

Java Applets and Audio

❖ Java applets

- ◆ When browsing the Web, you often see the graphical user interface and animation developed using Java
- ◆ These programs are called Java applets

❖ Class extends JApplet to make a Java applet

- ◆ Class extends JFrame to make a Java application
- ◆ Applets do not have a main() method
- ◆ Applets are invoked by browser as described in the HTML code of a web page
- ◆ HTML file contains `<applet>` `</applet>` element
 - ◆ Note the open applet tag and closing tag

❖ But first lets look at more Java GUI application examples

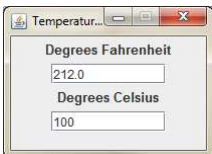
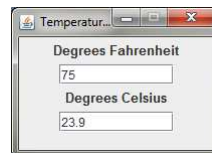
Copyright © 2012 R.M. Laurie 1

```

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4. public class TempConvert extends JFrame {
5.     private JTextField txtFahr = new JTextField("0", 10);
6.     private JTextField txtCelc = new JTextField("0", 10);
7.     public static void main(String[] args) {
8.         TempConvert fraWindow = new TempConvert();
9.         fraWindow.setTitle("Temperature Converter");
10.        fraWindow.setSize(200,140); // width, height
11.        fraWindow.setLocationRelativeTo(null); // Center the frame
12.        fraWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13.        fraWindow.setResizable(false);
14.        fraWindow.setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));
15.        fraWindow.setVisible(true);
16.    }
17.    public TempConvert() {
18.        /** GUI components */
19.        JLabel lblFahr = new JLabel("Degrees Fahrenheit");
20.        lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
21.        add(lblFahr);
22.        add(txtFahr);
23.        JLabel lblCelsius = new JLabel("Degrees Celsius");
24.        lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
25.        add(lblCelsius);
26.        add(txtCelc);
27.        /** Event Handlers */
28.        txtCelc.addActionListener(new ActionListener() {
29.            public void actionPerformed(ActionEvent event) {
30.                double dFahr = Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
31.                txtFahr.setText(String.format("%.1f", dFahr));
32.            }
33.        });
34.        txtFahr.addActionListener(new ActionListener() {
35.            public void actionPerformed(ActionEvent event) {
36.                double dCelc = (Double.parseDouble(txtFahr.getText()) - 32) * 5 / 9;
37.                txtCelc.setText(String.format("%.1f", dCelc));
38.            }
39.        });
40.    }
41. }
```

