

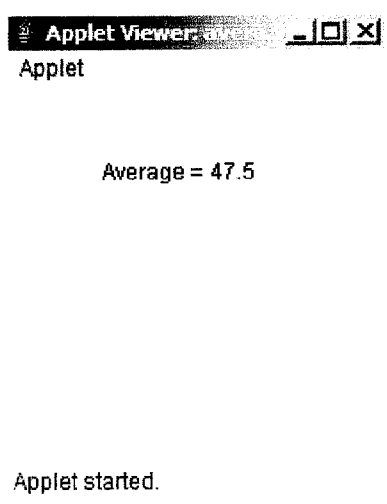
```
/*
 * Chris Tjon, Ex. 4.3, Wed 4pm
 *
 * Calculate the average of two values to 1 decimal place
 */
package average;

import java.applet.Applet;
import java.awt.Graphics;

public class Average extends Applet
{
    public void paint(Graphics g)
    {
        int mark1 = 44;
        int mark2 = 51;
        float average;

        average = (float)(mark1 + mark2) / 2; // cast the addition to get float result
        g.drawString("Average = " + average, 50, 50);
    }
}
```

5



```

/*
 * Chris Tjon, Ex 4.7 Wed 4pm
 *
 * Program to translate 2549 seconds into hours, minutes, and seconds
 */
package time_conv;

import java.applet.Applet;
import java.awt.Graphics;

public class TimeConv extends Applet
{
    public void paint(Graphics g)
    {
        int totalSeconds = 2549;
        int hours;
        int minutes;
        int seconds;
        int remainder;

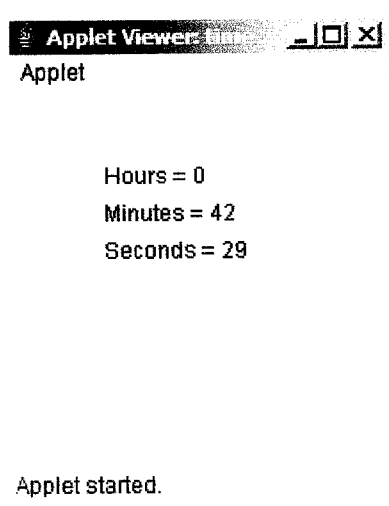
        hours = totalSeconds / 3600;      //3600 seconds/hr
        remainder = totalSeconds % 3600;  //how many seconds left for min:sec
        minutes = remainder / 60;         //60 seconds/min
        seconds = remainder % 60;         //remainder is :sec

        //Do the output
        g.drawString("Hours = " + hours, 50, 50);
        g.drawString("Minutes = " + minutes, 50, 70);
        g.drawString("Seconds = " + seconds, 50, 90);
    }
}

```



*demo - rlc*



```
/*  
 * Chris Tjon, Ex 5.2, Wed 4pm  
 *  
 * Draw a street of 4 houses 10 pixels apart  
 */
```

```
package drawhouses;
```

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

```
public class DrawHouses extends Applet  
{
```

```
    public void paint(Graphics g)  
    {  
        drawStreet(g, 30, 50, 50);  
    }
```

```
    private void drawStreet(Graphics g, int wallHeight, int bottomX, int bottomY)  
    {
```

```
        int i;  
        int gap = 10;  
        int width = wallHeight;  
        int numHouses = 4;
```

```
        for (i = 0; i < numHouses; i++)
```

```
        {  
            drawHouse(g, (bottomX + (width * i) + (gap * i)), bottomY, width, wallHeight);  
        }
```

```
    }
```

```
    private void drawTriangle(Graphics g, int bottomX, int bottomY, int base, int height)
```

```
    {
```

```
        g.drawLine(bottomX, bottomY, bottomX + base, bottomY);  
        g.drawLine(bottomX + base, bottomY, bottomX + base/2, bottomY - height);  
        g.drawLine(bottomX + base/2, bottomY - height, bottomX, bottomY);
```

```
    }
```

```
    private void drawHouse(Graphics g, int bottomX, int bottomY, int width, int height)
```

```
    {
```

```
        g.drawRect(bottomX, bottomY - height, width, height);  
        drawTriangle(g, bottomX, bottomY - height, width, height/2);
```

```
    }
```

```
}
```

(5)

*demo ok*

*> goal*