Chapter 5 Swing Components

1. AWT components are heavily dependent on their underlying platform. For each platform on which Java runs, the AWT components are automatically mapped to the platform-specific components through their respective agents, known as peers. Swing user interface components are painted directly on canvases using Java code, except for components that are subclasses of java.awt.Window or java.awt.Panel, which must be drawn using native GUI on a specific platform. Swing components are less dependent on the target platform and use fewer native GUI resources. For this reason, Swing components are referred to as lightweight components, and AWT components are referred to as heavyweight components.

2. A Swing user interface component usually consists of the following objects:

   - A component, such as JButton, which extends JComponent.
   - A UI delegate, such as BasicButtonUI, which is responsible for displaying the component.
   - A model, such as ButtonModel, which maintains a component's data.

   Swing components delegate visual presentation to their respective UI delegates. A UI delegate plays the role of the view in the classic model-view architecture. Programmers use Swing components to create user interfaces. Components indirectly create their UI delegates. The UI delegates work behind the scenes to paint the components whenever the models change. The UI delegates can be plugged into a component when the components are constructed. The major benefit of delegating painting to the UI delegates is for implementing pluggable look-and-feel, which is introduced in the next section.

3. Use the static method setLookAndFeel in the UIManage class. For example,

```
UIManager.setLookAndFeel
  (UIManager.getCrossPlatformLookAndFeelClassName());
```

4. No.

5. JComponent is a subclass of java.awt.Container. java.awt.Container is a subclass of Component. JComponent inherits properties from Container and Component. Additionally, it defines the properties border, autoscrolls, doubleBuffered, etc.

6. AbstractButton is the superclass for JButton, JCheckBox, JradionButton, and JToggleButton. The AbstractButton class contains the properties text, actionCommand, mnemonic, icon, pressedIcon, rolloverIcon, disabledIcon, horizontalAlignment, verticalAlignment, horizontalTextAlignment, verticalTextAlignment, selected, selectedIcon, etc.

7. Place an object of JTextArea to JScrollPane.

8. Create an instance of URL for the HTML file, and set it with an instance of JEditorPane using the setPage method.

9. JScrollPane supports scrolling automatically whereas you have implement scrolling in JScrollBar.

10. JSlider and JScrollBar both can be used as a control for scrolling. JSlider has more features than JScrollBar.

11. Swing components are not thread-safe, meaning that they can only be accessed from the event dispatch thread once they are displayed. The event dispatch thread is the thread that invokes callback methods like update and paint as well as the event handlers in the event listener interface. Occasionally, you need to update GUI components from a nonstandard event, as in this case, where the progress bar is repainted in the thread that copies files. To ensure that the progress bar is updated in the event dispatch thread, you need to use the SwingUtilities's invokeLater method to invoke the thread for updating the progress bar.

12. The value property.

13. No. Buttons cannot be shares.