A Simple Temperature Converter

Enter a temperature and enter key to fire event

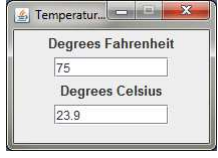



Windows Builder Version

Enter a temperature and enter key to fire event

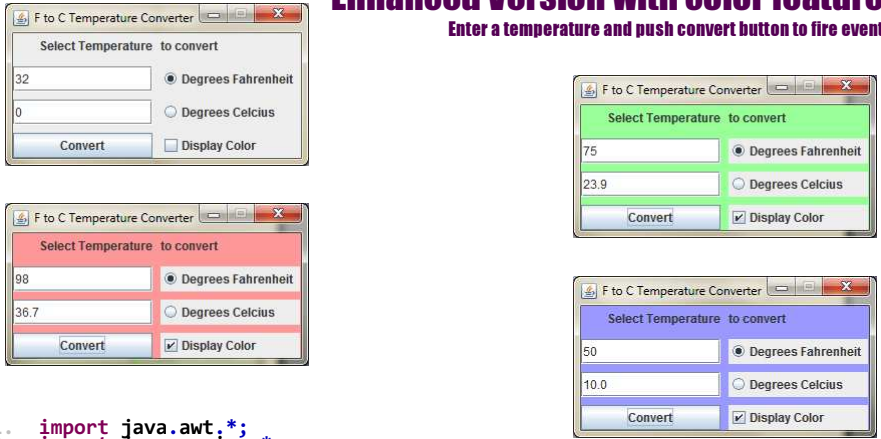
```

1. import java.awt.*;
2. import javax.swing.*;
3. import javax.swing.border.*;
4. import java.awt.event.*;
5. public class TempConvertBuilder extends JFrame {
6.     private JPanel panMain;
7.     private JTextField txtFahr;
8.     private JTextField txtCelc;
9.     public static void main(String[] args) {
10.        EventQueue.invokeLater(new Runnable() {
11.            public void run() {
12.                try {
13.                    TempConvertBuilder frame = new TempConvertBuilder();
14.                    frame.setVisible(true);
15.                } catch (Exception e) {
16.                    e.printStackTrace();
17.                }
18.            }
19.        });
20.    }
21.    public TempConvertBuilder() {
22.        setResizable(false);
23.        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24.        setBounds(100, 100, 160, 140);
25.        panMain = new JPanel();
26.        panMain.setBorder(new EmptyBorder(5, 5, 5, 5));
27.        setContentPane(panMain);
28.        panMain.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
29.        JLabel lblFahr = new JLabel("Degrees Fahrenheit");
30.        panMain.add(lblFahr);
31.        txtFahr = new JTextField();
32.        txtFahr.addActionListener(new ActionListener() {
33.            public void actionPerformed(ActionEvent arg0) {
34.                double dCelc = (Double.parseDouble(txtFahr.getText()) - 32) * 5 / 9;
35.                txtCelc.setText(String.format("%.1f", dCelc));
36.            }
37.        });
38.        txtFahr.setText("0");
39.        panMain.add(txtFahr);
40.        txtFahr.setColumns(10);
41.        JLabel lblCelc = new JLabel("Degrees Celcius");
42.        panMain.add(lblCelc);
43.        txtCelc = new JTextField();
44.        txtCelc.addActionListener(new ActionListener() {
45.            public void actionPerformed(ActionEvent e) {
46.                double dCelc = Double.parseDouble(txtFahr.getText()) * 9 / 5 + 32;
47.                txtFahr.setText(String.format("%.1f", dCelc));
48.            }
49.        });
50.        txtCelc.setText("0");
51.        panMain.add(txtCelc);
52.        txtCelc.setColumns(10);
53.    }
54. }
    
```



Enhanced Version with color feature

Enter a temperature and push convert button to fire event



```

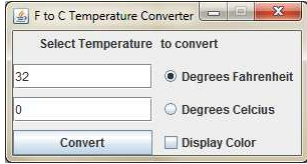
1. import java.awt.*;
2. import javax.swing.*;
3. import java.awt.event.*;
4. public class ConvCFK extends JFrame {
5.     private JPanel panMain;
6.     private JTextField txtCelc, txtFahr;
7.     private JRadioButton radCelc, radFahr;
8.     private JLabel lblTitleL, lblTitleR;
9.     private JCheckBox chkColor;
10.    private JButton btnConvert;
11.    public static void main(String[] args) {
12.        ConvCFK fraMain = new ConvCFK();
13.        fraMain.setVisible(true);
14.    }
    
```

```

1. public ConvCFK() {
2.     setTitle("F to C Temperature Converter");
3.     setResizable(false);
4.     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
5.     setBounds(100, 100, 293, 152);
6.     panMain = new JPanel();
7.     setContentPane(panMain);
8.     panMain.setLayout(new GridLayout(4, 2, 8, 8));
9.     lblTitleL = new JLabel("Select Temperature");
10.    lblTitleL.setHorizontalAlignment(SwingConstants.RIGHT);
11.    panMain.add(lblTitleL);
12.    lblTitleR = new JLabel("to convert");
13.    panMain.add(lblTitleR);
14.    txtFahr = new JTextField("32", 10);
15.    panMain.add(txtFahr);
16.    radFahr = new JRadioButton("Degrees Fahrenheit");
17.    radFahr.setSelected(true);
18.    panMain.add(radFahr);
19.    txtCelc = new JTextField("0", 10);
20.    panMain.add(txtCelc);
21.    radCelc = new JRadioButton("Degrees Celcius");
22.    panMain.add(radCelc);
23.    btnConvert = new JButton("Convert");
24.    panMain.add(btnConvert);
25.    chkColor = new JCheckBox("Display Color");
26.    panMain.add(chkColor);
27.    ButtonGroup bgpForC = new ButtonGroup();
28.    bgpForC.add(radCelc);
29.    bgpForC.add(radFahr);
30.    btnConvert.addActionListener(new ActionListener() {
31.        public void actionPerformed(ActionEvent btnPush) {
32.            if(radCelc.isSelected()) {
33.                double dFahr = Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
34.                txtFahr.setText(String.format("%.1f", dFahr));
35.            }
36.            else {
37.                double dCelc = (Double.parseDouble(txtFahr.getText()) - 32) * 5 / 9;
38.                txtCelc.setText(String.format("%.1f", dCelc));
39.            }
40.            if(chkColor.isSelected()){
41.                double dFahr = Double.parseDouble(txtFahr.getText());
42.                if(dFahr <= 32) panMain.setBackground(new Color(0xFF, 0xFF, 0xFF));
43.                else if(dFahr <= 60) panMain.setBackground(new Color(0x99, 0x99, 0xFF));
44.                else if(dFahr <= 85) panMain.setBackground(new Color(0x99, 0xFF, 0x99));
45.                else panMain.setBackground(new Color(0xFF, 0x99, 0x99));
46.            }
47.            else panMain.setBackground(new Color(240, 240, 240));
48.        }
49.    });
50. }
51. }
                
```

