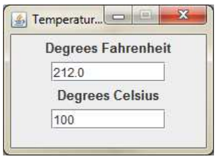
Chapter 18 Applets and Multimedia

- 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

A Simple Temperature Converter

Enter a temperature and enter key to fire event

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class TempConvert extends JFrame {
private JTextField txtFahr = new JTextField("0", 10);
private JTextField txtCelc = new JTextField("0", 10);
  public static void main(String[] args) {
     TempConvert fraWindow = new TempConvert();
     fraWindow.setTitle("Temperature Converter");
     fraWindow.setSize(200,140); // width, height
     fraWindow.setLocationRelativeTo(null); // Center the frame
     fraWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     fraWindow.setResizable(false);
     fraWindow.setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));
     fraWindow.setVisible(true);
  }
  public TempConvert() {
    /** GUI components */
    JLabel lblFahr = new JLabel("Degrees Fahrenheit");
    lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblFahr);
    add(txtFahr);
    JLabel lblCelsius = new JLabel("Degrees Celsius");
    lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblCelsius);
    add(txtCelc);
    /**  Event Handlers */
    txtCelc.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dFahr =  Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
        txtFahr.setText(String.format("%.1f", dFahr));
      }
    });
    txtFahr.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dCelc = (Double.parseDouble(txtFahr.getText()) - 32 ) * 5 / 9;
        txtCelc.setText(String.format("%.1f", dCelc));
      }
    });
  }
}
```

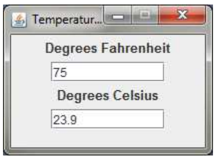


Degrees Fahrenheit

212.0

Degrees Celsius

100

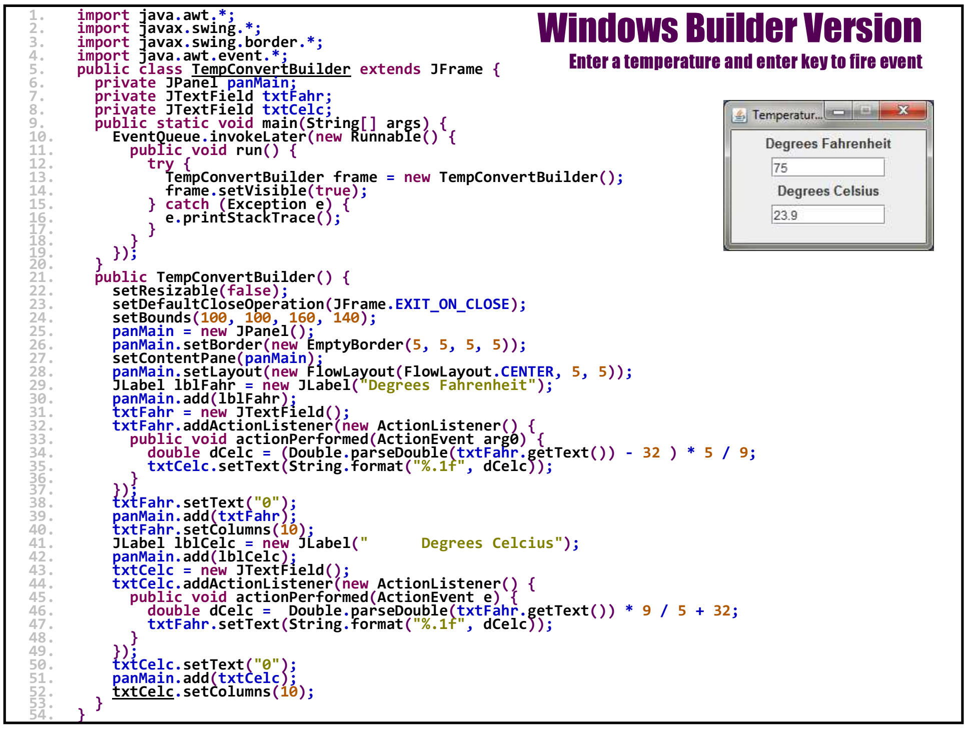


Degrees Fahrenheit

75

Degrees Celsius

23.9



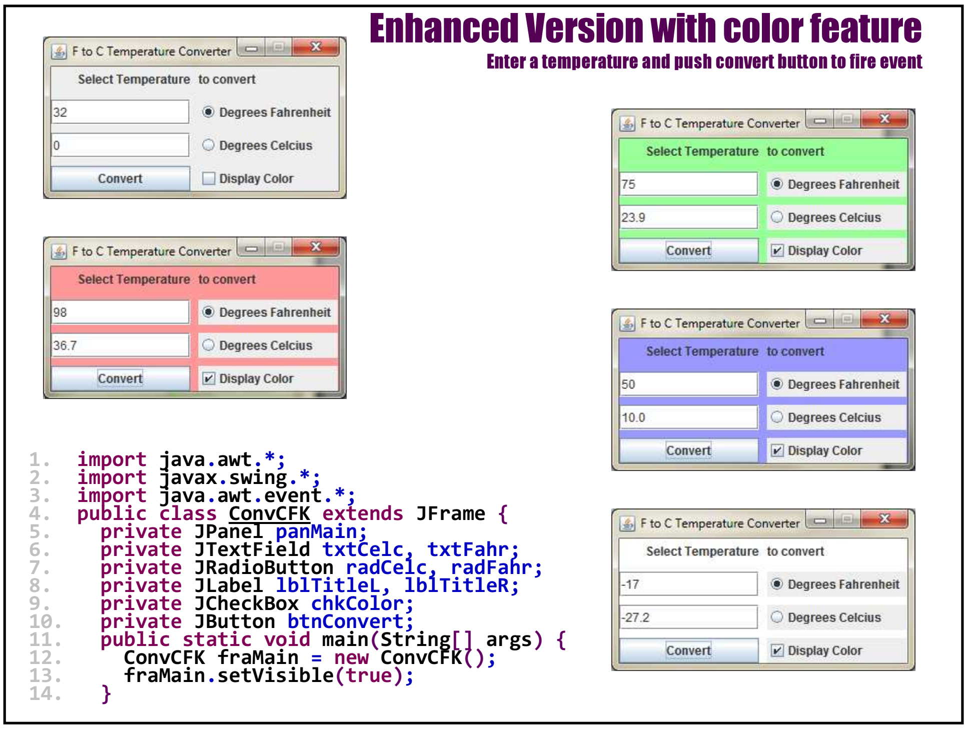
Windows Builder Version

Enter a temperature and enter key to fire event

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
public class TempConvertBuilder extends JFrame {
  private JPanel panMain;
  private JTextField txtFahr;
  private JTextField txtCelc;
  public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
      public void run() {
        try {
          TempConvertBuilder frame = new TempConvertBuilder();
          frame.setVisible(true);
        } catch (Exception e) {
          e.printStackTrace();
        }
      }
    });
  }
  public TempConvertBuilder() {
    setResizable(false);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 160, 140);
    panMain = new JPanel();
    panMain.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(panMain);
    panMain.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
    JLabel lblFahr = new JLabel("Degrees Fahrenheit");
    panMain.add(lblFahr);
    txtFahr = new JTextField();
    txtFahr.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent arg0) {
        double dCelc = (Double.parseDouble(txtFahr.getText()) - 32 ) * 5 / 9;
        txtCelc.setText(String.format("%.1f", dCelc));
      }
    });
    txtFahr.setText("0");
    panMain.add(txtFahr);
    txtFahr.setColumns(10);
    JLabel lblCelc = new JLabel("      Degrees Celcius");
    panMain.add(lblCelc);
    txtCelc = new JTextField();
    txtCelc.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        double dCelc =  Double.parseDouble(txtFahr.getText()) * 9 / 5 + 32;
        txtFahr.setText(String.format("%.1f", dCelc));
      }
    });
    txtCelc.setText("0");
    panMain.add(txtCelc);
    txtCelc.setColumns(10);
  }
}
```

