20

Introduction to Java Applets

Observe due measure, for right timing is in all things the most important factor.

> **— Georg Wilhelm Friedrich Hegel**

Painting is only a bridge linking the painter’s mind with that of the viewer.

> **— Eugene Delacroix**

The direction in which education starts a man will determine his future in life.

> **— Plato**

◀ ▶

OBJECTIVES

In this chapter you will learn:

- To differentiate between applets and applications.
- To observe some of Java's exciting capabilities through the JDK's demonstration applets.
- To write simple applets.
- To write a simple HyperText Markup Language (HTML) document to load an applet into an applet container and execute the applet.
- Five methods that are called automatically by an applet container during an applet's life cycle.

Outline

20.1 Introduction

20.2 Sample Applets Provided with the JDK

20.3 Simple Java Applet: Drawing a String

20.3.1 Executing an Applet in the `appletviewer`

20.3.2 Executing an Applet in a Web Browser

20.4 Applet Life-Cycle Methods

20.5 Initializing an Instance Variable with Method `init`

20.6 Sandbox Security Model

20.7 Internet and Web Resources

20.8 Wrap-Up

20.1 Introduction

- **Applets**
  - **Java programs that can be embedded in HyperText Markup Language (HTML) documents**
  - **The browser that executes an applet is generically known as the applet container**

20.2 Sample Applets Provided with the JDK

- **Demonstration applets provided with the JDK**
  - **Demonstration programs are located in directory demo**
    - **Default location in Windows:**
      C:\Program Files\Java\jdk1.5.0\demo
    - **Default location in UNIX/Linux/Mac OS X:**
      the directory in which you install the JDK followed by
      jdk1.5.0/demo
  - **JDK and the demos can be downloaded from the Sun Microsystems Java Web site**
    - **java.sun.com/j2se/5.0/**

Example	Description
`Animator`	Performs one of four separate animations.
`ArcTest`	Demonstrates drawing arcs. You can interact with the applet to change attributes of the arc that is displayed.
`BarChart`	Draws a simple bar chart.
`Blink`	Displays blinking text in different colors.
`CardTest`	Demonstrates several GUI components and layouts.
`Clock`	Draws a clock with rotating hands, the current date and the current time. The clock updates once per second.
`DitherTest`	Demonstrates drawing with a graphics technique known as dithering that allows gradual transformation from one color to another.
`DrawTest`	Allows the user mouse to draw lines and points in different colors by dragging the mouse.
`Fractal`	Draws a fractal. Fractals typically require complex calculations to determine how they are displayed.
`GraphicsTest`	Draws shapes to illustrate graphics capabilities.

Fig. 20.1 | The examples from the `applets` directory. (Part 1 of 3.)

Example	Description
`GraphLayout`	Draws a graph consisting of many nodes (represented as rectangles) connected by lines. Drag a node to see the other nodes in the graph adjust on the screen and demonstrate complex graphical interactions.
`ImageMap`	Demonstrates an image with hot spots. Positioning the mouse pointer over certain areas of the image highlights the area and displays a message in the lower-left corner of the applet container window. Position over the mouth in the image to hear the applet say "hi."
`JumpingBox`	Moves a rectangle randomly around the screen. Try to catch it by clicking it with the mouse!