Temperature Converter

Only Click button event causes processing to occur



The <applet> HTML Tag

```

<applet
  code=classname.class
  width=applet_viewing_width_in_pixels
  height=applet_viewing_height_in_pixels
  [archive=archivefile]
  [codebase=applet_url]
  [vspace=vertical_margin]
  [hspace=horizontal_margin]
  [align=applet_alignment]
  [alt=alternative_text]
>
<param name=param_name1
  value=param_value1>
</applet>
                
```

Required attributes are:
code, width, and height
All others are optional

Copyright © 2012 R.M. Laurie 6

```

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*; JApplet
4. public class TempConvert extends JFrame {
5.     private JTextField txtFahr = new JTextField("0", 10);
6.     private JTextField txtCelc = new JTextField("0", 10);
7.     public static void main(String[] args) {
8.         TempConvert frawindow = new TempConvert();
9.         frawindow.setTitle("Temperature Converter");
10.        frawindow.setSize(200,140); // width, height
11.        frawindow.setLocationRelativeTo(null); // Center the frame
12.        frawindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13.        frawindow.setResizable(false);
14.        frawindow.setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));
15.        frawindow.setVisible(true);
16.    }
17.    public TempConvert() {
18.        setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6)); // Add the FlowLayout
19.        JLabel lblFahr = new JLabel("Degrees Fahrenheit");
20.        lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
21.        add(lblFahr);
22.        add(txtFahr);
23.        JLabel lblCelsius = new JLabel("Degrees Celsius");
24.        lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
25.        add(lblCelsius);
26.        add(txtCelc);
27.        /* Event Handlers */
28.        txtCelc.addActionListener(new ActionListener() {
29.            public void actionPerformed(ActionEvent event) {
30.                double dFahr = Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
31.                txtFahr.setText(String.format("%.1f", dFahr));
32.            }
33.        });
34.        txtFahr.addActionListener(new ActionListener() {
35.            public void actionPerformed(ActionEvent event) {
36.                double dCelc = (Double.parseDouble(txtFahr.getText()) - 32) * 5 / 9;
37.                txtCelc.setText(String.format("%.1f", dCelc));
38.            }
39.        });
40.    }
41. }

```

Convert Application to an Applet

- 1) Replace JFrame with JApplet
- 2) Create same layout in Applet
- 3) Delete main() method
- 4) Delete all frame methods

Applications vs. Applets

- ❖ **Similarities**
 - ◆ Since JFrame and JApplet both are subclasses of the Container class, all the user interface components, layout managers, and event-handling features are the same for both classes.
- ❖ **Differences**
 - ◆ Applications are invoked from the static main method by the Java interpreter, and applets are run by the Web browser.
 - ◆ The Web browser creates an instance of the applet using the applet's no-arg constructor and controls and executes the applet through the init, start, stop, and destroy methods.
 - ◆ Applets have security restrictions
 - ◆ Web browser creates graphical environment for applets, GUI applications are placed in a frame.

```

1. <html>
2. <head>
3. <title>This is my first Applet</title>
4. </head>
5. <body>
6. <applet code="TempConvertApplet.class" width=200 height=140> </applet>
7. </body>
8. </html>