Degrees Fahrenheit: 75

Degrees Celsius: 23.9



Enhanced Version with color feature

Enter a temperature and push convert button to fire event

32 ● Degrees Fahrenheit

0 ○ Degrees Celcius

□ Display Color

98 ● Degrees Fahrenheit

36.7 ○ Degrees Celcius

☑ Display Color

75 ● Degrees Fahrenheit

23.9 ○ Degrees Celcius

☑ Display Color

50 ● Degrees Fahrenheit

10.0 ○ Degrees Celcius

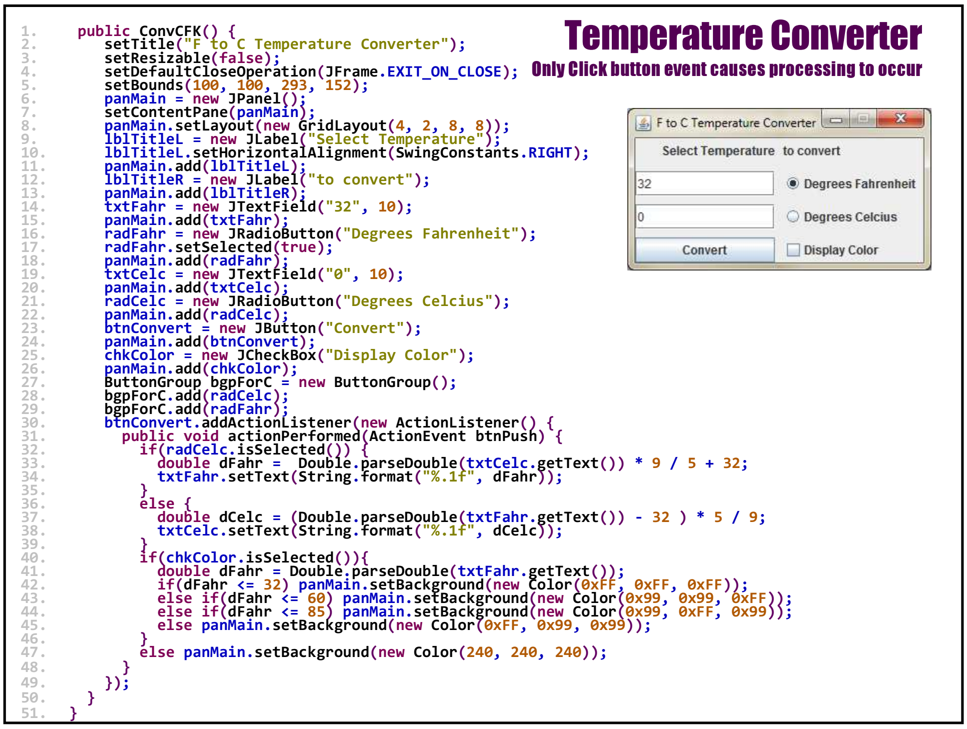
☑ Display Color

-17 ● Degrees Fahrenheit

-27.2 ○ Degrees Celcius

☑ Display Color

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class ConvCFK extends JFrame {
  private JPanel panMain;
  private JTextField txtCelc, txtFahr;
  private JRadioButton radCelc, radFahr;
  private JLabel lblTitleL, lblTitleR;
  private JCheckBox chkColor;
  private JButton btnConvert;
  public static void main(String[] args) {
    ConvCFK fraMain = new ConvCFK();
    fraMain.setVisible(true);
  }
```



Temperature Converter

Only Click button event causes processing to occur

