Joel Edwards
Course: Java Programming 1
Homework 7
April 15, 2011

Source:

WindChill.java:

```
import java.applet.Applet;
import java.awt.Button;
import java.awt.Checkbox;
import java.awt.CheckboxGroup;
import java.awt.Color;
import java.awt.event.AdjustmentListener;
import java.awt.event.AdjustmentEvent;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Panel;
import java.awt.Scrollbar;

public class WindChill
    extends Applet
    implements AdjustmentListener, ItemListener
{
    public static final long serialVersionUID = 1L;

    private Scrollbar speedSlider = new
Scrollbar(Scrollbar.HORIZONTAL, 10, 10, 5, 60);
    private Label speedLabel = new Label();
    private CheckboxGroup groupSpeed = new CheckboxGroup();
    private Checkbox radioSpeedM = new Checkbox("MPH (Miles per
Hour)", groupSpeed, true);
    private Checkbox radioSpeedK = new Checkbox("KPH (Kilometers per
Hour)", groupSpeed, false);

    private Scrollbar tempSlider = new
Scrollbar(Scrollbar.HORIZONTAL, 32, 30, -50, 120);
    private Label tempLabel = new Label();
    private CheckboxGroup groupTemp = new CheckboxGroup();
    private Checkbox radioTempF = new Checkbox("°F (Farenheit)",
groupTemp, true);
    private Checkbox radioTempC = new Checkbox("°C (Celcius)",
groupTemp, false);

    private Label indexLabel = new Label("");
```

```
public void init() {
    setLayout(new GridLayout(3,1));
    Font font = getFont();
    setFont(font.deriveFont(Font.BOLD));

    Panel speedPanel = new Panel(new GridLayout(4,1));
    speedPanel.add(speedSlider);
    speedPanel.add(speedLabel);
    speedPanel.add(radioSpeedM);
    speedPanel.add(radioSpeedK);
    speedPanel.setBackground(color(0.8, 0.8, 1.0));
    add(speedPanel);

    Panel tempPanel = new Panel(new GridLayout(4,1));
    tempPanel.add(tempSlider);
    tempPanel.add(tempLabel);
    tempPanel.add(radioTempF);
    tempPanel.add(radioTempC);
    tempPanel.setBackground(color(1.0, 0.5999999, 0.0));
    add(tempPanel);

    Panel displayPanel = new Panel(new GridLayout(1,1));
    displayPanel.add(indexLabel);
    displayPanel.setBackground(color(0.8, 1.0, 0.4));

    add(displayPanel);

    // Configure widgets
    speedSlider.addAdjustmentListener(this);
    tempSlider.addAdjustmentListener(this);

    radioSpeedM.addItemListener(this);
    radioSpeedK.addItemListener(this);
    radioTempF.addItemListener(this);
    radioTempC.addItemListener(this);

    // Calculater initial index
    reportWindSpeed();
    reportTemperature();
    updateIndex();
}

private Color color(double r, double g, double b) {
    return new Color((float)r, (float)g, (float)b);
}

private void updateIndex() {
    double windSpeed = speedSlider.getValue();
```