```

Copyright © 2012 R.M. Laurie 8

Security Restrictions on Applets

❖ Security Restrictions on Applets

- ◆ Applets are not allowed to read from, or write to, the file system of the computer running applets
- ◆ Applets are not allowed to run any programs on the browser's computer
- ◆ Applets are not allowed to establish connections between the user's computer and another computer except with the server where the applets are stored

❖ Conversion Between Applications and Applets

- ◆ Conversions between applications and applets are simple
- ◆ You can always convert an applet into an application, just add static main method
- ◆ You can convert an application to an applet as long as security restrictions are not violated

Copyright © 2012 R.M. Laurie 9

```

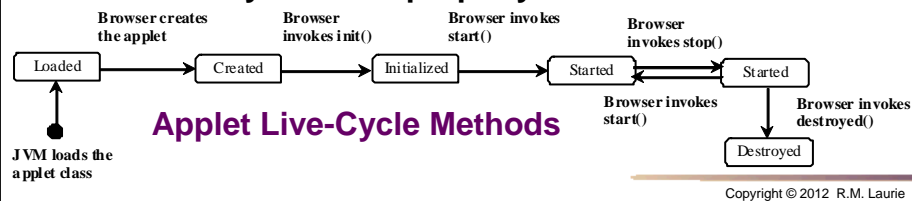
1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4. public class TempConvertAppletAp extends JApplet {
5.     private JTextField txtFahr = new JTextField("0", 10);
6.     private JTextField txtCelc = new JTextField("0", 10);
7.     public TempConvertAppletAp() {
8.         /** GUI components */
9.         setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));
10.        JLabel lblFahr = new JLabel("Degrees Fahrenheit");
11.        lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
12.        add(lblFahr);
13.        add(txtFahr);
14.        JLabel lblCelsius = new JLabel("Degrees Celsius");
15.        lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
16.        add(lblCelsius);
17.        add(txtCelc);
18.        /** Event Handlers */
19.        txtCelc.addActionListener(new ActionListener() {
20.            public void actionPerformed(ActionEvent event) {
21.                double dFahr = Double.parseDouble(txtFahr.getText()) * 9 / 5 + 32;
22.                txtFahr.setText(String.format("%.1f", dFahr));
23.            }
24.        });
25.        txtFahr.addActionListener(new ActionListener() {
26.            public void actionPerformed(ActionEvent event) {
27.                double dCelc = (Double.parseDouble(txtFahr.getText()) - 32) * 5 / 9;
28.                txtCelc.setText(String.format("%.1f", dCelc));
29.            }
30.        });
31.    }
32.    public static void main(String[] args) {
33.        JFrame fraBack2Application = new JFrame("Applet is in frame");
34.        TempConvertAppletAp appletMain = new TempConvertAppletAp();
35.        fraBack2Application.add(appletMain);
36.        fraBack2Application.setSize(200, 80);
37.        fraBack2Application.setLocationRelativeTo(null);
38.        fraBack2Application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
39.        fraBack2Application.setVisible(true);
40.    }
41. }
```

Enabling Applet to Run also as Application

- 1) Create main() method
- 2) Create instance of JFrame
- 3) Create instance of applet
- 4) Add instance of applet to JFrame
- 5) Within main() set window properties

The Applet Class

- ❖ When the applet is loaded, the Web browser creates an instance of the applet by invoking the applet's no-arg constructor
 - ◆ The browser uses the init, start, stop, and destroy methods to control the applet during life-cycle
 - ◆ By default, these methods do nothing
 - ◆ To perform specific functions, they need to be modified in the user's applet so that the browser can call your code properly



The `init()` and `start()` Methods

- ❖ The `init()` Method
 - ◆ Invoked when the applet is first loaded and again if the applet is reloaded
 - ◆ A subclass of Applet should override this method if the subclass has an initialization to perform
 - ◆ Functions often implemented in this method include GUI components, and getting string parameter values from the <applet> tag in the HTML page
- ❖ The `start()` Method
 - ◆ Invoked after the `init()` method is executed or whenever the applet becomes active again after a period of inactivity (for example, when the user returns to the page containing the applet after surfing other Web pages)
 - ◆ A subclass of Applet overrides this method if it has any operation that needs to be performed whenever the Web page containing the applet is visited
 - ◆ An applet with animation, for example, might use the `start` method to resume animation