```java
 public ConvCFK() {
    setTitle("F to C Temperature Converter");
    setResizable(false);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 293, 152);
    panMain = new JPanel();
    setContentPane(panMain);
    panMain.setLayout(new GridLayout(4, 2, 8, 8));
    lblTitleL = new JLabel("Select Temperature");
    lblTitleL.setHorizontalAlignment(SwingConstants.RIGHT);
    panMain.add(lblTitleL);
    lblTitleR = new JLabel("to convert");
    panMain.add(lblTitleR);
    txtFahr = new JTextField("32", 10);
    panMain.add(txtFahr);
    radFahr = new JRadioButton("Degrees Fahrenheit");
    radFahr.setSelected(true);
    panMain.add(radFahr);
    txtCelc = new JTextField("0", 10);
    panMain.add(txtCelc);
    radCelc = new JRadioButton("Degrees Celcius");
    panMain.add(radCelc);
    btnConvert = new JButton("Convert");
    panMain.add(btnConvert);
    chkColor = new JCheckBox("Display Color");
    panMain.add(chkColor);
    ButtonGroup bgpForC = new ButtonGroup();
    bgpForC.add(radCelc);
    bgpForC.add(radFahr);
    btnConvert.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent btnPush) {
        if(radCelc.isSelected()) {
          double dFahr =  Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
          txtFahr.setText(String.format("%.1f", dFahr));
        }
        else {
          double dCelc = (Double.parseDouble(txtFahr.getText()) - 32 ) * 5 / 9;
          txtCelc.setText(String.format("%.1f", dCelc));
        }
        if(chkColor.isSelected()){
          double dFahr = Double.parseDouble(txtFahr.getText());
          if(dFahr <= 32) panMain.setBackground(new Color(0xFF, 0xFF, 0xFF));
          else if(dFahr <= 60) panMain.setBackground(new Color(0x99, 0x99, 0xFF));
          else if(dFahr <= 85) panMain.setBackground(new Color(0x99, 0xFF, 0x99));
          else panMain.setBackground(new Color(0xFF, 0x99, 0x99));
        }
        else panMain.setBackground(new Color(240, 240, 240));
      }
    });
  }
}
```

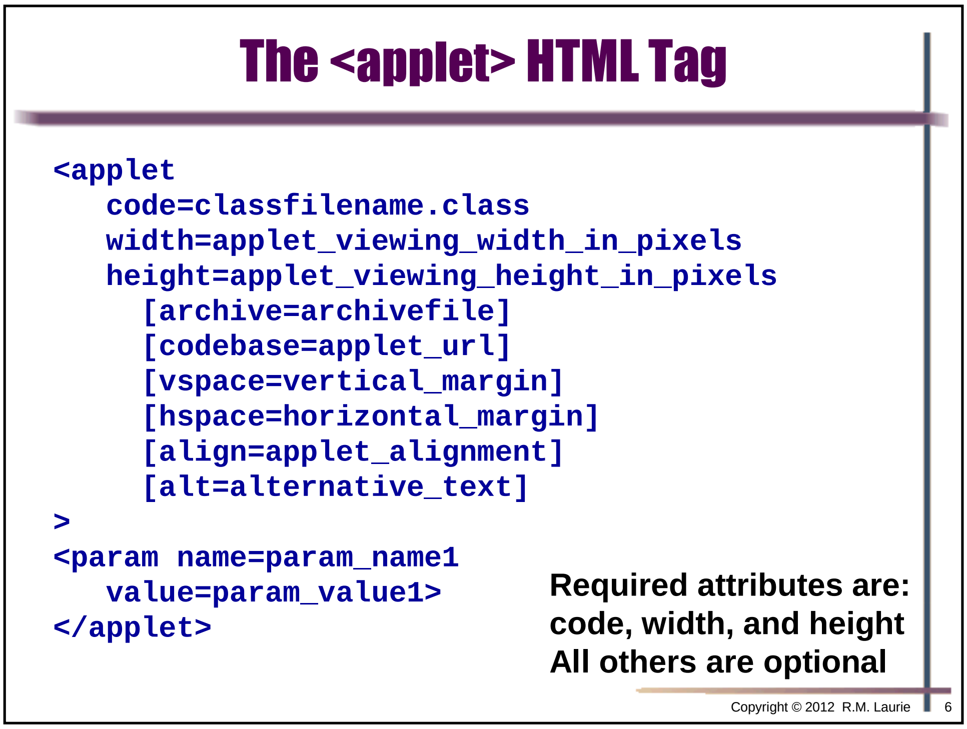
F to C Temperature Converter

Select Temperature to convert

32 ● Degrees Fahrenheit

0 ○ Degrees Celcius

Convert □ Display Color

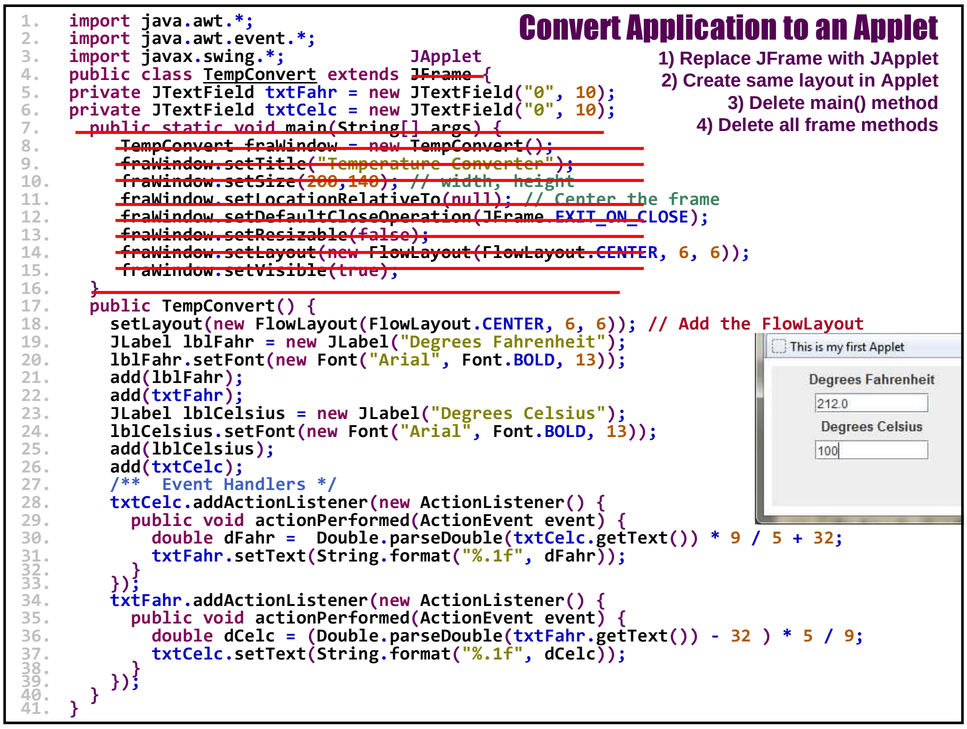


The <applet> HTML Tag

```
<applet
   code=classfilename.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



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

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;                   JApplet
public class TempConvert extends ~~JFrame~~ {
private JTextField txtFahr = new JTextField("0", 10);
private JTextField txtCelc = new JTextField("0", 10);
  ~~public static void main(String[] args) {~~
     ~~TempConvert fraWindow = new TempConvert();~~
     ~~fraWindow.setTitle("Temperature Converter");~~
     ~~fraWindow.setSize(200,140); // width, height~~
     ~~fraWindow.setLocationRelativeTo(null); // Center the frame~~
     ~~fraWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);~~
     ~~fraWindow.setResizable(false);~~
     ~~fraWindow.setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));~~
     ~~fraWindow.setVisible(true);~~
  ~~}~~
  public TempConvert() {
    setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6)); // Add the FlowLayout
    JLabel lblFahr = new JLabel("Degrees Fahrenheit");
    lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblFahr);
    add(txtFahr);
    JLabel lblCelsius = new JLabel("Degrees Celsius");
    lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblCelsius);
    add(txtCelc);
    /**  Event Handlers */
    txtCelc.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dFahr =  Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
        txtFahr.setText(String.format("%.1f", dFahr));
      }
    });
    txtFahr.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dCelc = (Double.parseDouble(txtFahr.getText()) - 32 ) * 5 / 9;
        txtCelc.setText(String.format("%.1f", dCelc));
      }
    });
  }
}
```

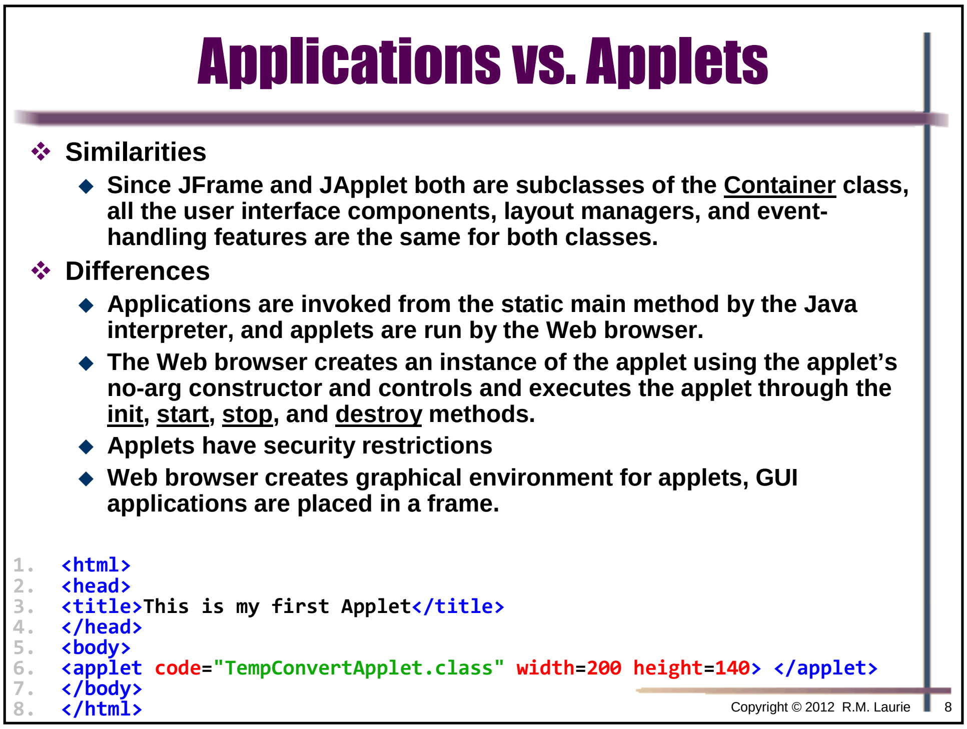
This is my first Applet

Degrees Fahrenheit

212.0

Degrees Celsius

100



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.

