INF2140: Exercises Week 1

Exercise 1: Write an applet that shows a pair of dice. When the user clicks on the applet, the dice should be rolled (i.e, the dice are assigned newly random values). Each of the dice should be drawn as a square showing from 1 to 6 dots. Write a subroutine, "`void drawDie(Graphics g, int val, int x, int y)`", to draw a die at the specified `(x,y)` coordinates. The second parameter, `val`, specifes the value that is showing on the die. Assume that the size of the applet is 100 by 100 pixels.

Hint: see the proposed solution in the file ClickableDice.java

Exercise 2: Improve your dice applet from the previous exercise so that it also responds to keyboard input. When the applet has the input focus, it should be highlighted with a colored border, and the dice should be rolled whenever the user presses a key on the keyboard. A part of that, the dice still rolls when the user clicks the mouse on the applet.

Exercise 3: Instead of using keyboard and mouse to roll the dices as in Exercise 1, using a canvas to store the dice and place a "Roll" button below the canvas to roll the dice each time the user clicks on it. Hints: using the classes of Layouts and Components and their methods

Exercise 4: Modify the Checkboard.java given below to write a checkerboard applet where the user can select a square by clicking on it. Highlight the selected square by drawing a colored border around it. When the applet is first created, no square is selected. When the user clicks on a square that is not currently selected, it becomes selected. If the user clicks the square that is selected, it becomes unselected. Assume that the size of the applet is 160 by 160 pixels, so that each square on the checkerboard is 20 by 20 pixels.

/* Checkboard.java */

```
import java.awt.*;
    import java.applet.*;

    public class Checkerboard extends Applet {

       /*  This applet draws a red-and-black checkerboard.
           It is assumed that the size of the applet is 160
           by 160 pixels.
       */

       public void paint(Graphics g) {

          int row;   // Row number, from 0 to 7
          int col;   // Column number, from 0 to 7
          int x,y;   // Top-left corner of square

          for ( row = 0;  row < 8;  row++ ) {

             for ( col = 0;  col < 8;  col++) {
                x = col * 20;
                y = row * 20;
                if ( (row % 2) == (col % 2) )
                   g.setColor(Color.red);
                else
                   g.setColor(Color.black);
                g.fillRect(x, y, 20, 20);
             }

          } // end for row

       }  // end paint()

    }  // end class
```