The stop() and destroy() Methods

❖ The stop() Method

- ◆ Invoked when the user moves off the page
- ◆ Overrides this method if any operation needs to be performed each time the Web page containing the applet is no longer visible
- ◆ When the user leaves the page, any threads the applet has started but not completed will continue to run
- ◆ You should override the stop method to suspend the running threads so that the applet does not use system resources when it is inactive

❖ The destroy() Method

- ◆ Invoked when the browser exits normally and applet should release any system resources it has allocated
- ◆ A subclass of Applet overrides this method if it has any operation that needs to be performed before it is destroyed
- ◆ Usually, you won't need to override this method unless you wish to release specific resources, such as threads that the applet created

Copyright © 2012 R.M. Laurie 13

Writing Applets

- ❖ Always extends the `JApplet` class, which is a subclass of `Applet` for Swing components
- ❖ Override `init()`, `start()`, `stop()`, and `destroy()` if necessary
 - ◆ By default, these methods are empty
- ❖ Add your own methods and data if necessary
- ❖ Applets are embedded in an HTML page
 - ◆ Can pass arguments from web page to Applet parameter list
 - ◆ `<param name = Cinit value = 100 />`

```
1. <html>
2. <head> <title>Applet with Parameters</title> </head>
3. <body>
4. <applet code="TempConvertAppletParam.class" width="180" height="140">
5. <param name="Cint" value="100" />
6. <param name="Fint" value="212" />
7. </applet>
8. </body>
9. </html>
```

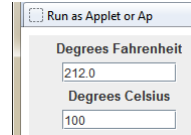
Copyright © 2012 R.M. Laurie 14

Applet with parameters that receives from HTML page

- 1) Nest <param /> elements with <applet> element
- 2) utilize getParameter within Java Applet

```

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4. public class TempConvertAppletParam extends JApplet {
5.     private JTextField txtFahr = new JTextField("0", 10);
6.     private JTextField txtCelc = new JTextField("0", 10);
7.     public TempConvertAppletParam() {
8.         /** GUI components */
9.         setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));
10.        JLabel lblFahr = new JLabel("Degrees Fahrenheit");
11.        lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
12.        add(lblFahr);
13.        add(txtFahr);
14.        JLabel lblCelsius = new JLabel("Degrees Celsius");
15.        lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
16.        add(lblCelsius);
17.        add(txtCelc);
18.        /** Event Handlers */
19.        txtCelc.addActionListener(new ActionListener() {
20.            public void actionPerformed(ActionEvent event) {
21.                double dFahr = Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
22.                txtFahr.setText(String.format("%.1f", dFahr));
23.            }
24.        });
25.        txtFahr.addActionListener(new ActionListener() {
26.            public void actionPerformed(ActionEvent event) {
27.                double dCelc = (Double.parseDouble(txtFahr.getText()) - 32) * 5 / 9;
28.                txtCelc.setText(String.format("%.1f", dCelc));
29.            }
30.        });
31.    }
32.    public void init() {
33.        txtFahr.setText(getParameter("Fint"));
34.        txtCelc.setText(getParameter("Cint"));
35.    }
36. }
    
```



Locating Resource from Applets

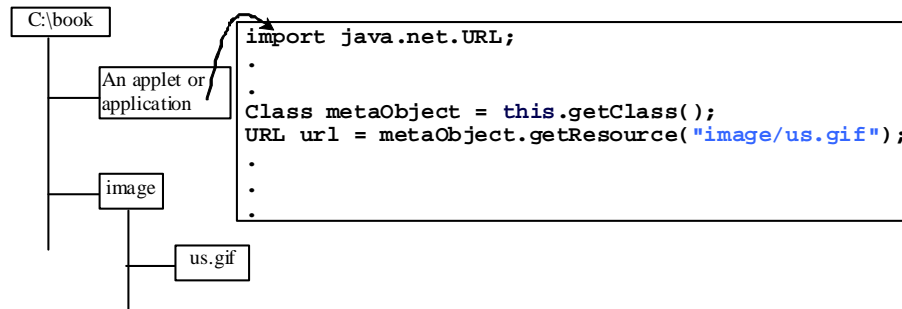
- ❖ Due to security restrictions, applets cannot access local files
- ❖ Creating ImageIcon using relative path filename


```
ImageIcon imageIcon = new ImageIcon("image/us.gif");
lbl.setIcon(imageIcon);
```