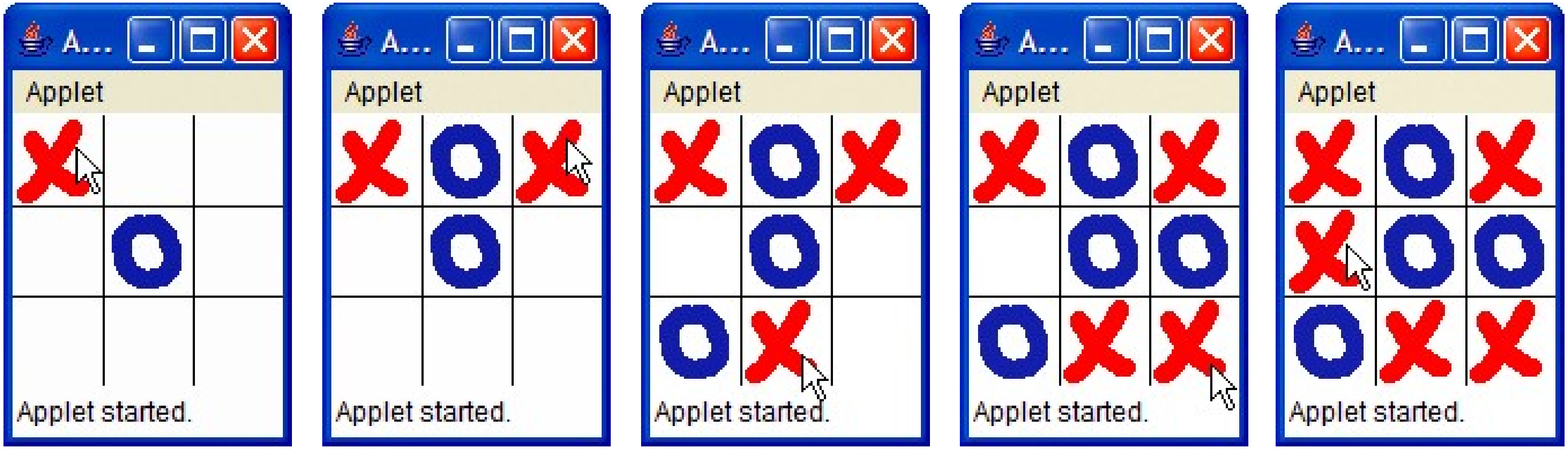
Fig. 20.1 | The examples from the `applets` directory. (Part 2 of 3.)

Example	Description
`MoleculeViewer`	Presents a three-dimensional view of several chemical molecules. Drag the mouse to view the molecule from different angles.
`NervousText`	Draws text that jumps around the applet.
`SimpleGraph`	Draws a complex curve.
`SortDemo`	Compares three sorting techniques. Sorting (described in Chapter 16) arranges information in order—like alphabetizing words. When you execute this example from a command window, three `appletviewer` windows appear. When you execute this example in a browser, the three demos appear side-by-side. Click in each demo to start the sort. Note that the sorts all operate at different speeds.
`SpreadSheet`	Demonstrates a simple spreadsheet of rows and columns.
`TicTacToe`	Allows the user to play Tic-Tac-Toe against the computer.
`WireFrame`	Draws a three-dimensional shape as a wire frame. Drag the mouse to view the shape from different angles.

Fig. 20.1 | The examples from the `applets` directory. (Part 3 of 3.)

20.2 Sample Applets Provided with the JDK (Cont.)

- `TicTacToe` applet
  - Allows you to play Tic-Tac-Toe against the computer
  - Run the applet with the `appletviewer` command
    - Change directories to subdirectory `TicTacToe`
    - Type command `appletviewer example1.html`
    - Point the mouse at the square where you want to place an **X**
  - To play again
    - Click the **Applet** menu
    - Select the **Reload** menu item
  - To terminate the `appletviewer`
    - Click the **Applet** menu
    - Select the **Quit** menu item



Applet started.

Applet started.

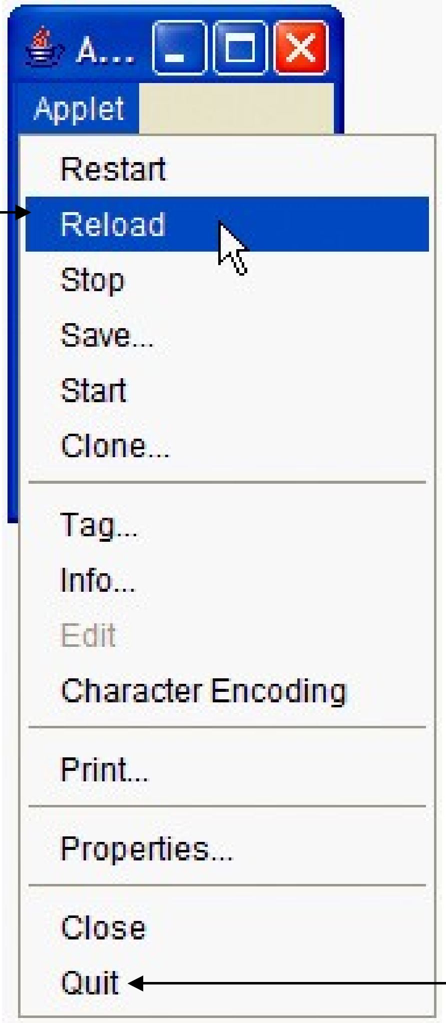
Applet started.