```
<html>
<head>
<title>This is my first Applet</title>
</head>
<body>
<applet code="TempConvertApplet.class" width=200 height=140> </applet>
</body>
</html>
```

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

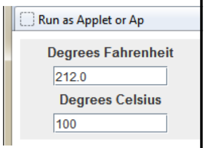
- Conversions between applications and applets are simple
- You can always convert an applet into an application, just add static main method
- You can convert an application to an applet as long as security restrictions are not violated

9

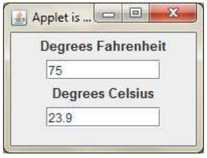
Enabling Applet to Run also as Application

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

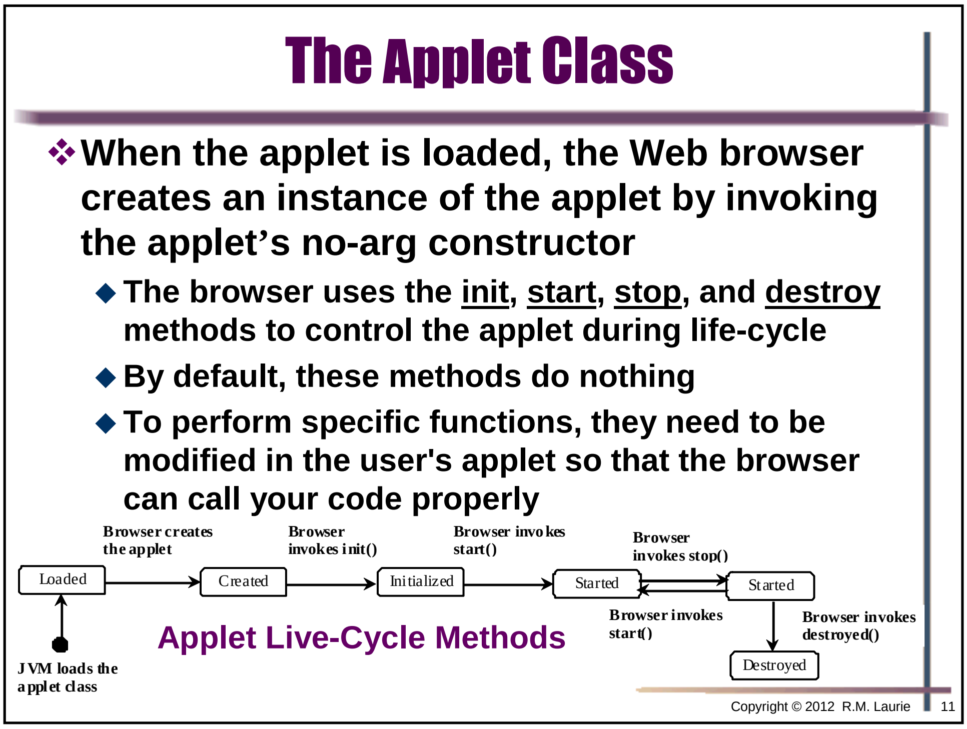
```java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class TempConvertAppletAp extends JApplet {
  private JTextField txtFahr = new JTextField("0", 10);
  private JTextField txtCelc = new JTextField("0", 10);
  public TempConvertAppletAp() {
    /** GUI components */
    setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));
    JLabel lblFahr = new JLabel("Degrees Fahrenheit");
    lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblFahr);
    add(txtFahr);
    JLabel lblCelsius = new JLabel("Degrees Celsius");
    lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblCelsius);
    add(txtCelc);
    /**  Event Handlers */
    txtCelc.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dFahr =  Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
        txtFahr.setText(String.format("%.1f", dFahr));
      }
    });
    txtFahr.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dCelc = (Double.parseDouble(txtFahr.getText()) - 32 ) * 5 / 9;
        txtCelc.setText(String.format("%.1f", dCelc));
      }
    });
  }
  public static void main(String[] args) {
    JFrame fraBack2Application = new JFrame("Applet is in frame");
    TempConvertAppletAp appletMain = new TempConvertAppletAp();
    fraBack2Application.add(appletMain);
    fraBack2Application.setSize(200, 80);
    fraBack2Application.setLocationRelativeTo(null);
    fraBack2Application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    fraBack2Application.setVisible(true);
  }
}
```



Degrees Fahrenheit
212.0
Degrees Celsius
100



Degrees Fahrenheit
75
Degrees Celsius
23.9

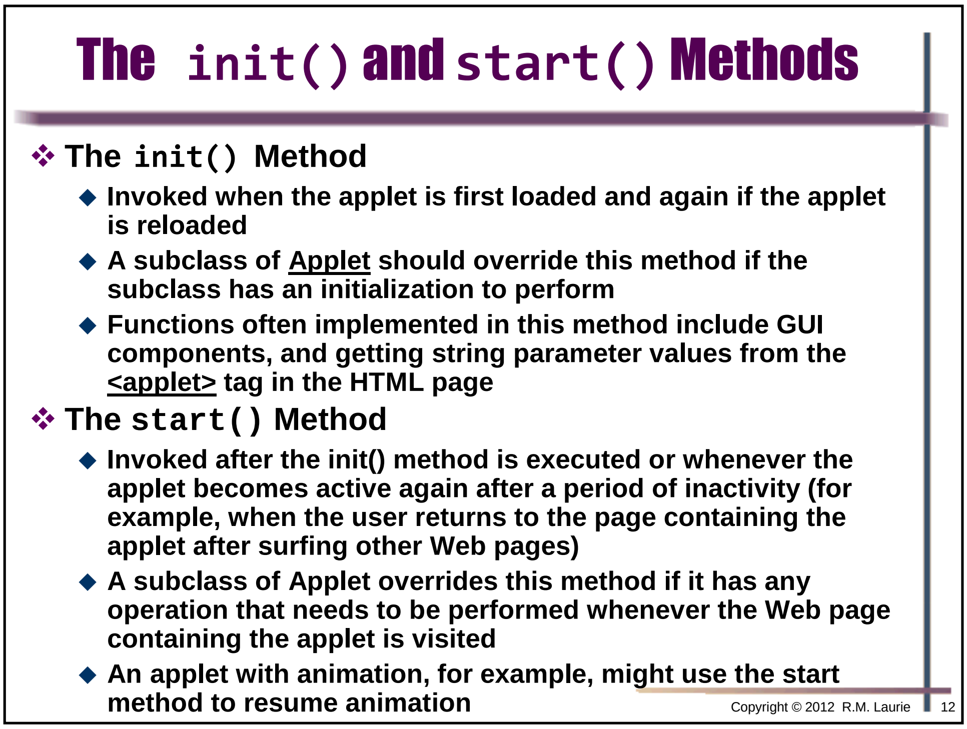


The Applet Class

- 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

Applet Live-Cycle Methods

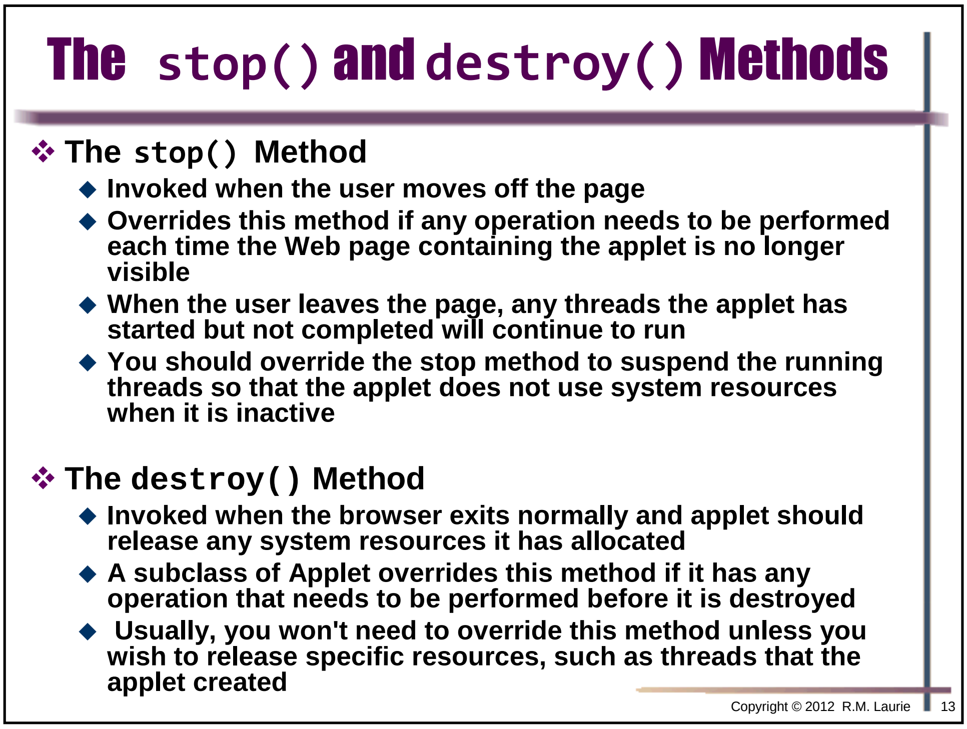
Copyright © 2012 R.M. Laurie 11



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

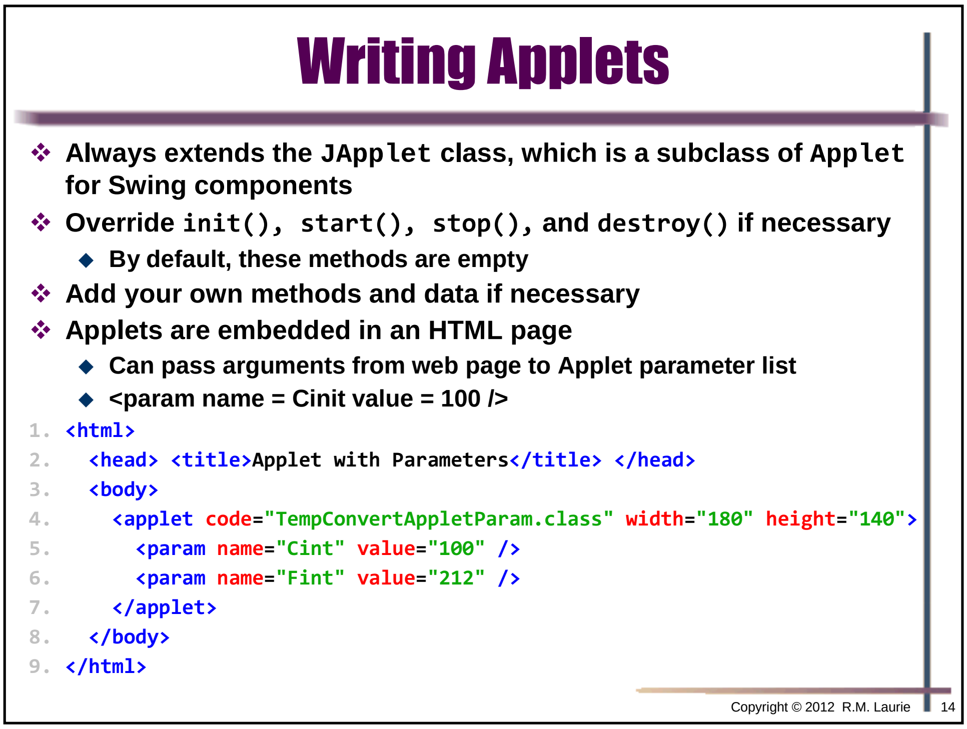
- The `init()` Method
  - Invoked when the applet is first loaded and again if the applet is reloaded
  - A subclass of Applet should override this method if the subclass has an initialization to perform
  - Functions often implemented in this method include GUI components, and getting string parameter 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

Copyright © 2012 R.M. Laurie 12



The `stop()` and `destroy()` Methods

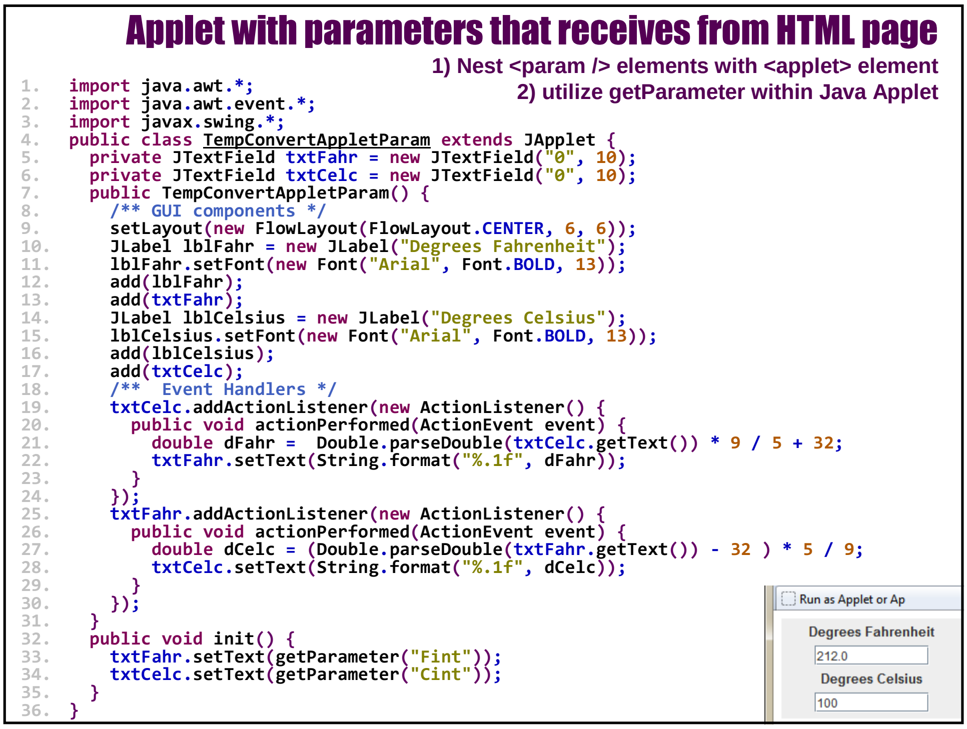
- The `stop()` Method
  - Invoked when the user moves off the page
  - Overrides this method if any operation needs to be performed each time the Web page containing the applet is no longer visible
  - When the user leaves the page, any threads the applet has started but not completed will continue to 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



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 />

