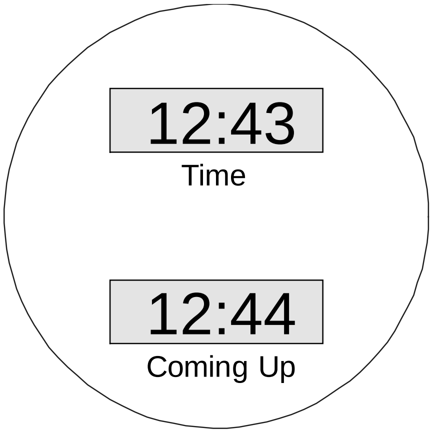
CS 100: Section Assignment 6

(For the week of March 8)

Section assignments are discussed in section and are not submitted for grading. They relate to recent lecture topics and usually to the current Programming Assignment. Prelim questions are based on Section Assignments, Programming Assignments, and Lecture examples.

1. Write a drawDigitalClock(Graphics g, int hc, int vc, int r, int hour, int minute) that displays the following designer watch face:



The rectangles should have width r and height r/3. Their centers should be displaced .45r from the center of the watch face. For the time displays use 36pt bold TimesRoman font. For the labels "Time" and "Coming Up" use 18-point Helvetica. Color the watch face magenta and the time windows pink. For purposes of this assignment, you can "guess" where the 4 drawString messages need to be located. However, make your guesses "educated" functions of r, hc, and vc. Write a paint method that draws the watch when the time is 12:59. Choose your own values for r, hc, and vc.

2. Write a graphics method drawCheckerRow(Graphics g, int hL, int vT, int s, int n, boolean first) that draws a row of an n-by-n checkerboard. The upper left corner of the row should be situated at (hL,vT). The tiles should be s-by-s and alternate in color between cyan and magenta. The first tile should be cyan if first is true and magenta if first is false.

3. Suppose s is an initialized String object with one of the four values "Autumn", 'Winter", "Spring", or "Summer". Write a fragment that assigns "September 22", "December 21", "March 22" , or "June 22"" accordingly to a string date. The fragment should then print an appropriate season announcement message, e.g., "Autumn starts on September 22".

4. A ray leaving the origin (0,0) makes an angle of *x* degrees with the positive *x*-axis. Assume that x is an int variable that contains this value and that 0 <= *x* <=360. Write a fragment that prints on one line the quadrant(s) into which the ray points. If x = 0, 90, 180, or 360 then the ray points into two quadrants. Sample output:

x	**Output**
300	Fourth
90	First and Second
0	First and Fourth
200	Third

If the ray points into more than one quadrant, then the order of the output doesn't matter, e.g., "First and Second" is ok and so is "Second and First".

```java
public class ShowClock extends Frame
{
    // Graphics g draws a clock face showing the time specified by hour and minute.
    // The center is at (hc,vc) and the radius of the clock face is r.
    public static void drawClock(Graphics g,int hc, int vc, int r, int hour, int min)
    {
        drawFace(g,hc,vc,r);
        int rHour = (int) (.6*r);                // Length of hour hand.
        int angleHour = 90 - 30*hour - min/2;    // Angle of hour hand.
        drawRay(g,hc,vc,rHour,angleHour);
        int rMin  = (int) (.85*r);               // Length of minute hand.
        int angleMin  = 90 - 6*min;              // Angle of minute hand.
        drawRay(g,hc,vc,rMin,angleMin);
    }

    // Graphics g draws a length r line segment from (hc,vc). The line segment
    // makes a counter clockwise angle of theta degrees with the positive h-axis.
    public static void drawRay(Graphics g, int hc, int vc, int r, int theta)
    {
        g.setColor(Color.black);
        int h1 = (int)(hc + r*Math.cos(theta*Math.PI/180));
        int v1 = (int)(vc - r*Math.sin(theta*Math.PI/180));
        g.drawLine(hc,vc,h1,v1);
    }

    // Graphics g draws a filled disk of radius r and center (hc,vc).
    public static void fillCircle(Graphics g,int hc,int vc, int r)
    {  g.fillOval(hc-r,vc-r,2*r,2*r); }

    // Graphics g draws a designer clock face of radius r and center (hc,vc).
    public static void drawFace(Graphics g, int hc, int vc, int r)
    {
        // Color the light gray face and black rim.
        g.setColor(Color.black);
        fillCircle(g,hc,vc,r);
        g.setColor(Color.lightGray);
        fillCircle(g,hc,vc,(int)(.9*r));

        // Draw 60 equally spaced markers around the rim.
        int hTick,vTick;               // (hTick,vTick) = center of kth marker
        double tau;                    // angle of kth marker.
        int rTick = (int)(.03*r);      // radius of the 5-minute markers
        for(int k=0;k<=59;k++){
           tau = 6*k*Math.PI/180;
           hTick = (int)(hc + .95*r*Math.cos(tau));
           vTick = (int)(vc - .95*r*Math.sin(tau));
           if(k%5==0)
           {
              // Draw a 5-minute marker.
              g.setColor(Color.red);
              fillCircle(g,hTick,vTick,rTick);
           }
           else
           {
              // Draw a non 5-minute marker.
              g.setColor(Color.blue);
              fillCircle(g,hTick,vTick,rTick/2);
           }
        }
    }
```

```
      public void paint(Graphics g)
      {
         // Draw a clock
         int hour = 12;
         int min  = 43;

         drawClock(g,400,300,200,hour,min);


         // "Pretty print" the exact time in digital form.
         String s;
         if (min<10)
            s =  hour + ":0" + min;
         else
            s =  hour + ":" + min;

         g.setFont(new Font("TimesRoman",Font.PLAIN,36));
         g.drawString(s,360,550);


      }

}

public class L11
{
      public static void main(String args[])
      {
            ShowClock d = new ShowClock();
            d.resize(800,600);
            d.move(0,75);
            d.setTitle("Time");
            d.show();
            d.toFront();
      }
}
```