Applet started.

Applet started.

Fig. 20.2 | **`TicTacToe` applet sample execution.**

Reload the applet to execute it again.



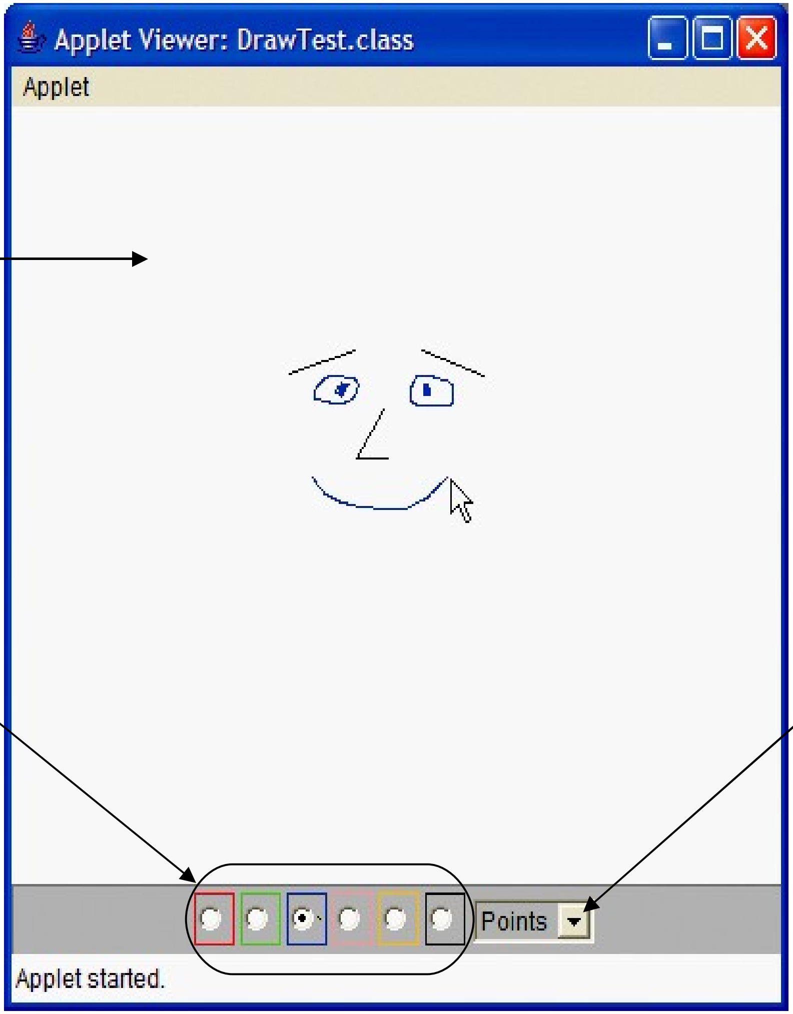
Select **Quit** to terminate the `appletviewer`.

Fig. 20.3 | **Applet menu in the `appletviewer`.**

20.2 Sample Applets Provided with the JDK (Cont.)

- `DrawTest` applet
  - Allows you to draw lines and points in different colors
  - Run the applet with the `appletviewer` command
    - Change directories to subdirectory `drawTest`
    - Type command `appletviewer example1.html`
    - Drag the mouse across the applet to draw lines
    - Select a color by clicking one of the radio buttons at the bottom of the applet
      - Select from red, green, blue, pink, orange and black
    - Change the shape to draw from **Lines** to **Points** by selecting **Points** from the combo box
    - Select **Reload** from the **Applet** menu to start a new drawing

Drag the mouse in the white area to draw.



Select the drawing color by clicking one of the radio buttons.

Select **Lines** or **Points** from the combo box to specify what will be drawn when you drag the mouse.

