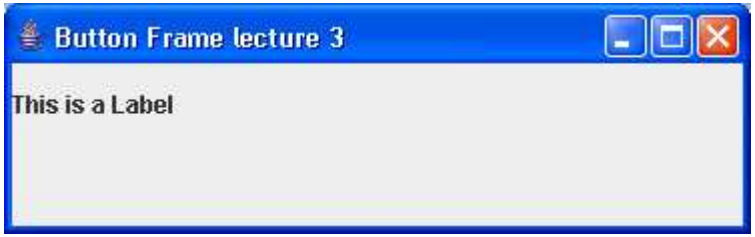
More Widgets

JLabel

- This is very simple, it just displays text with no user input.
  - It does not generate events.
  - So does not have a listener.
- Useful next to a `JTextField`.
  - `JTextField` inputs text but does not display it.

```
JLabel label1 = new JLabel("This is a Label");
add(label1);
```

- This is how it looks in a 2 x 2 grid.



JTextField

- A `JTextField` object displays a text input box
  - The user can type in a string.
  - An event is generated when the user types a newline (ENTER)
- The constructor specifies the number of columns in the input box.
  - It can also specify an initial string.
- There are 3 constructor:
  - `JTextField(int ncol);`
  - `JTextField(String text);`
  - `JTextField(String text, int ncols);`
- The program can set the text in a `JTextField` object with `setText`.

JTextField Events

- A `JTextField` will generate an event when the user types ENTER.
- It is an `ActionEvent`.
- So we need to add an `ActionListener`.
  - Just like a button.
- The event is delivered to the `actionPerformed` method.
- We can find out which text field sent the event from `getSource().`
  - In the same way as buttons.
- We can then call the `getText` method of `JTextField`
  - To find out the text that the user entered.

Example

- The following example creates 3 labels and associated text fields.
  - Shown in a 3 x 2 grid.
- Extra spaces are added to some of the labels.
  - To make them all the same size.
  - So that they line up neatly.

Creating the Widgets

```
        // create labels and text fields
label1 = new JLabel("Just ncol    ");
label2 = new JLabel("Just text    ");
label3 = new JLabel("Text and ncol");
field1 = new JTextField(20);
field2 = new JTextField("Initial text");
field3 = new JTextField("Initial text", 20);

        // listen to the text fields
field1.addActionListener(this);
field2.addActionListener(this);
field3.addActionListener(this);
```

Adding Them To The JFrame

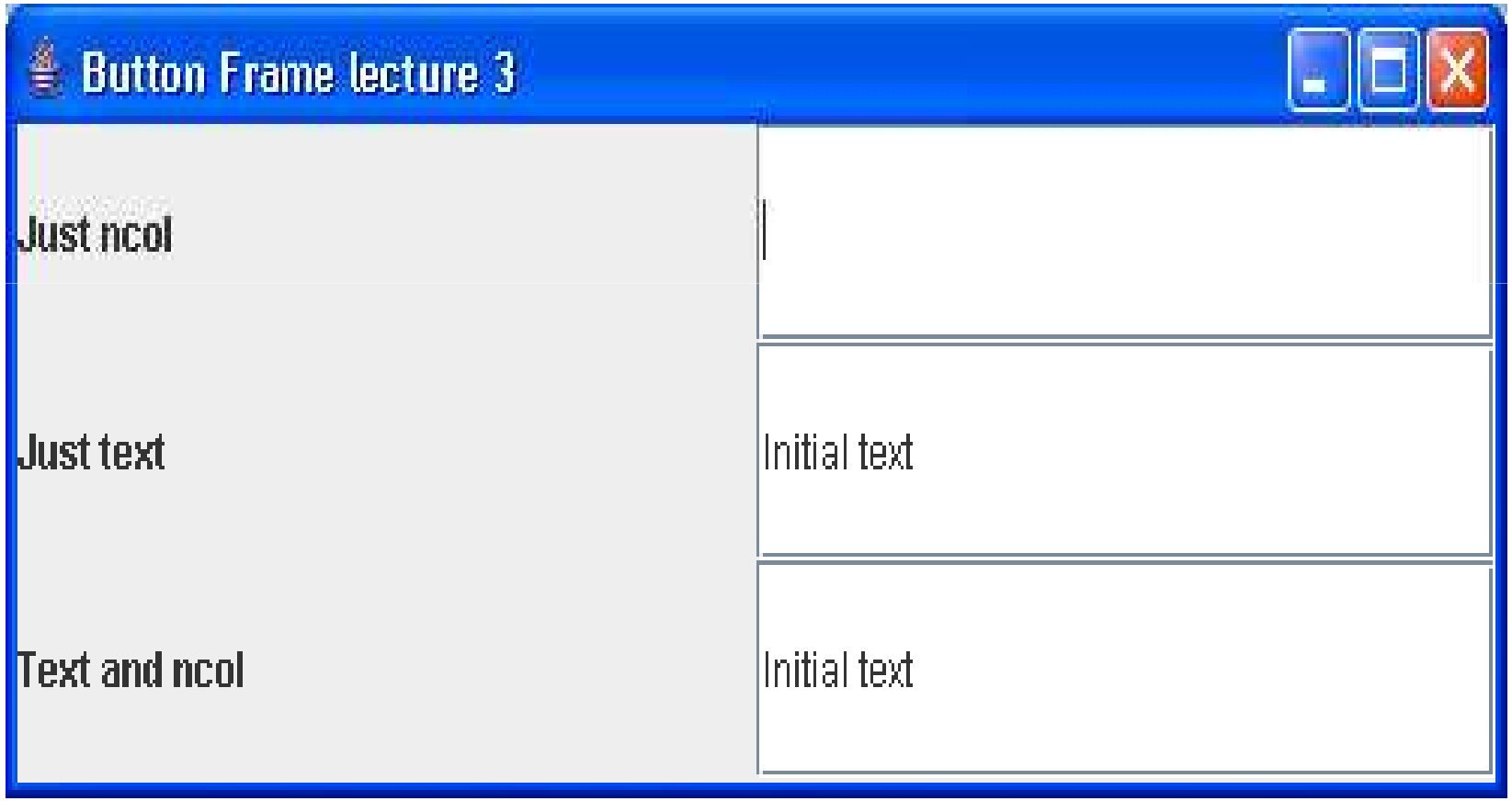
```
        // add them in the right order
add(label1);
add(field1);
add(label2);
add(field2);
add(label3);
add(field3);
```

