strong>18.7</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>
Question #1 [12 marks total]
This question tests your knowledge of how images are displayed on a CRT in a manner that is perceived by humans to be “real”. All answers should be given to two decimal places, and with appropriate units (seconds, milliseconds, microseconds, or nanoseconds). You may use a calculator. **Show your calculations** if you want to receive partial credit for incorrect answers.

A very high resolution graphics display has a frame buffer that is 1600 pixels wide and 1280 pixels high. It has a non-interlaced 72Hz refresh. Assume that 8% of the scan line time is required for horizontal retrace and that 7% of the frame time is required for vertical retrace.

(a)  **[3 marks]** What is the total time required for a single frame, including all of the horizontal and vertical retrace times?

\[14 \text{ milliseconds} = \frac{1}{72}\]

(b)  **[3 marks]** What is the time required for vertical retrace?

\[0.97 \text{ milliseconds} = 0.07 \times (a)\]

(c)  **[3 marks]** What is the time required for a single scan line, including one horizontal retrace?

\[10 \text{ microseconds} = (1 - 0.07) \times (a) / 1280\]

(d)  **[3 marks]** What is time required for a single pixel?

\[5.8 \text{ nanoseconds} = (1 - 0.08) \times (c) / 1600\]
Question #2 [20 marks total]
This question tests your knowledge of **virtual input devices** or **input primitives** as they are called in the textbook.

Name each of the five virtual input devices (2 marks), give a brief description of the data that is provided to an application program by the input primitive (1 mark), and also identify one physical device that naturally implements the input primitive (1 mark).

(a) [4 marks] First virtual input device:

**STRING** returns a string of characters to the application program (these could be in ASCII, or some other character set), usually with a special termination character at the end, or possibly with a character count.

The physical keyboard is the physical device most naturally associated with the virtual keyboard, as the name suggests.

(b) [4 marks] Second virtual input device:

**CHOICE** returns a selection from a fixed number of items, such as from a menu of commands or options.

A physical button is the natural implementation for the choice virtual device.

*(Continue with three more virtual input devices on the next page.)*
(c) [4 marks] Third virtual input device:

VALUATOR is a scalar or possibly a vector quantity returned to the application program (such as size, saturation, velocity, or force). These quantities can be either continuous or discrete, and either bounded or unbounded.

A physical slider or dial is a natural physical device for this virtual device.

(d) [4 marks] Fourth virtual input device:

LOCATOR returns an \((x,y)\) coordinate pair to the application program, sometimes in world coordinates but often in screen, viewport, or normalized device coordinates. Some devices provide \((x,y,z)\) and a few 6DOF (degree of freedom) devices provide another three coordinates of orientation as well as position. A "trigger" may also be returned, indicating the type of event that caused the locator to return position information.

The mouse, tablet, and trackball are all typical physical devices used to implement the virtual locator device in a natural way.

(e) [4 marks] Fifth virtual input device:

PICK returns the identity the portion or element within the display list that has been selected by the user. In a hierarchical display list, the identity may include a number of levels within the display list each of which includes the particular element(s) that were selected by the user.

The lightpen in a calligraphic display is the only physical device that naturally implements the pick virtual input device.
Question #3 [28 marks total]
This question tests your knowledge of the OpenGL and GLUT components that are required for the first assignment.

Each part of this question will refer to the following C function, which is the display callback function that will be invoked by GLUT whenever the screen is to be updated. You may assume that GLUT has been appropriately initialized and that when preceded by exactly the calls given below, the function `DrawTextbook()` correctly renders a shaded 3-D model of the textbook located at the origin so that it appears in the middle of the screen, with the front cover in normal orientation (top at the top, bottom at the bottom, facing toward the viewer), and with the spine of the book on the left side.

```c
MyDisplay(void)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.,4./3.,0.5,1.5);  ### Line 1
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0.,0.,1.,0.,0.,0.,0.,1.,0.);  ### Line 2
    glRotatef(0.,0.,1.,0.);  ### Line 3
    glRotatef(0.,1.,0.,0.);  ### Line 4
    DrawTextbook();
}
```