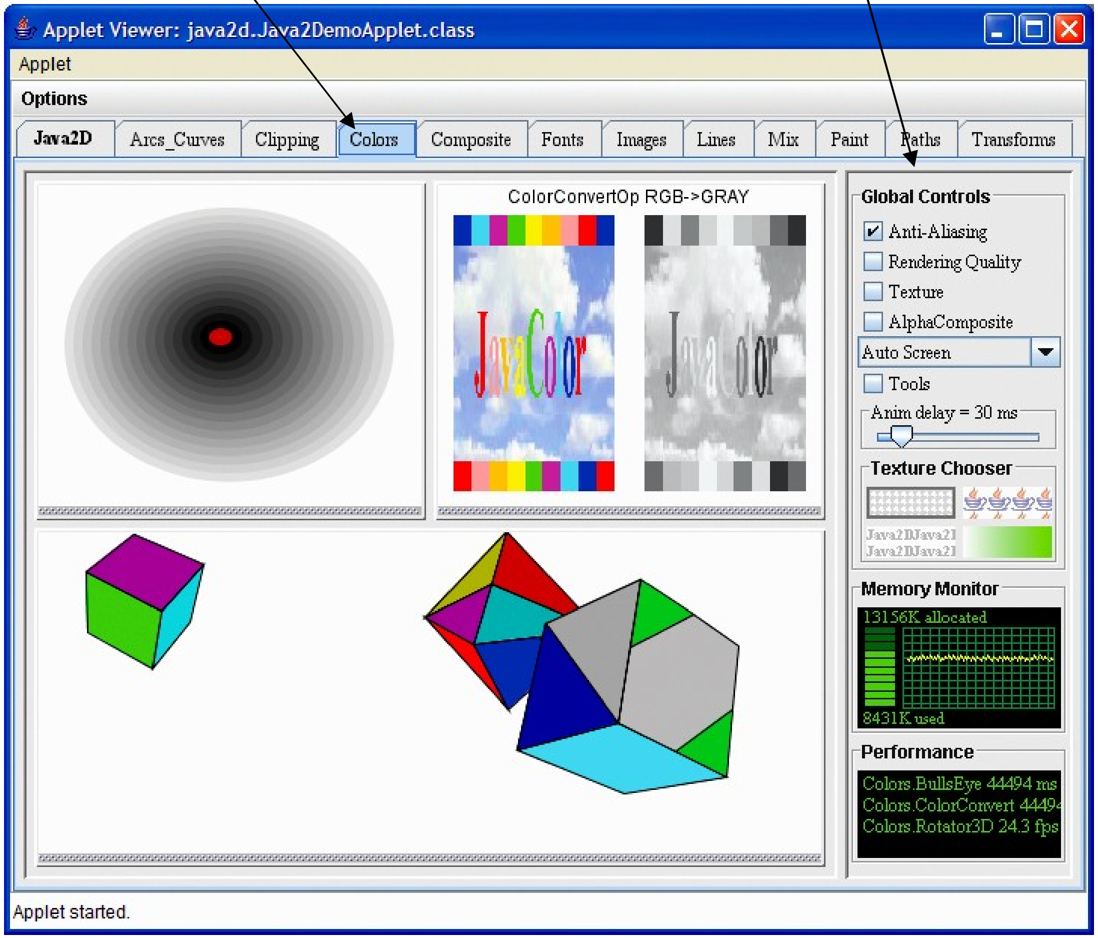
Fig. 20.4 | **`DrawTest` applet sample execution.**

20.2 Sample Applets Provided with the JDK (Cont.)

- `Java2D` applet
  - Demonstrates many features of the Java 2D API
  - Run the applet with the `appletviewer` command
    - Change directories to the `jfc` directory in the JDK's `demo` directory, then change to the `Java2D` directory
    - Type command `appletviewer Java2Demo.html`
    - To change to a different part of the demo, click a different tab at the top of the applet
    - Change the options in the upper-right corner
      - Example: click the checkbox to the left of the word **Anti-Aliasing**
        - A graphical technique for producing smoother graphics in which edges of the graphic are blurred

Click a tab to select a two-dimensional graphics demo.

Try changing the options to see their effect on the demonstration.



Global Controls

- ☑ Anti-Aliasing
- □ Rendering Quality
- □ Texture
- □ AlphaComposite
- Auto Screen
- □ Tools
- Anim delay = 30 ms

Memory Monitor

```
13156K allocated
8431K used
```

Performance

```
Colors.BullsEye 44494 ms
Colors.ColorConvert 4449
Colors.Rotator3D 24.3 fps
```

Applet started.

Fig. 20.5 | Java2D applet sample execution.

