Section Fifteen – Absolute Layouts

In this section the Developer will learn…

- How to create Absolute Layouts for Component display in a JFrame

Absolute Layouts - Introduction

The next section will introduce the Developer to the use of Layouts which allow the positioning of Components within Containers achieving a standard design.

This section examines the use of Absolute Layouts which can be used when only a simple output is required.

The principle method used in the first example is the use of setBounds() which is relative to the Container in which the Component resides, in previous examples setBounds() was used to place the JFrame on the screen.

Absolute Layouts – setBounds()

Here the Developer is creating output based on a JFrame, JPanel and three JLabels, in order to make this work the Developer must specify that the JPanel doesn’t use a layout (layouts are normally added to mid-level containers) …

```
pane.setLayout(null);
```

The JPanel will be added to a JFrame which has had its size set …

```
frame.setSize(300, 200);
```

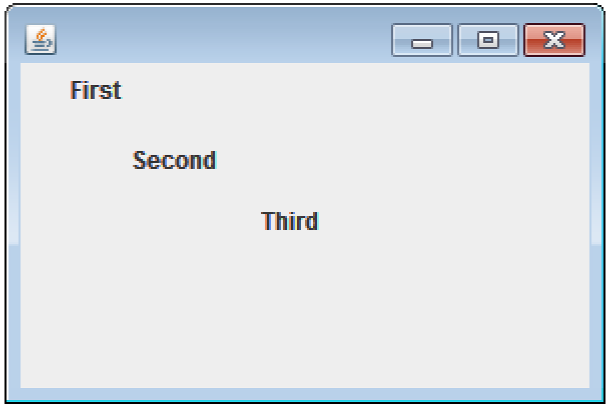
The JLabel components will have their positions set relevant to the outside edge of the JFrame …

Absolute Layouts – setBounds()

```
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class absolute  {

     static JPanel pane = new JPanel();
     static JLabel label1 = new JLabel("First");
     static JLabel label2 = new JLabel("Second");
     static JLabel label3 = new JLabel("Third");

    public static void main(String[] args) {

         JFrame frame = new JFrame();
         frame.setSize(300, 200);
         pane.setLayout(null);
         pane.add(label1);
         pane.add(label2);
         pane.add(label3);
```

Absolute Layouts – setBounds()

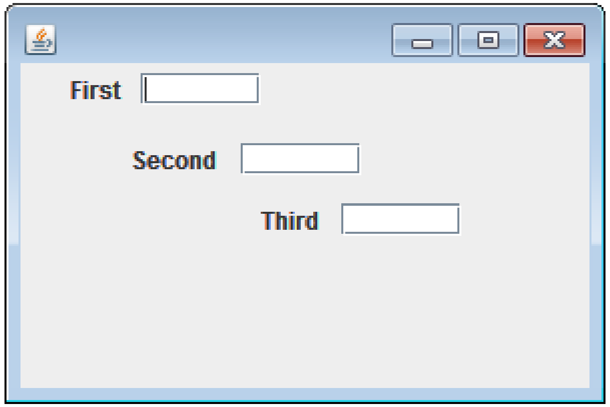
```
    label1.setBounds(25, 5, 30, 16); // from the left, from the top, width, height
    label2.setBounds(55, 40, 60, 16);
    label3.setBounds(120, 70, 30, 16);

    frame.getContentPane().add(pane);
    frame.setVisible(true);

  }

}
```



Absolute Layouts – setBounds()

In this example the Developer is using a JTextField which will be introduced later in the course to align JLabels with their input fields …

```
label1.setBounds(25, 5, 30, 16); // from the left, from the top, width, height
field1.setBounds(60, 5, 60, 16); // Ordering of the setBounds() is not significant
label2.setBounds(55, 40, 60, 16);
field2.setBounds(110, 40, 60, 16);
label3.setBounds(120, 70, 60, 16);
field3.setBounds(160, 70, 60, 16);
```


First
Second
Third