```
<html>
  <head> <title>Applet with Parameters</title> </head>
  <body>
    <applet code="TempConvertAppletParam.class" width="180" height="140">
      <param name="Cint" value="100" />
      <param name="Fint" value="212" />
    </applet>
  </body>
</html>
```



Applet with parameters that receives from HTML page

1) Nest <param /> elements with <applet> element

2) utilize getParameter within Java Applet

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class TempConvertAppletParam extends JApplet {
  private JTextField txtFahr = new JTextField("0", 10);
  private JTextField txtCelc = new JTextField("0", 10);
  public TempConvertAppletParam() {
    /** GUI components */
    setLayout(new FlowLayout(FlowLayout.CENTER, 6, 6));
    JLabel lblFahr = new JLabel("Degrees Fahrenheit");
    lblFahr.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblFahr);
    add(txtFahr);
    JLabel lblCelsius = new JLabel("Degrees Celsius");
    lblCelsius.setFont(new Font("Arial", Font.BOLD, 13));
    add(lblCelsius);
    add(txtCelc);
    /**  Event Handlers */
    txtCelc.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dFahr =  Double.parseDouble(txtCelc.getText()) * 9 / 5 + 32;
        txtFahr.setText(String.format("%.1f", dFahr));
      }
    });
    txtFahr.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent event) {
        double dCelc = (Double.parseDouble(txtFahr.getText()) - 32 ) * 5 / 9;
        txtCelc.setText(String.format("%.1f", dCelc));
      }
    });
  }
  public void init() {
    txtFahr.setText(getParameter("Fint"));
    txtCelc.setText(getParameter("Cint"));
  }
}
```