20.3 Simple Java Applet: Drawing a String

- **Creating the applet class**
  - **An applet container can create only objects of classes that are `public` and extend `JApplet`**
  - **An applet container expects every Java applet class to have methods named `init`, `start`, `paint`, `stop` and `destroy`**
    - **These methods are inherited from class `JApplet` and can be overridden**
    - **When an applet container loads an applet class, the container creates an object of the class then calls methods `init`, `start` and `paint`**

```java
// Fig. 20.6: WelcomeApplet.java
// A first applet in Java.
import java.awt.Graphics;   // program uses class Graphics
import javax.swing.JApplet; // program uses class JApplet

public class WelcomeApplet extends JApplet
{
   // draw text on applet’s background
   public void paint( Graphics g )
   {
      // call superclass version of method paint
      super.paint( g );

      // draw a String at x-coordinate 25 and y-coordinate 25
      g.drawString( "Welcome to Java Programming!", 25, 25 );
   } // end method paint
} // end class WelcomeApplet
```

Import **`Graphics`** and **`JApplet`**

`WelcomeApplet.java`

Class **`WelcomeApplet`** extends class **`JApplet`**

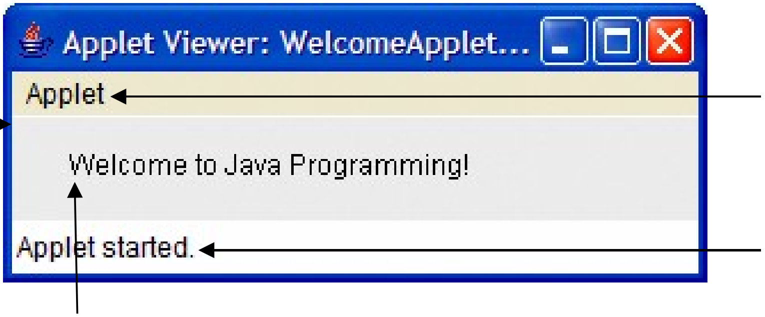
Call the superclass version of method **`paint`**

Use **`Graphics`** method **`drawString`** to draw **`Welcome to Java Programming!`**

`WelcomeApplet` executing in the **`appletviewer`**

x-axis

y-axis



```
Welcome to Java Programming!
```

```
Applet started.
```

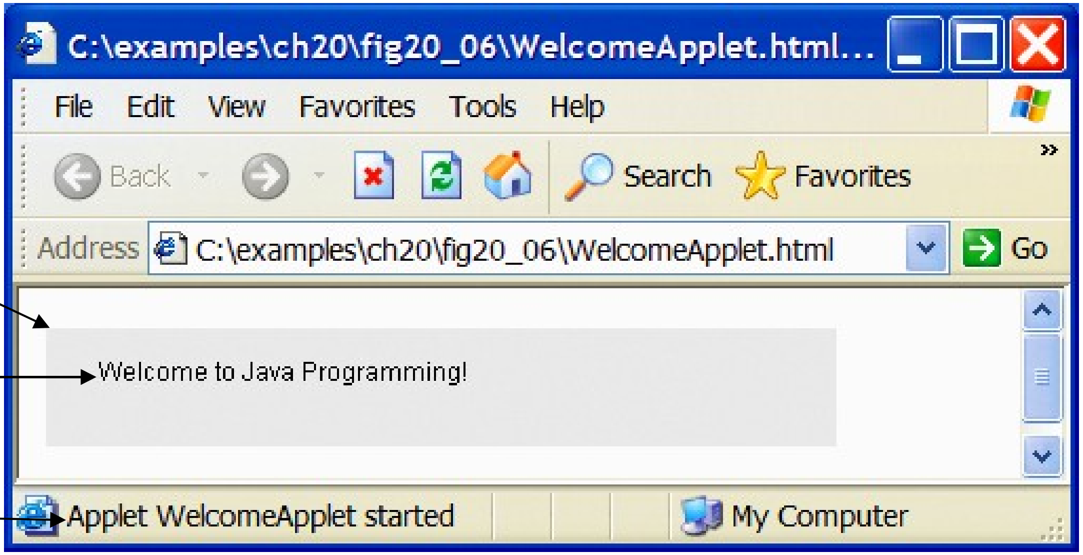
Upper-left corner of drawing area is location *(0, 0)*. Drawing area extends from below the **`Applet`** menu to above the status bar. *x*-coordinates increase from left to right. *y*-coordinates increase from top to bottom.