 - ◆ Works fine with Java applications on all platforms
 - ◆ Does not work with Java applets because applets cannot load local files
 - ◆ Need to locate the file using the URL class, to make it to work with both applications and applets
- ❖ A resource can be something a file or a directory
- ❖ The java.net.URL class can be used to identify files (image, audio, text, etc.) on the Internet
 - ◆ In general, a URL (Uniform Resource Locator) is a pointer to a "resource" on the World Wide Web, local machine, or remote web host

Creating a URL from a Class Reference

A URL for a file can also be accessed from a class in a way that is independent of the location of the file, as long as the resource file is located in the class directory



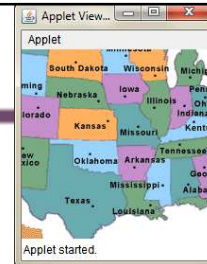
Copyright © 2012 R.M. Laurie 17

Displaying Image located in folder

```

1. import java.net.URL;
2. import javax.swing.*;
3. public class DisplayImageUsingURL extends JApplet
4. {
5.     public DisplayImageUsingURL()
6.     {
7.         // resources folder is located in bin folder with files.class
8.         // this.getClass() accesses current folder location of class file
9.         URL urlMap = this.getClass().getResource("resources/us-map.gif");
10.        ImageIcon imgMap = new ImageIcon(urlMap);
11.        JLabel lblMap = new JLabel(imgMap);
12.        add(lblMap);
13.    }
14.    /** Main method */
15.    public static void main(String[] args)
16.    {
17.        // Create a frame
18.        JFrame fraMain = new JFrame("Display Image Using URL");
19.        fraMain.setSize(500, 400);
20.        // Create an instance of the applet
21.        DisplayImageUsingURL appletMap = new DisplayImageUsingURL();
22.        // Add the applet instance to the frame
23.        fraMain.getContentPane().add(appletMap);
24.        fraMain.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25.        fraMain.setVisible(true);
26.    }
27. }

```



Copyright © 2012 R.M. Laurie 18

Creating AudioClip from an Audio File

- ❖ Create an audio clip object for the audio file
 - ◆ Can be played repeatedly without reloading the file
 - ◆ Use the static method newAudioClip() in the java.applet.Applet class

```
AudioClip audioClip = Applet.newAudioClip(url);
```

- ❖ Audio was originally used with Java applets
 - ◆ Therefore, AudioClip interface is in the java.applet package
 - ◆ Can also be used in Java applications
 - ◆ Can utilize only WAV, AU, MIDI, RMF audio file formats in Java

- ❖ The following statements, for example, create an AudioClip for the beep.au audio file in the same directory with the class you are running

```
Class theClass = this.getClass();
URL theUrl = theClass.getResource("beep.au");
AudioClip audioClip = Applet.newAudioClip(theUrl);
```