The various parts of this question assume that a slightly modified version of the code above is used to render a 3D model of the textbook. For example, consider the following change to the code that rotates the 3D model 180 degrees about the z-axis, resulting in a final image with the cover of the book upside down, but still centered in the middle of the screen with none of the other sides of the book visible.

```c
    glRotatef(180.,0.,0.,1.);  ### Line 3 changed
```

You may find the following OpenGL and GLUT function prototypes useful when you answer the seven parts of this question on the next page.

```c
    glLoadIdentity(void);
    glMatrixMode(GLenum mode);
    glRotatef(GLfloat degrees,
        GLfloat x, GLfloat y, GLfloat z);
    glViewport(GLint x, GLint y, GLsizei width, GLsizei height);
    gluLookAt(GLdouble eyex, GLdouble eyey, GLdouble eyez,
        GLdouble centerx, GLdouble centery, GLdouble centerz,
        GLdouble upx, GLdouble upy, GLdouble upz);
    gluOrthod2D(GLdouble left, GLdouble right,
        GLdouble bottom, GLdouble top);
    gluPerspective(GLdouble fovy, GLdouble aspect,
        GLdouble zNear, GLdouble zFar);
    glutInitWindowSize(GLsizei width, GLsizei height);
    glutWindowPosition(GLint left, GLint top);
```
In each part below, describe what happens when the indicated line(s) is (are) changed as indicated, but all other lines remain as they originally were (i.e., only the indicated lines are changed from the original version of the function DrawTextbook() given on the previous page). Indicate your answer by choosing the best phrase from the following list and entering its number before the question.

1. The front cover of the book is seen on the screen upside-up with the spine on the left side.
2. The front cover of the book is seen on the screen upside-down with the spine on the right side.
3. The front cover of the book is seen on the screen tilted so that the spine is at the top of the screen.
4. The front cover of the book is seen on the screen tilted so that the spine is at the bottom of the screen.
5. The front cover of the book is seen on the screen right-side-up with the spine on the left side, but smaller than before.
6. The spine of the book is seen on the screen with the front cover facing to the left.
7. The spine of the book is seen on the screen with the front cover facing to the right.
8. The spine of the book is seen on the screen with the front cover facing up.
9. The spine of the book is seen on the screen with the front cover facing down.
10. The front and back cover of the book are seen jumbled together because of a failure to properly resolve which are the visible surfaces.
11. Nothing except a simple outline of the book is seen, with neither the front or back cover rendered as a shaded model.
12. NONE OF THE ABOVE.

[4 marks each]

_10__ (a)  gluPerspective(45.,4./3.,0.0001,1000000.); ### Line 1 changed
_1   NOTE: If the graphics board uses floating-point for its z-buffer, the image may be correct (i.e., 1 would be an acceptable answer).
_11__ (b)  gluPerspective(45.,4./3.,1.0,1.0); ### Line 1 changed
_1   NOTE: Some implementations of OpenGL detect this as an error.
_12__ They may provide a default perspective transformation (i.e., 1 or 12 would be an acceptable answer).
_15__ (c)  gluPerspective(60.,4./3.,0.5,1.5); ### Line 1 changed
_4   (d)  gluLookAt(0.,0.,1.,0.,0.,0.,0.,1.,0.,0.); ### Line 2 changed
_1   (e)  gluLookAt(0.,0.,1.,0.,0.,0.,0.,2.,0.); ### Line 2 changed
_7   (f)  glRotatef(90.,0.,1.,0.); ### Line 3 changed
_12__ (g)  glRotatef(90.,0.,1.,0.); ### Line 3 changed
_11__ (h)  glRotatef(90.,0.,0.,1.); ### Line 4 changed

NOTE: The original question had a typographic error, with two answers having the same number (i.e., 11 or 12 would be an acceptable answer).
Question #4 [40 marks – 2 marks each]
This question tests your general knowledge of the concepts and terminology introduced in the course.