`Applet` menu

Status bar mimics what would be displayed in the browser's status bar as the applet loads and begins executing.

Pixel coordinates *(25, 25)* at which the string is displayed

`WelcomeApplet` executing in Microsoft Internet Explorer


C:\examples\ch20\fig20_06\WelcomeApplet.html...
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
C:\examples\ch20\fig20_06\WelcomeApplet.html
Go
My Computer

```
Welcome to Java Programming!
```

```
Applet WelcomeApplet started
```

Upper-left corner of drawing area

Pixel coordinate *(25, 25)*

Status bar

Fig. 20.7 | **Sample outputs of the `WelcomeApplet` in Fig. 20.6.**

20.3 **Simple Java Applet: Drawing a String (Cont.)**

- **Overriding method `paint` for drawing**
  - **The applet container calls method `paint` with a `Graphics` object as an argument to tell the applet when to draw**

20.3.1 Executing an Applet in the `appletviewer`

- Applets are embedded in Web pages for execution in an applet container
  - Before executing the applet, you must create an HTML document that specifies which applet to execute
    - HTML documents typically end with an ".html" or ".htm" file-name extension
    - Most HTML elements are delimited by pairs of tags
      - All HTML tags begin with a left angle bracket, `<`, and end with a right angle bracket, `>`
  - Execute `WelcomeApplet` in the `appletviewer`
    - In the directory containing your applet and HTML document, type `appletviewer WelcomeApplet.html`
    - The `appletviewer` understands only the `<applet>` and `</applet>` HTML tags and ignores all other tags

```
<html>
<applet code = "WelcomeApplet.class" width = "300" height = "45">
</applet>
</html>
```

Specify an applet element

Applet element attributes

Fig. 20.8 | `WelcomeApplet.html` loads `WelcomeApplet` (Fig. 20.6) into an applet container.

Look-and-Feel Observation 20.1

To ensure that it can be viewed properly on most computer screens, an applet should generally be less than 1024 pixels wide and 768 pixels tall—dimensions supported by most computer screens.

Common Programming Error 20.1

Forgetting the ending `</applet>` tag prevents the applet from executing in some applet containers. The `appletviewer` terminates without indicating an error. Some Web browsers simply ignore the incomplete `applet` element.

Error-Prevention Tip 20.1

If you receive a `MissingResourceException` error message when loading an applet into the `appletviewer` or a browser, check the `<applet>` tag in the HTML document carefully for syntax errors, such as commas (`,`) between the attributes.

Error-Prevention Tip 20.2

Test your applets in the `appletviewer` applet container before executing them in a Web browser. Browsers often save a copy of an applet in memory until all the browser's windows are closed. If you change an applet, recompile it, then reload it in your browser, the browser may still execute the original version of the applet. Close all your browser windows to remove the old applet from memory. Open a new browser window and load the applet to see your changes.

Error-Prevention Tip 20.3

Test your applets in every Web browser in which they will execute to ensure that they operate correctly.

20.3.2 Executing an Applet in a Web Browser

- **To execute an applet in Internet Explorer:**
  - **Select Open… from the `File` menu**
  - **Click the Browse… button**
  - **Locate the directory containing the HTML document for the applet you wish to execute**
  - **Select the HTML document**
  - **Click the Open button**
  - **Click the OK button**

20.3.2 Executing an Applet in a Web Browser (Cont.)

- If your applet executes in the `appletviewer` but not in your Web browser
  - Java may not be installed and configured for your browser
    - Visit the Web site `java.com` and click the **Get It Now** button to install Java for your browser
    - You may need to manually configure Internet Explorer to use J2SE 5.0
      - Click the **Tools** menu
      - Select **Internet Options…**
      - Click the **Advanced** tab
      - Check the "**Use JRE v1.5.0 for <applet> (requires restart)**" option
      - Click **OK**
      - Close all browser windows before attempting to execute another applet in the browser

Method	When the method is called and its purpose
`public void init()`	Called once by the applet container when an applet is loaded for execution. This method initializes an applet. Typical actions performed here are initializing fields, creating GUI components, loading sounds to play, loading images to display (see Chapter 20, Multimedia: Applets and Applications) and creating threads (see Chapter 23, Multithreading).
`public void start()`	Called by the applet container after method `init` completes execution. In addition, if the user browses to another Web site and later returns to the applet's HTML page, method `start` is called again. The method performs any tasks that must be completed when the applet is loaded for the first time and that must be performed every time the applet's HTML page is revisited. Actions performed here might include starting an animation (see Chapter 21) or starting other threads of execution (see Chapter 23).

