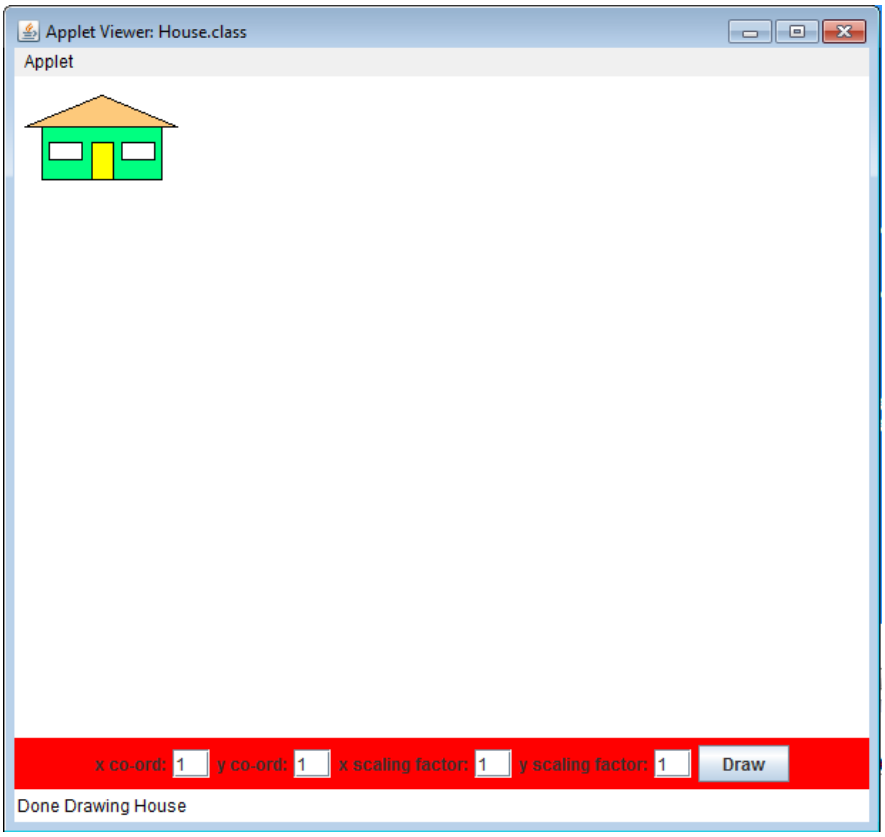
The Little House

This program has a method to draw a house based on the user's specifications.

- Adapt the drawHouse method so that it draws something else.
- The something else should have polygons (or rectangles or ovals) and your own colours.
- When finished, your drawing should move around the screen and get bigger or smaller as the current house does.




```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * Draws a house on the screen, allows the user to move and scale it
 */
public class House extends Applet implements ActionListener
{

        //Widgets, Panels
        JButton draw;
        JLabel lx, ly, lxscale, lyscale;
        JTextField tx, ty, txscale, tyscale;
        Panel flow;
        //House Coordinates, initialized to 1. (Top Right and No Scaling)
        int x = 1;
        int y = 1;
        int xz = 1;
        int yz = 1;
```

```
public void init ()
{ //Set Up Input Fields for House Coordinates
        lx = new JLabel ("x co-ord:");
        ly = new JLabel ("y co-ord:");
        lxscale = new JLabel ("x scaling factor:");
        lyscale = new JLabel ("y scaling factor:");
        tx = new JTextField ("1", 2);
        ty = new JTextField ("1", 2);
        txscale = new JTextField ("1", 2);
        tyscale = new JTextField ("1", 2);

        //Set up draw button and actionlistener
        draw = new JButton ("Draw");
        draw.addActionListener (this);
        draw.setActionCommand ("draw");

        //Set up layout, add widgets
        setLayout (new BorderLayout ());
        flow = new Panel (new FlowLayout ());
        flow.add (lx);
        flow.add (tx);
        flow.add (ly);
        flow.add (ty);
        flow.add (lxscale);
        flow.add (txscale);
        flow.add (lyscale);
        flow.add (tyscale);
        flow.add (draw);
        flow.setBackground (Color.red);
        add (flow, "South");
        resize (600, 500);
}


public void paint (Graphics g)
{ //draw a white box over the background to erase everything
        g.setColor (Color.white);
        g.fillRect (0, 0, 600, 500);
        //draw the house passing in the global variables
        littlehouse (x, y, xz, yz);
        //used only to force an update
        showStatus ("Done Drawing House");
}

/** Draws a little house
 * @param x is the top x co-ordinate of the house
 * @param y is the left most y co-ordinate of the house
 * @param xz is the x scaling factor, stretches horizontally
 * @param yz is the y scaling factor, stretches vertically
 */
public void littlehouse (int x, int y, int xz, int yz)
{
        Color lightgreen = new Color (0, 255, 128);
        Color lightorange = new Color (253, 201, 123);
        Graphics g = getGraphics ();

        x = x / xz; //reset top based on scaling factor
        y = y / yz; //reset right coordinate based on scaling factor

        //green box
        g.setColor (lightgreen);
        g.fillRect ((x + 18) * xz, (y + 34) * yz, 84 * xz, 37 * yz);
        g.setColor (Color.black);
        g.drawRect ((x + 18) * xz, (y + 34) * yz, 84 * xz, 37 * yz);
```

```
                //windows
                g.setColor (Color.white);
                g.fillRect ((x + 23) * xz, (y + 45) * yz, 23 * xz, 12 * yz);
                g.fillRect ((x + 74) * xz, (y + 45) * yz, 23 * xz, 12 * yz);
                g.setColor (Color.black);
                g.drawRect ((x + 23) * xz, (y + 45) * yz, 23 * xz, 12 * yz);
                g.drawRect ((x + 74) * xz, (y + 45) * yz, 23 * xz, 12 * yz);
                //door
                g.setColor (Color.yellow);
                g.fillRect ((x + 53) * xz, (y + 45) * yz, 15 * xz, 26 * yz);
                g.setColor (Color.black);
                g.drawRect ((x + 53) * xz, (y + 45) * yz, 15 * xz, 26 * yz);
                //roof
                g.setColor (lightorange);
                int X [] = { (x + 6) * xz, (x + 113) * xz, (x + 60) * xz, (x + 6) * xz};
                int Y [] = { (y + 34) * yz, (y + 34) * yz, (y + 12) * yz, (y + 34) * yz};
                g.fillPolygon (X, Y, 4);
                g.setColor (Color.black);
                g.drawPolygon (X, Y, 4);
        }


        /**
Draw a new little house or clears the textfields
@param e The ActionEvent invoked. e.getActionCommand contains nothing or draw
         */
        public void actionPerformed (ActionEvent e)
        {
                //get new co-ordinates and draw the house
                if (e.getActionCommand ().equals ("draw"))
                {
                        showStatus ("");
                        try
                        {
                                x = Integer.parseInt (tx.getText ());
                                y = Integer.parseInt (ty.getText ());
                                xz = Integer.parseInt (txscale.getText ());
                                if (xz <= 0)
                                        xz = 1; //divide by zero, negative error check
                                yz = Integer.parseInt (tyscale.getText ());
                                if (yz <= 0)
                                        yz = 1; //divide by zero, negative error check
                                repaint ();

                        }
                        catch (java.lang.NumberFormatException error)
                        { //if the user entered text, display an error message and clear the textfield
                                showStatus ("Enter NUMBERS in the text fields. eg. 2. No decimals.");
                                tx.setText ("1");
                                ty.setText ("1");
                                txscale.setText ("1");
                                tyscale.setText ("1");
                        }

                }

        }
}
```