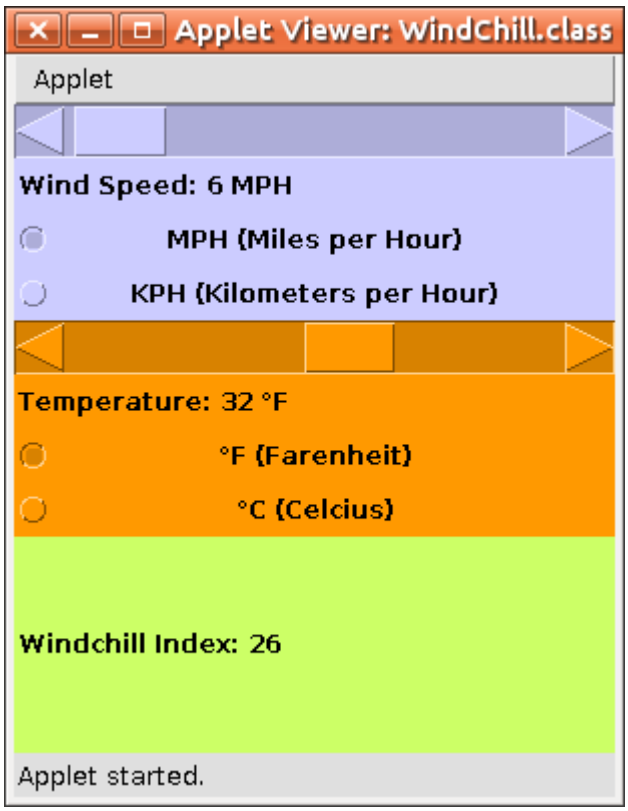
```
        double temperature = tempSlider.getValue();
        double x = 0.303439 * Math.sqrt(windSpeed) - 0.0202886 *
windSpeed;
        double index = 91.9 - (91.4 - temperature) * (x + 0.474266);

        index = radioTempF.getState() ? index : (index - 32) *
(5.0/9.0);

        indexLabel.setText("Windchill Index: " +(int)index);
    }

    private void reportWindSpeed() {
        int windSpeed = speedSlider.getValue();
        String units = "MPH";
        if (radioSpeedK.getState()) {
            units = "KPH";
            windSpeed *= 1.609344;
        }
        speedLabel.setText("Wind Speed: " +windSpeed+ " " +units);
    }

    private void reportTemperature() {
        int temperature = tempSlider.getValue();
        String units = "°F";
        if (radioTempC.getState()) {
            temperature = (int)((temperature - 32) * (5.0 / 9.0));
            units = "°C";
        }
        tempLabel.setText("Temperature: " +temperature+ " " +units);
    }

    public void itemStateChanged(ItemEvent evt) {
        Object source = evt.getSource();
        if ((source == radioSpeedM) || (source == radioSpeedK)) {
            reportWindSpeed();
            updateIndex();
        } else if ((source == radioTempF) || (source == radioTempC))
{
            reportTemperature();
            updateIndex();
        }
    }

    public void adjustmentValueChanged(AdjustmentEvent evt) {
        Object source = evt.getSource();
        if (source == speedSlider) {
            reportWindSpeed();
            updateIndex();
        } else if (source == tempSlider) {
```

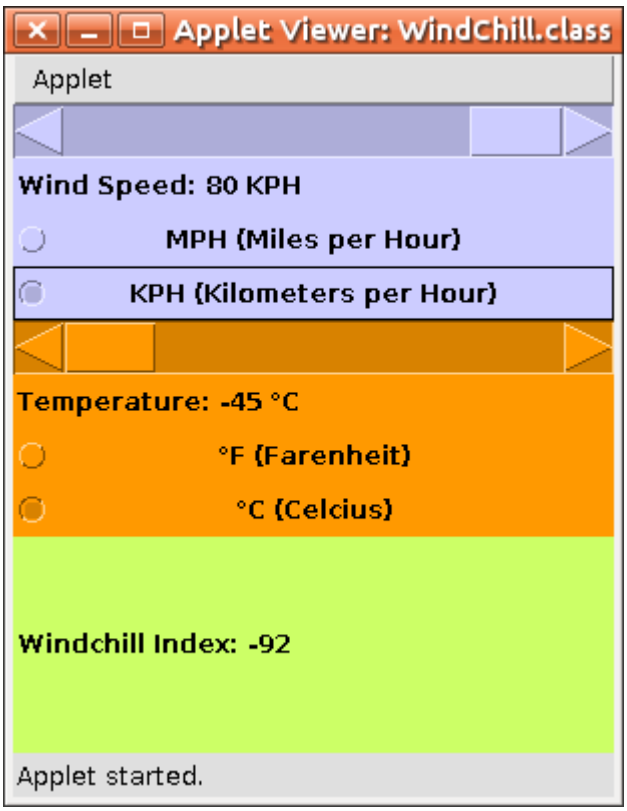
```
                reportTemperature();
                updateIndex();
            }
        }
}
```

WindChill.html:

```
<html>
    <head>
        <title>Wind Chill Index</title>
        <meta http-equiv="pragma" content="no-cache" />
    </head>
    <body>
        <applet code="WindChill.class" width="300" height="325">
        </applet>
    </body>
</html>
```

Test Output:





Webpage:
http://scaglietti.dyndns.org/WindChill/