Fig. 20.9 | `JApplet` life cycle methods that are called by an applet container during an applet's execution. (Part 1 of 3.)

Method	When the method is called and its purpose
`public void paint( Graphics g )`	Called by the applet container after methods `init` and `start`. Method `paint` is also called when the applet needs to be repainted. For example, if the user covers the applet with another open window on the screen and later uncovers the applet, the `paint` method is called. Typical actions performed here involve drawing with the `Graphics` object `g` that is passed to the `paint` method by the applet container.
`public void stop()`	This method is called by the applet container when the user leaves the applet's Web page by browsing to another Web page. Since it is possible that the user might return to the Web page containing the applet, method `stop` performs tasks that might be required to suspend the applet's execution, so that the applet does not use computer processing time when it is not displayed on the screen. Typical actions performed here would stop the execution of animations and threads.

Fig. 20.9 | **`JApplet` life cycle methods that are called by an applet container during an applet's execution. (Part 2 of 3.)**

Method	When the method is called and its purpose
`public void destroy()`	
	This method is called by the applet container when the applet is being removed from memory. This occurs when the user exits the browsing session by closing all the browser windows and may also occur at the browser's discretion when the user has browsed to other Web pages. The method performs any tasks that are required to clean up resources allocated to the applet.

Fig. 20.9 | `JApplet` life cycle methods that are called by an applet container during an applet's execution. (Part 3 of 3.)

Common Programming Error 20.2

Declaring methods `init`, `start`, `paint`, `stop` or `destroy` with method headers that differ from those shown in Figure 20.9 results in methods that will not be called by the applet container. The code specified in your versions of the methods will not execute.

20.5 Initializing an Instance Variable with Method `init`

- Applet **`AdditionApplet`**
  - computes the sum of two values input by the user and displays the result by drawing a **`String`** inside a rectangle on the applet
    - The sum is stored in an instance variable of class **`AdditionApplet`**
      - So it can be used in both method **`init`** and method **`paint`**

```
// Fig. 20.10: AdditionApplet.java
// Adding two floating-point numbers.
import java.awt.Graphics;       // program uses class Graphics
import javax.swing.JApplet;     // program uses class JApplet
import javax.swing.JOptionPane; // program uses class JOptionPane

public class AdditionApplet extends JApplet
{
   private double sum; // sum of values entered by user

   // initialize applet by obtaining values from user
   public void init()
   {
      String firstNumber;  // first string entered by user
      String secondNumber; // second string entered by user

      double number1; // first number to add
      double number2; // second number to add

      // obtain first number from user
      firstNumber = JOptionPane.showInputDialog(
         "Enter first floating-point value" );

      // obtain second number from user
      secondNumber = JOptionPane.showInputDialog(
         "Enter second floating-point value" );

```

Outline

AdditionApplet .java

(1 of 3)

```
      // convert numbers from type String to type double
      number1 = Double.parseDouble( firstNumber );
      number2 = Double.parseDouble( secondNumber );

      sum = number1 + number2; // add numbers
   } // end method init

   // draw results in a rectangle on applet’s background
   public void paint( Graphics g )
   {
      super.paint( g ); // call superclass version of method paint

      // draw rectangle starting from (15, 10) that is 270
      // pixels wide and 20 pixels tall
      g.drawRect( 15, 10, 270, 20 );

      // draw results as a String at (25, 25)
      g.drawString( "The sum is " + sum, 25, 25 );
   } // end method paint
} // end class AdditionApplet
```

Outline

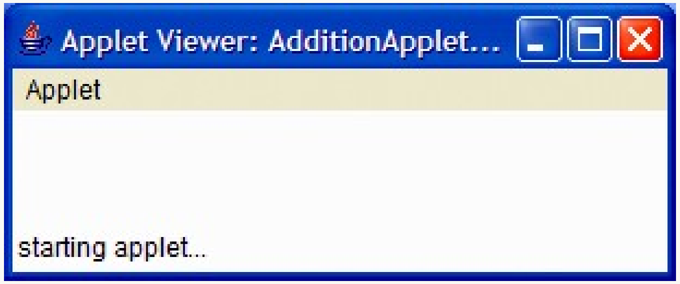
AdditionApplet.java

(2 of 3)