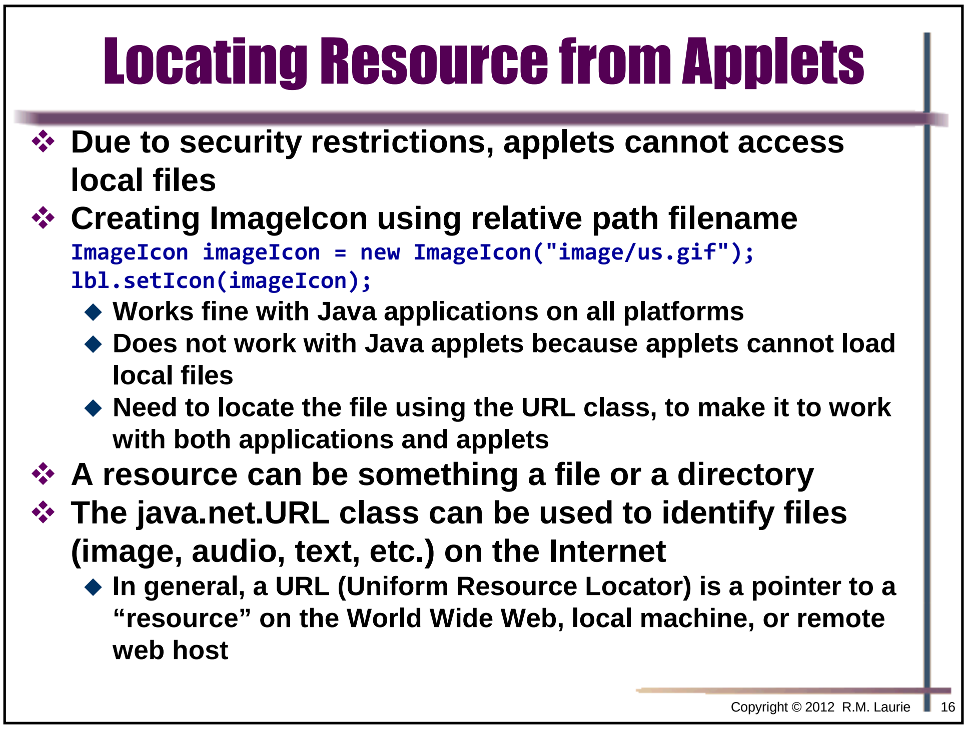
Run as Applet or Ap

Degrees Fahrenheit

212.0

Degrees Celsius

100



Locating Resource from Applets

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

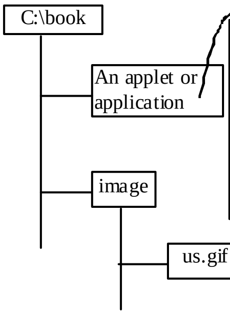
```
ImageIcon imageIcon = new ImageIcon("image/us.gif");
lbl.setIcon(imageIcon);
```

