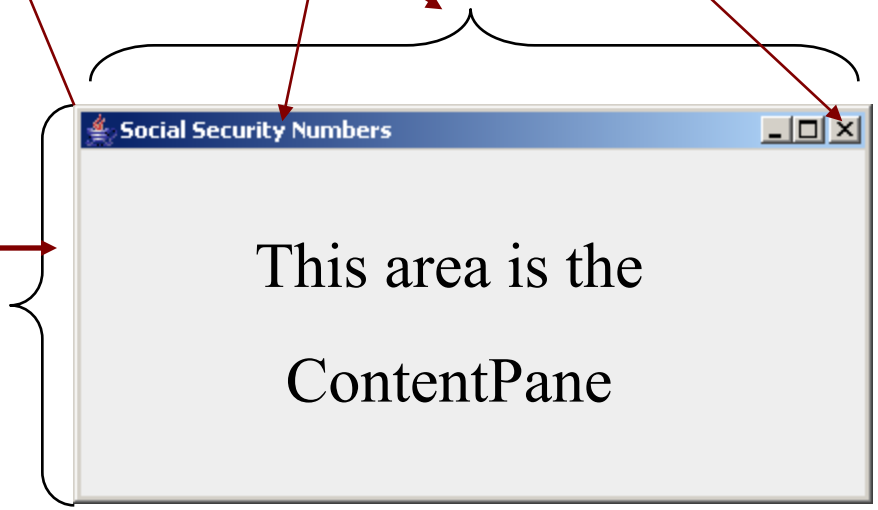
Using a Simple GUI

Creating a JFrame

```
public static void initialize() {
    mySSNGUI=new JFrame();
    mySSNGUI.setSize(400, 200);
    mySSNGUI.setLocation(100, 100);
    mySSNGUI.setTitle("Social Security Numbers");

mySSNGUI.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    mySSNGUI.setVisible(true);
  }
```