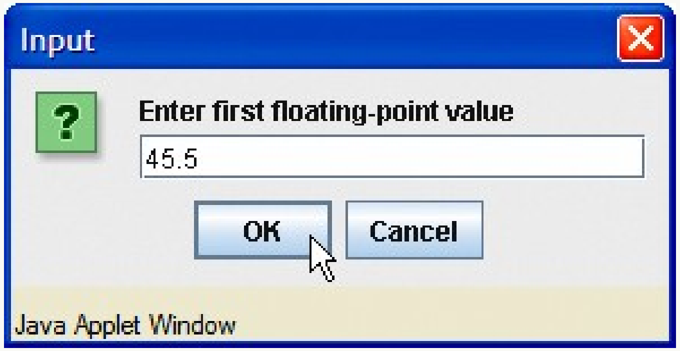
Outline

`AdditionApplet.java`

(3 of 3)

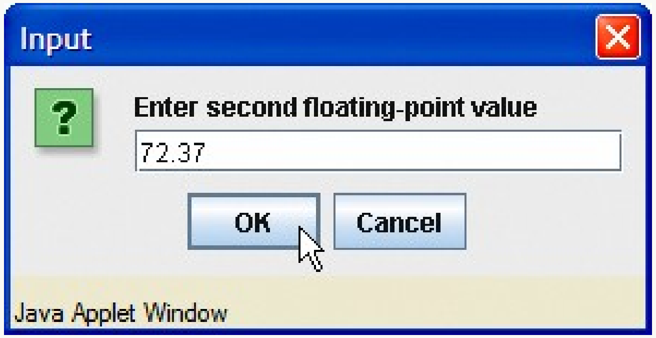


starting applet...



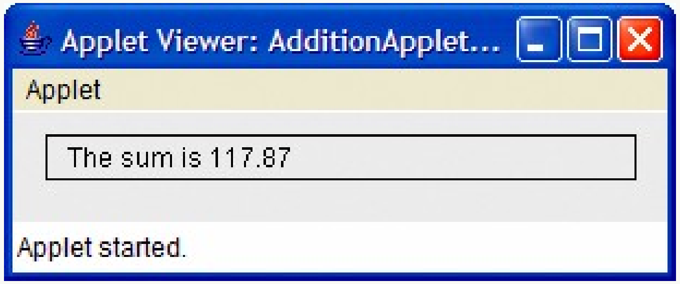
Enter first floating-point value

45.5



Enter second floating-point value

72.37



The sum is 117.87

Applet started.

```
<html>
<applet code = "AdditionApplet.class" width = "300" height = "65">
</applet>
</html>
```

Load **`AdditionApplet`**

Fig. 20.11 | **`AdditionApplet.html`** **loads class** **`AdditionApplet`** **of Fig. 20.10 into an applet container.**

Software Engineering Observation 20.1

The only statements that should be placed in an applet's `init` method are those that should execute only once when the applet is initialized.

20.6 Sandbox Security Model

- Sandbox security model
  - Code executing in the “sandbox” is not allowed to “play outside the sandbox”
  - Used by the Java platform to prevent code that is downloaded to your local computer from accessing local system resources, such as files
  - For information on security and applets
    - developer.java.sun.com/developer/technicalAr
  - For information on the Java 2 Platform security model
    - java.sun.com/j2se/5.0/docs/guide/security/sp

20.7 Internet and Web Resources

- **Sun Microsystems Java Web site**
  - `java.sun.com`
  - Java applet resources
    - `java.sun.com/applets`
  - Free online tutorials
    - `java.sun.com/learning`
- **To install and configure Java for your browser**
  - Visit `java.com`
  - Click the **Get It Now** button

20.7 Internet and Web Resources

- **JARS** `www.jars.com`
  - **Originally called the Java Applet Rating Service**
    - **Rated every applet registered at the site**
    - **Allowed users to view the best applets on the Web**
  - **Now an all-around resource for Java programmers**