Copyright © 2012 R.M. Laurie 19

```

1. import javax.swing.*;
2. import java.net.URL;
3. import java.applet.*;
4. import java.awt.*;
5. import java.awt.event.*;
6. public class PlayAudio extends JApplet {
7.     private AudioClip audioClip;
8.     JButton btnStop = new JButton("Stop");
9.     JButton btnPlay = new JButton("Play");
10.    public PlayAudio() {
11.        JPanel panMain = new JPanel(new FlowLayout());
12.        URL urlImage = getClass().getResource("resources/JapanFlag.gif");
13.        panMain.add(new JLabel(new ImageIcon(urlImage)));
14.        URL urlAudio = getClass().getResource("resources/Kimi_ga_Yo.wav");
15.        audioClip = Applet.newAudioClip(urlAudio);
16.        audioClip.play();
17.        panMain.add(btnStop);
18.        panMain.add(btnPlay);
19.        add(panMain);
20.        btnStop.addActionListener( // Note open parenthesis
21.            new ActionListener() { // Anonymous Action Listener btnYes starts here
22.                public void actionPerformed(ActionEvent theEvent) {
23.                    if (audioClip != null) audioClip.stop();
24.                } // Closing actionPerformed method
25.            }); // Closing Anonymous action listener object for btnYes
26.        btnPlay.addActionListener( // Note open parenthesis
27.            new ActionListener() { // Anonymous Action Listener btnYes starts here
28.                public void actionPerformed(ActionEvent theEvent) {
29.                    if (audioClip != null) audioClip.play();
30.                } // Closing actionPerformed method
31.            }); // Closing Anonymous action listener object for btnYes
32.    }
33.    public static void main(String[] args) {
34.        JFrame fraMain = new JFrame("Display Image and Play Audio");
35.        PlayAudio applet = new PlayAudio();
36.        fraMain.add(applet, BorderLayout.CENTER);
37.        fraMain.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
38.        fraMain.setSize(400, 340);
39.        fraMain.setResizable(false);
40.        fraMain.setVisible(true);
41.    }
42. }
43. }
44. }
```

Displaying Image and playing audio



Resource import and Jar export

- ❖ Create a new package and name it resources
 - ◆ Import the files into the resources package individually
- ❖ Projects can be exported as Jar files
 - ◆ File > Export > Java Jar File
 - ◆ Upload to web site
- ❖ HTML code to access Jar file and execute applet stored in Jar file

```

1. <html>
2. <head><title>Audio URL</title> </head>
3. <body>
4. </applet>
5. <APPLET CODE="SoundJar.class" ARCHIVE="japan.jar" WIDTH=400 HEIGHT=340>
6. </APPLET>
7. </body>
8. </html>
    
```

Copyright © 2012 R.M. Laurie | 21

Sounds can be mixed as shown in this example

```

1. import javax.swing.*;
2. import java.net.URL;
3. import java.applet.*;
4. import java.awt.*;
5. import java.awt.event.*;
6. public class GunMixer extends JApplet {
7.     private AudioClip snd50cal;
8.     private AudioClip sndAk47;
9.     public GunMixer() {
10.         JPanel panMain = new JPanel(new BorderLayout(10, 10));
11.         JLabel lblCaption = new JLabel("Which gun is firing?");
12.         lblCaption.setFont(new Font("Verdana", Font.BOLD+Font.ITALIC, 28));
13.         lblCaption.setHorizontalAlignment(SwingConstants.CENTER);
14.         panMain.add(lblCaption, BorderLayout.NORTH);
15.         URL url150Cal = getClass().getResource("resources/50cal.png");
16.         URL url1Ak47 = getClass().getResource("resources/ak47ru.jpg");
17.         JButton btn50Cal = new JButton(new ImageIcon(url150Cal));
18.         JButton btnAk47 = new JButton(new ImageIcon(url1Ak47));
19.         panMain.add(btn50Cal, BorderLayout.WEST);
20.         panMain.add(btnAk47, BorderLayout.EAST);
21.         URL url1Snd50Cal = getClass().getResource("resources/30cal5.wav");
22.         URL url1SndAk47 = getClass().getResource("resources/ak47.wav");
23.         snd50cal = Applet.newAudioClip(url1Snd50Cal);
24.         sndAk47 = Applet.newAudioClip(url1SndAk47);
25.         add(panMain);
26.         btn50Cal.addActionListener( new ActionListener() {
27.             public void actionPerformed(ActionEvent theEvent) {
28.                 if (snd50cal != null) snd50cal.play();
29.             }
30.         });
31.         btnAk47.addActionListener( new ActionListener() {
32.             public void actionPerformed(ActionEvent theEvent) {
33.                 if (sndAk47 != null) sndAk47.play();
34.             }
35.         });
36.     }
37.     public static void main(String[] args) {
38.         JFrame fraMain = new JFrame("Name that Gun");
39.         GunMixer applet = new GunMixer();
40.         fraMain.add(applet, BorderLayout.CENTER);
41.         fraMain.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
42.         fraMain.setSize(500, 250);
43.         fraMain.setResizable(false);
44.         fraMain.setVisible(true);
45.     }
46. }
    
```