NAME:

Note: This exam is open-notes, open-book, open-computer. However, you may NOT connect to the wireless network or use the internet.

The exam consists of 5 questions. Each question is worth 20 points. Please answer all the questions in the space provided, overflowing on to the back of the page if necessary.

This exam has been designed to take 1 hr. However, you have 3 hours to complete the exam.

Computer graphics relies heavily on geometric transformations. Most common are transformations such as rotations, translations, and scales. In the following, $T(dx,dy)$ refers to a translation by $(dx, dy)$, $R(a)$ refers to a rotation by $a$ degrees, and $S(sx, sy)$ refers to a scaling by $(sx, sy)$. For simplicity, assume all transformations are 2D.

The order of transformations may matter. Also, sometimes the order may be rearranged, but the arguments will change. State whether the following statements are True or False.

$T(1,0) S(2,2) = T(2,0) S(1,1)$?

$T(-1,0) T(0,2) = T(0,1) T(-1,1)$?

$R(180) = R(-180)$?

$R(45) S(2,1) = S(2,1) R(45)$?

$T(1,0) R(90) = T(0,1)$ ?

$R(-90) T(1,0) R(90) = T(0,-1)$ ?

Transformations have inverses. Applying a transformation followed by its inverse has no effect. State whether the following formulas involving inverses are True or False.

$R(180)^{-1} = R(180)$?

$T(1,1)^{-1} = T(-1,-1)$?

$[R(45) T(1,0)]^{-1} = T(1,0) R(45)$?

$[R(45) S(2,2)]^{-1} = R(-45) S(.5,.5)$?
2. [20 points] Windows

Suppose we have a display that has size (ScreenWidth, ScreenHeight). The screen is shown below with the thickest border. On the screen is a window. The window has size (w,h) and its upper left corner is at (x,y). On the left and right the window is in different positions and has a different size.

We want to draw a clock face and have it appear to stick to the screen. That is, as the window is moved and resized, the clock sticks to the display and does not move with the window.

Assume the window coordinates would be set to (xmin,xmax,ymin,ymax) if the window occupied the full screen (ScreenWidth, ScreenHeight).

For this problem, GLUT has been modified to call back the function reposition when the window is changed. Describe the sequence of OpenGL calls with the proper arguments so that when the clock is drawn, it appears in the correct position. Make sure to include glViewport.

```c
void reposition( int x, int y, int w, int h )
{
}
```
3. [20 points] Geometry

Points and vectors are not the same. In particular, points and vectors do not transform in the same way.

3A [5 points]. We are given a point $P=(px,py,pz)$ and a vector $V=(vx,vy,vz)$. We now translate by $(tx,ty,tz)$. What is the new position of the point? What are the new coordinates of the vector?

3B [5 points]. Suppose we rotate the coordinate system by 90 degrees about z. What is the new position of the point? And the vector? (In this question, ignore the translation in 3A).
3C [5 points]. We wrote a program that computes a new point from two points using the expression \( p = a \cdot p_1 + b \cdot p_2 \), where \( p, p_1 \) and \( p_2 \) are points, and \( a \) and \( b \) are floats. Suppose the points \( p_1 \) and \( p_2 \) are translated by \((tx, ty, tz)\). That is, \( p_1' = T(tx, ty, tz) \cdot p_1 \) and \( p_2' = T(tx, ty, tz) \cdot p_2 \). We would expect the point \( p' = T(tx, ty, tz) \cdot p \). Prove whether this is true or false.

3D [5 points]. Suppose we compute \( p = (1-a) \cdot p_1 + a \cdot p_2 \). Now we translate \( p_1 \) and \( p_2 \) as we did in 3C. Again, we would expect \( p' = T(tx, ty, tz) \cdot p \). Is this true or false?
4. [20 points] Bezier Curve

The most widely used curve in computer graphics is the Bezier curve. The quadratic Bezier curve is given by 3 points, P0, P1, and P2. The cubic Bezier curve is given by 4 points, P0, P1, P2, and P3.

To draw another curve, we need to convert it to graphics primitives. Normally we draw a curve by drawing a set of line segments. However, it is possible to convert a curve to a Bezier curve. The advantage of this approach is that it takes many fewer Bezier curves than line segments to closely approximate our curve.

In this problem, we want to draw the parabola $y = x^2$.

We want to draw a section of the parabola from (0,0) to (1,1).

4A [10 points] Determine the positions of the 3 controls points P0, P1, and P2 of the quadratic Bezier curve. Position these points so that the Bezier curve goes through (0,0) and (1,1) and is tangent to the parabola at these two points.
4B [10 points] Calculate the positions of the 4 controls points P0, P1, P2 and P3 of the cubic Bezier curve. Position these points so that the Bezier curve goes through (0,0) and (1,1) and is tangent to the parabola at these points. Also require that the point (1/2, 1/4) on the parabola be on the Bezier curve. That is, P(1/2) = (1/2, 1/4).
5 [20 points] Input

5A [10 points]. When designing a graphics application, it is often necessary to select different objects drawn on the screen. In order to implement selection, you need a hit-testing procedure that determines whether the mouse position \( x, y \) is over the object.

In the MicroUI assignment, you created UI widgets at different positions on the screen. In this problem, we want to add the capability to transform the UI widgets to any location on the screen using a sequence of OpenGL transformations. For example,

// make a rectangular widget of the given width w and height h
UIWidget widget(w, h);

// transform and draw
glLoadIdentity();
glTranslatef(x1,y1,0);
glRotate(a1, 0,0,1);
glPushMatrix();
  glTranslatef(x2,y2,0);
  glRotate(a2, 0,0,1)
  widget.Draw();
glPopMatrix();

Describe in detail how would you detect whether a mouse location \( x, y \) is inside the widget? Provide all the mathematical details.
5B [10 points]. A very common method for converting rotary motion to a direction is to use quadrature encoding. Trackballs and mechanical mice use quadrature encoding.

Quadrature encoders provide two signals A and B. The values of A and B are either 0 or 1. As rotation occurs, the values of A and B change as shown in the diagram. The waveforms are different depending on whether the motion is clockwise (CW) or counter-clockwise (CCW). In the figure below, we are rotating CW and A is on the top and B is on the bottom. If we were to rotate CCW, the waveform would be reversed and the signals A and B would run backward.

Write in pseudo-code a procedure that keeps track of A and B and returns whether the motion is CW or CCW.