  - 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

Copyright © 2012 R.M. Laurie 16

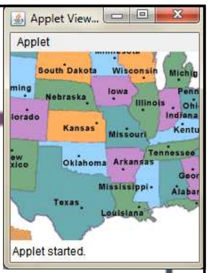
Creating a URL from a Class Reference

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

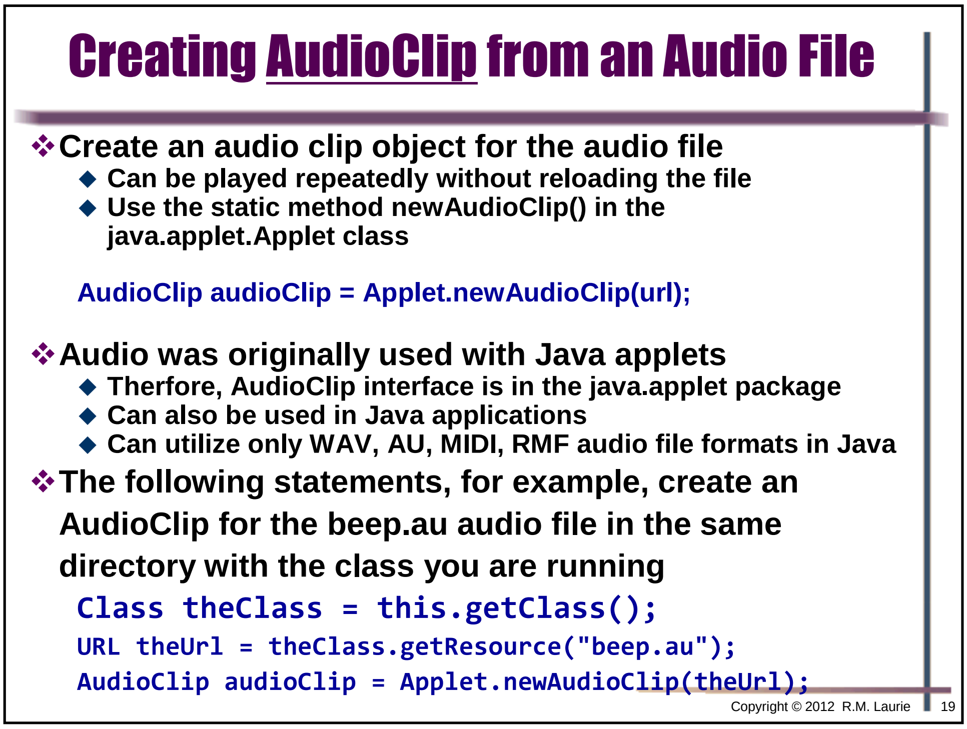


```
import java.net.URL;
.
.
Class metaObject = this.getClass();
URL url = metaObject.getResource("image/us.gif");
.
.
.
```

Displaying Image located in folder



```
import java.net.URL;
import javax.swing.*;
public class DisplayImageUsingURL extends JApplet
{
  public DisplayImageUsingURL()
  {
    // resources folder is located in bin folder with files.class
    // this.getClass() accesses current folder location of class file
    URL urlMap = this.getClass().getResource("resources/us-map.gif");
    ImageIcon imgMap = new ImageIcon(urlMap);
    JLabel lblMap = new JLabel(imgMap);
    add(lblMap);
  }
  /** Main method */
  public static void main(String[] args)
  {
    // Create a frame
    JFrame fraMain = new JFrame("Display Image Using URL");
    fraMain.setSize(500, 400);
    // Create an instance of the applet
    DisplayImageUsingURL appletMap = new DisplayImageUsingURL();
    // Add the applet instance to the frame
    fraMain.getContentPane().add(appletMap);
    fraMain.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    fraMain.setVisible(true);
  }
}
```



Creating AudioClip from an Audio File

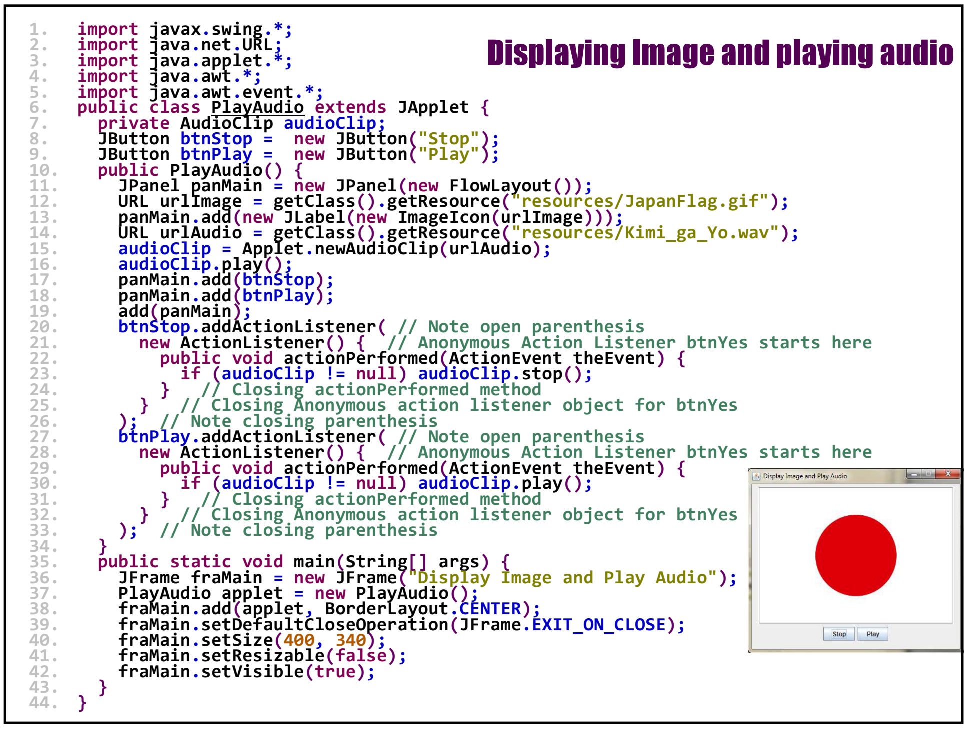
- ❖ Create an audio clip object for the audio file
  - ◆ Can be played repeatedly without reloading the file
  - ◆ Use the static method newAudioClip() in the java.applet.Applet class

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

- ❖ Audio was originally used with Java applets
  - ◆ Therfore, 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

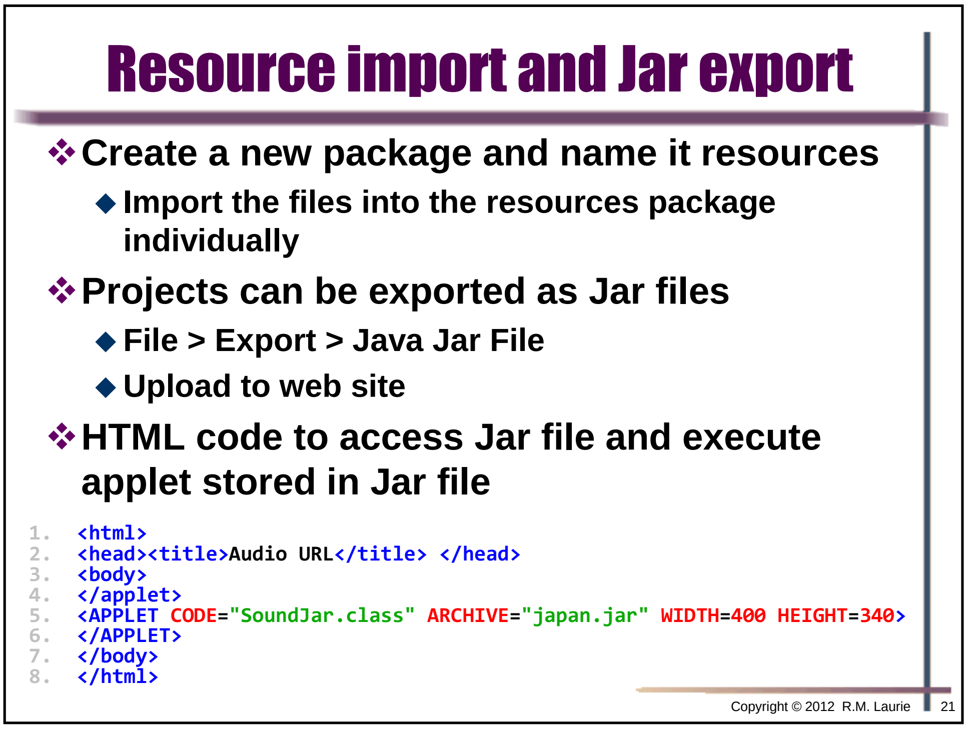
```
Class theClass = this.getClass();
URL theUrl = theClass.getResource("beep.au");
AudioClip audioClip = Applet.newAudioClip(theUrl);
```