The following terms or people’s names are possible answers for the questions on subsequent pages. Use the number corresponding to a term or name below as an in the space provided answer if you think it is the best match for one of the concepts or terms on subsequent pages. Each term may be used once, more than once, or not at all.

(1) antialiasing
(2) alpha
(3) beta
(4) bitmap
(5) calligraphic
(6) culling
(7) delta
(8) doping
(9) flicker
(10) flutter
(11) GLUT
(12) graphics pipeline
(13) interlace
(14) interpolation
(15) jaggies
(16) pixel
(17) refresh rate
(18) screen window
(19) Sketchpad
(20) update rate
(21) viewport
(22) virtual camera
(23) virtual memory
(24) virtual trackball
(25) window
(26) Windows/NT
(27) wireframe
(28) WYSIWYG
(29) Xenon
(30) x-ray
(31) z-axis modulation
(32) z-buffer
For each definition or phrase below and on subsequent pages, write the number of the term listed on the previous page that best matches the definition or phrase.

[2 marks each]

_20_ (a) The number of times per second (usually 30-72 Hz, but a minimum of 10-15 Hz) that the image being displayed on a screen must be modified to produce the illusion of continuous, smooth motion.

_17_ (b) The number of times per second (usually 60-72 Hz, but a minimum of 40 Hz – the “critical fusion frequency”) that an image must be displayed on a screen in order to maintain the illusion of a continuous image (“persistence of vision”).

_9_ (c) Below the critical fusion frequency images on a CRT suffer from this problem.

_13_ (d) A standard television set employs this technique to effectively double the rate at which images seem to be displayed on the screen, thus providing a better illusion of a continuous image.

_8_ (e) Normally the chemical compounds used as CRT phosphors are selected for their colour and other visible attributes, but when a lightpen is used with the CRT this is sometimes added to the phosphor so the circuitry in the lightpen can better detect when it sees light produced by the CRT.

_19_ (f) The “father of computer graphics” made many early contributions to the field, starting with this doctoral dissertation from MIT in 1963.

_15_ (g) The discrete, grid-like structure of pixels gives rise to this visual artifact when straight lines are drawn on a raster display if the lines are not exactly vertical or horizontal.

_1_ (h) These techniques, such as using gray-scale for pixels that partially overlap a line or character, are used to overcome artifacts caused by the discrete, grid-like structure of pixels.

_2_ (i) In addition to the usual RGB channels, some frame buffers have this additional channel that can be used to represent opacity (transparency) or area coverage.

_32_ (j) A 16-bit, 32-bit, or 64-bit field associated with each pixel in a frame buffer that can be used to determine the visible surface at each pixel.
[2 marks each]

__4__ (k)  The frame buffer or pixel array associated with a “bi-level” raster display.

__27__ (l)  This style of rendering draws only the edges of objects.

__5__ (m)  These early computer graphics displays were limited to line drawing, which is sometimes called “random scan” or “stroke” graphics.

__22__ (n)  A useful analogy or metaphor that assists in understanding the various viewing and perspective parameters that are required for functions such as gluLookAt() and gluPerspective().

__24__ (o)  A “widget” that is implemented with a mouse or other pointing device to control two or more degrees of rotational freedom in a fairly natural manner.

__12__ (p)  The entire process of creating and perceiving an image on a CRT or other display device starting with a software application program and ending with the human visual system and brain.

__25__ (q)  The subset of the World Coordinate System that is eventually rendered as an image on the face of the CRT.

__21__ (r)  The portion of the virtual frame buffer provided by OpenGL into which a graphics program renders pixels for an image.

__18__ (s)  The portion of the face of the CRT that displays an image stored in the physical frame buffer.

__11__ (t)  This provides a set of “widgets” and auxiliary functions for quickly building user interfaces using lower-level functions in OpenGL and X Windows or some other window system.