Processing The Events

```
public void actionPerformed(ActionEvent e)
{
    String text;

    if (e.getSource() == field1)
          text = field1.getText();
    else if (e.getSource() == field2)
          text = field2.getText();
    else
          text = field3.getText();
    System.err.println(text);
}
```

The Window


Button Frame lecture 3
Just ncol
Just text
Initial text
Text and ncol
Initial text

Example With setText

- The previous example is changed slightly so that anything entered in field1 is copied to field2 and field3.

```
if (e.getSource() == field1)
{
      text = field1.getText();
      field2.setText(text);
      field3.setText(text);
}
```

Read Only Text Fields

- A JTextField is normally editable.
  - The user can enter information and chance existing information.
- Editing can be disabled by calling the method
  - field.setEditable(false);
- Naturally, using a true parameter makes it editable again.

Input Focus

- The input ***focus*** of a GUI is the widget where input is sent.
- It is normally indicated by the cursor position or highlighted in other ways.
- Any widget can request the focus by calling the `requestFocus` method.
  - `inField.requestFocus();`
- The user does not have to reposition the focus with the mouse.

- The focus can also be moved by pressing the tab key.
  - The focus will move from widget to widget in the order that they were added to the window or panel.
- Shift-Tab will move the focus in the opposite order.

Entering Numbers

- A `JTextField` only delivers strings from the user to the program.
- If we want to enter a number, we must.
  - Get it as a string using `getText`.
  - Convert the string to a number.
- The `Integer` and `Double` classes will do this.
  - `int i = Integer.parseInt(String);`
  - `double d = Double.parseDouble(String);`

White Space

- Leading and trailing white space can cause problems.
  - It is better to remove it with the `trim` method of `String`.
  - `trim` returns a string with leading and trailing white space removed.
- A `NumberFormatException` will be thrown if the string is not a number.
  - Either by `Integer.parseInt()` or `Double.parseDouble()`.
- If we don't catch it our program may crash with an unhandled exception.

field2 Expects a Number

```
else if (e.getSource() == field2)
{
   text = field2.getText();
   try
   {
       int n = Integer.parseInt(text.trim());
       System.err.println("Integer value = " + n);
   }
   catch (NumberFormatException x)
   {
       System.err.println(text + " is not a Number");
   }
```

Putting A Number Into A Text Field

- This is easier because we already know how to convert a number into a String.
  - `“” + number`.
- If we wanted to put the number 42 into our text field
  - `setText(“” + 42);`
- This example initialises `field3` to the number 42.
  - `field3 = new JTextField("" + 42, 20);`

Summary

- `JLabel` just displays text.
- `JTextField` lets the user enter text
  - Needs an `ActionListener`
  - Events delivered to `actionPerformed`.
  - `getText` to find out the text entered.
  - `setText` to add our text.
- `String.trim()` removes white space around a string.
- `Integer.parseInt(String)` converts a string to an integer.
- `Double.parseDouble` converts a string to a double.