19



Displaying Image and playing audio

```
import javax.swing.*;
import java.net.URL;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class PlayAudio extends JApplet {
  private AudioClip audioClip;
  JButton btnStop =  new JButton("Stop");
  JButton btnPlay =  new JButton("Play");
  public PlayAudio() {
    JPanel panMain = new JPanel(new FlowLayout());
    URL urlImage = getClass().getResource("resources/JapanFlag.gif");
    panMain.add(new JLabel(new ImageIcon(urlImage)));
    URL urlAudio = getClass().getResource("resources/Kimi_ga_Yo.wav");
    audioClip = Applet.newAudioClip(urlAudio);
    audioClip.play();
    panMain.add(btnStop);
    panMain.add(btnPlay);
    add(panMain);
    btnStop.addActionListener( // Note open parenthesis
      new ActionListener() {  // Anonymous Action Listener btnYes starts here
        public void actionPerformed(ActionEvent theEvent) {
          if (audioClip != null) audioClip.stop();
        }   // Closing actionPerformed method
      }   // Closing Anonymous action listener object for btnYes
    );  // Note closing parenthesis
    btnPlay.addActionListener( // Note open parenthesis
      new ActionListener() {  // Anonymous Action Listener btnYes starts here
        public void actionPerformed(ActionEvent theEvent) {
          if (audioClip != null) audioClip.play();
        }   // Closing actionPerformed method
      }   // Closing Anonymous action listener object for btnYes
    );  // Note closing parenthesis
  }
  public static void main(String[] args) {
    JFrame fraMain = new JFrame("Display Image and Play Audio");
    PlayAudio applet = new PlayAudio();
    fraMain.add(applet, BorderLayout.CENTER);
    fraMain.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    fraMain.setSize(400, 340);
    fraMain.setResizable(false);
    fraMain.setVisible(true);
  }
}
```

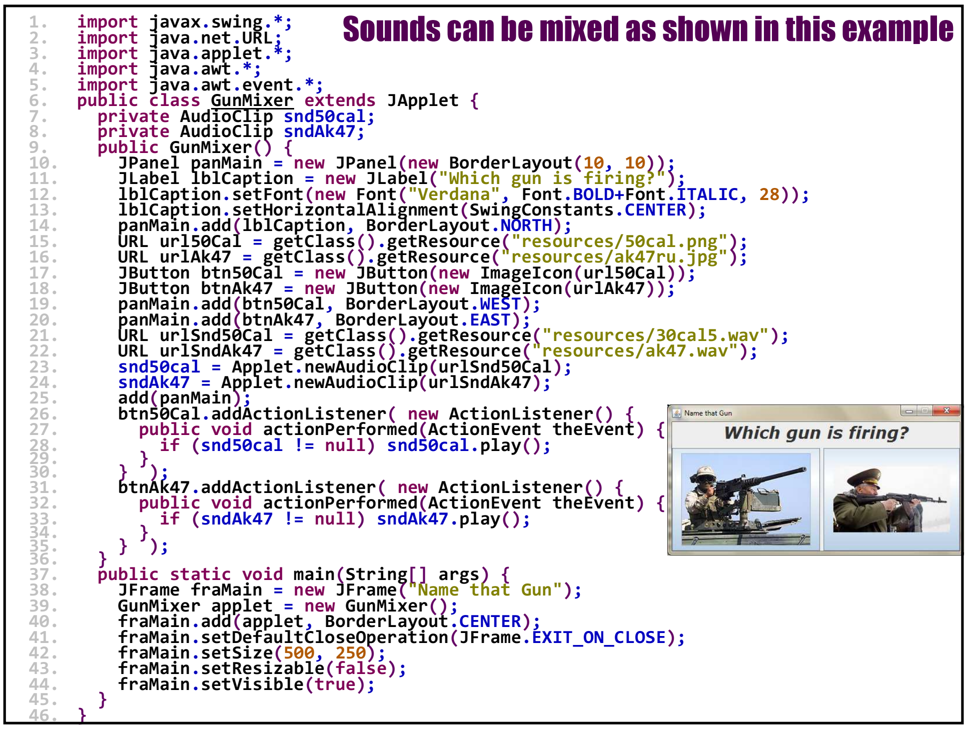


Resource import and Jar export

- 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

```html
<html>
<head><title>Audio URL</title> </head>
<body>
</applet>
<APPLET CODE="SoundJar.class" ARCHIVE="japan.jar" WIDTH=400 HEIGHT=340>
</APPLET>
</body>
</html>
```

Copyright © 2012 R.M. Laurie 21



Sounds can be mixed as shown in this example

```java
import javax.swing.*;
import java.net.URL;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class GunMixer extends JApplet {
  private AudioClip snd50cal;
  private AudioClip sndAk47;
  public GunMixer() {
    JPanel panMain = new JPanel(new BorderLayout(10, 10));
    JLabel lblCaption = new JLabel("Which gun is firing?");
    lblCaption.setFont(new Font("Verdana", Font.BOLD+Font.ITALIC, 28));
    lblCaption.setHorizontalAlignment(SwingConstants.CENTER);
    panMain.add(lblCaption, BorderLayout.NORTH);
    URL url50Cal = getClass().getResource("resources/50cal.png");
    URL urlAk47 = getClass().getResource("resources/ak47ru.jpg");
    JButton btn50Cal = new JButton(new ImageIcon(url50Cal));
    JButton btnAk47 = new JButton(new ImageIcon(urlAk47));
    panMain.add(btn50Cal, BorderLayout.WEST);
    panMain.add(btnAk47, BorderLayout.EAST);
    URL urlSnd50Cal = getClass().getResource("resources/30cal5.wav");
    URL urlSndAk47 = getClass().getResource("resources/ak47.wav");
    snd50cal = Applet.newAudioClip(urlSnd50Cal);
    sndAk47 = Applet.newAudioClip(urlSndAk47);
    add(panMain);
    btn50Cal.addActionListener( new ActionListener() {
      public void actionPerformed(ActionEvent theEvent) {
        if (snd50cal != null) snd50cal.play();
      }
    } );
    btnAk47.addActionListener( new ActionListener() {
      public void actionPerformed(ActionEvent theEvent) {
        if (sndAk47 != null) sndAk47.play();
      }
    } );
  }
  public static void main(String[] args) {
    JFrame fraMain = new JFrame("Name that Gun");
    GunMixer applet = new GunMixer();
    fraMain.add(applet, BorderLayout.CENTER);
    fraMain.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    fraMain.setSize(500, 250);
    fraMain.setResizable(false);
    fraMain.setVisible(true);
  }
}
```