Displaying a Swing component

Construct and initialize the component.

```java
button = new JButton("ButtonLabel");
```

Add it to the content pane of the window or to a JPanel that is added to the display.

```java
getContentPane().add(button);
```

Import `javax.swing.`, `and sometimes also java.awt.` at the beginning of the class creating the components.

```java
import javax.swing.; import java.awt.
```

Getting events from GUI component

Declare that the class handling the event implements the appropriate listener interface.

```java
implements ActionListener
```

Define the method that the listener interface requires.

```java
public void actionPerformed (ActionEvent event)
```

Add a listener appropriate for the component to the component.

```java
button.addActionListener (this);
```

Import `java.awt.event.`, `and occasionally java.swing.event.` at the beginning of the class that is the listener.

```java
import java.awt.; import java.awt.event.
```

Finding out which component sent the event

When the listener method is called, you can find out which component sent the event by calling `getsource()` on the event:

```java
Object theButton = event.getSource();
if (theButton == framedCircleButton) {
    // Create a framed circle
}
```

If a method returns a String, remember to compare the result using the equals method, not ==:

```java
aMenu.getSelectedItem().equals("A value");
```

Containers

**JPanel constructor:** new JPanel()

**Define the type of layout:** void setLayout (LayoutManager lm)

**Add an object to a container:** void add (Component c)

**Add an object to a container:** void add (Component c, int position)

Both JPanel and the object obtained by sending `getContentPane()` to a WindowController object are containers (and have type Container). These methods are available for all containers.

For `BorderLayout`, position may be either `BorderLayout.NORTH`, `BorderLayout.SOUTH`, `BorderLayout.EAST`, `BorderLayout.WEST`, or `BorderLayout.CENTER`.

Layout Managers

**BorderLayout constructor:** new BorderLayout()

**FlowLayout constructor:** new FlowLayout()

**GridLayout constructor:** new GridLayout (int rows, int cols, int rowSpacing)

BorderLayout is the default layout for WindowController, whereas FlowLayout is default for JPanel.

GUI Components - General

The following methods can be applied to any Component:

```java
void setFont (Font f)
void setForeground (Color c)
void setBackground (Color c)
```

To construct a font use:

```java
new Font (String name, int style, int size)
```

Style can be one of the following:

- `Font.BOLD`
- `Font.ITALIC`
- `Font.PLAIN`
- `Font.BOLD+Font.ITALIC`

Sponsored by Readability-Score.com

Measure your website readability!

https://readability-score.com
### GUI Components - JButton

**Constructor:**
new JButton (String s)

**General Methods:**
- String getText ( )
- void setText (String s)

**Listener Interface:**
- ActionListener

**Adding the listener:**
void addActionListener (ActionListener al)

**Listening Method:**
void actionPerformed (ActionEvent e)

### GUI Components - JComboBox

**Constructor and Initialization:**
new JComboBox ( )

**General Methods:**
- Object getSelectedItem ( )
- String text=(String)menu.getSelectedItem();
- int getSelectedIndex ( )

**Listener Interface:**
- ItemListener
- ActionListener

**Adding the listener:**
void addItemListener (ItemListener il)
void addActionListener (ActionListener al)

**Listening Method:**
void itemStateChanged (ItemEvent e)
void actionPerformed (ActionEvent e)

About methods:
- getSelectedIndex ( ) returns the index of the selected item.
- getSelectedItem ( ) returns the selected item.

(String) menu.getSelectedItem ( ); is a typecast which treats the above returned value as a String

**About the listeners:**
This component can hear the user making a menu selection depending on the chosen interface. Be consistent in your choice of listener interface, adding method, and listening method.

### GUI Components - JLabel

**Constructors:**
- new JLabel (String s)
- new JLabel (String s, int align)

**General Methods:**
- void setText (String s)
- String getText ( )

**Listener Interface:**
- No listeners available.

align can be either JLabel.RIGHT, JLabel.LEFT or JLabel.CENTER.

**General Methods:**
- int getSelectedIndex ( )

**About the listeners:**
This component can hear the user making a menu selection depending on the chosen interface. Be consistent in your choice of listener interface, adding method, and listening method.

### GUI Components - JTextField

**Constructors:**
- new JTextField (String s)

**General Methods:**
- void setText (String s)
- String getText ( )

**Listener Interface:**
- ActionListener

**Adding the listener:**
void addActionListener (ActionListener al)

**Listening Method:**
void actionPerformed (ActionEvent e)

### GUI Components - JSlider

**Constructor:**
new JSlider (int orientation, int minimum, int maximum, int initialValue)

**General Methods:**
- void setValue (int newVal)
- int getValue ( )

**Listener Interface:**
- ChangeListener

**Adding the Listener:**
addChangeListener (ChangeListener)

**Listening Method:**
void stateChanged (ChangeEvent e)

orientation can be either JSlider.HORIZONTAL or JSlider.VERTICAL.

### GUI Components - JTextField

**Constructors:**
- new JTextField (String s)

**General Methods:**
- void setText (String s)
- String getText ( )

**Listener Interface:**
- ActionListener

**Adding the Listener:**
void addActionListener (ActionListener al)

**Listening Method:**
void actionPerformed (ActionEvent e)