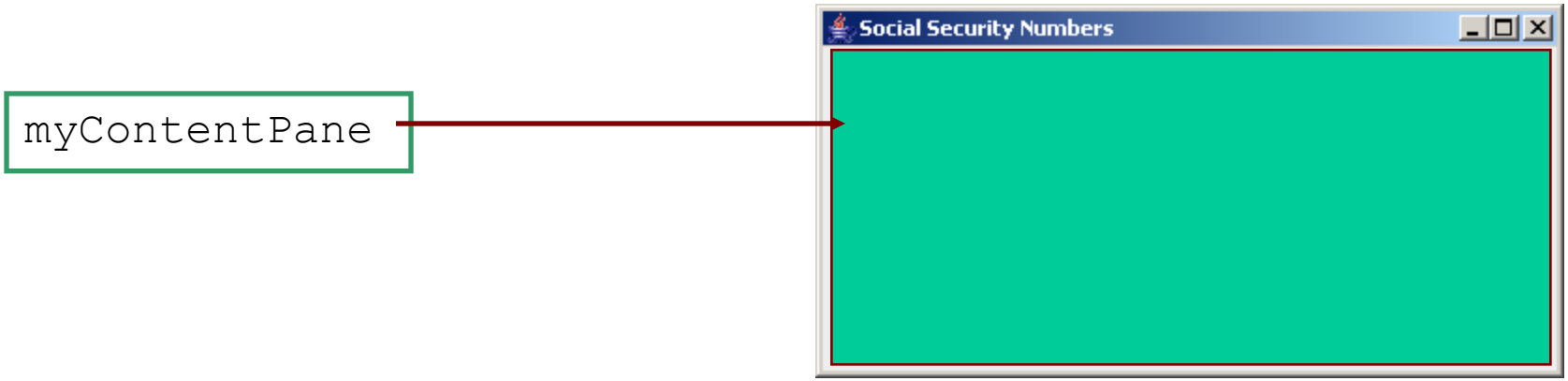
mySSNGUI



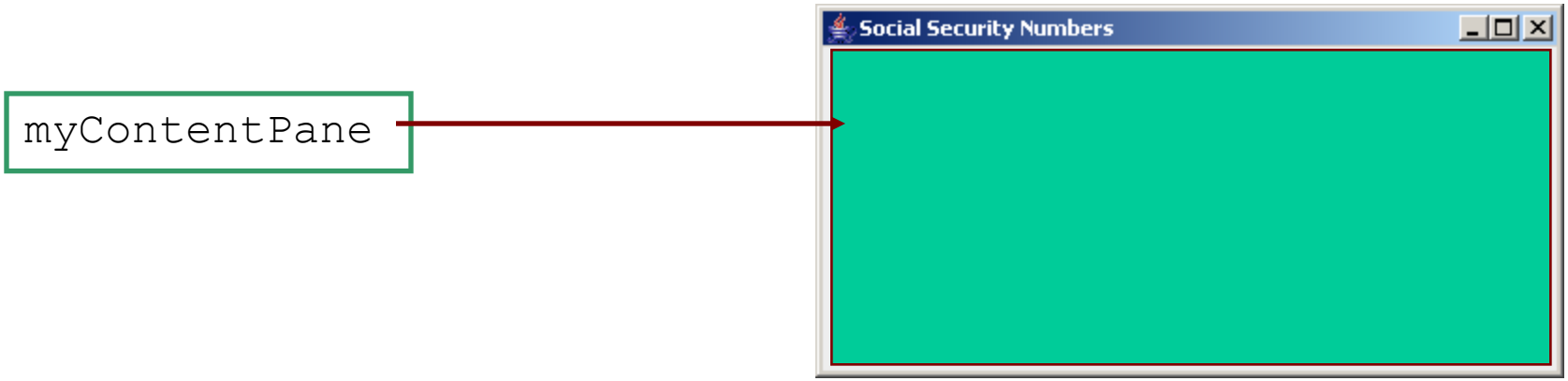
Putting data in a JFrame

```
public static void printSSNtoJFrame
                    (JFrame jf, String[] list, int size) {
  Container myContentPane = jf.getContentPane();
  TextArea myTextArea = new TextArea();
  myContentPane.add(myTextArea);
    for (int i=0;i<size;i++)
      if (!isValidSSN(list[i]))
        myTextArea.append("Invalid SSN: "+list[i]+"\n");
      else
        myTextArea.append(list[i]+"\n");
    jf.setVisible(true);
}
```

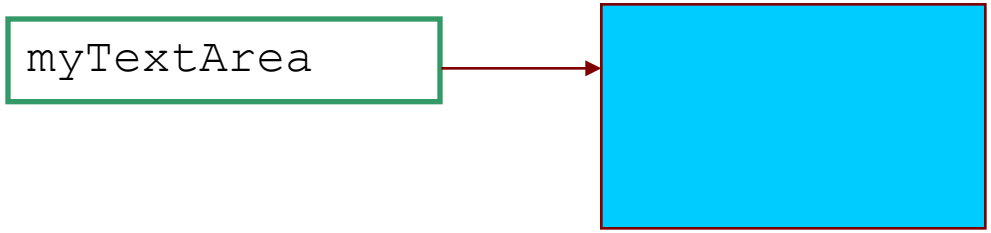
```
Container myContentPane = jf.getContentPane();
```



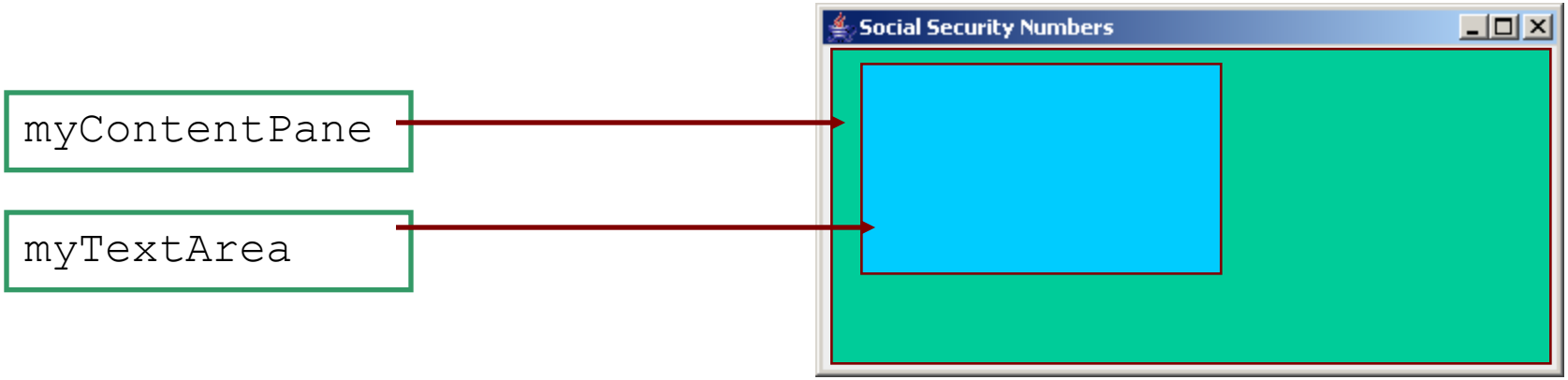
```
Container myContentPane = jf.getContentPane();
```



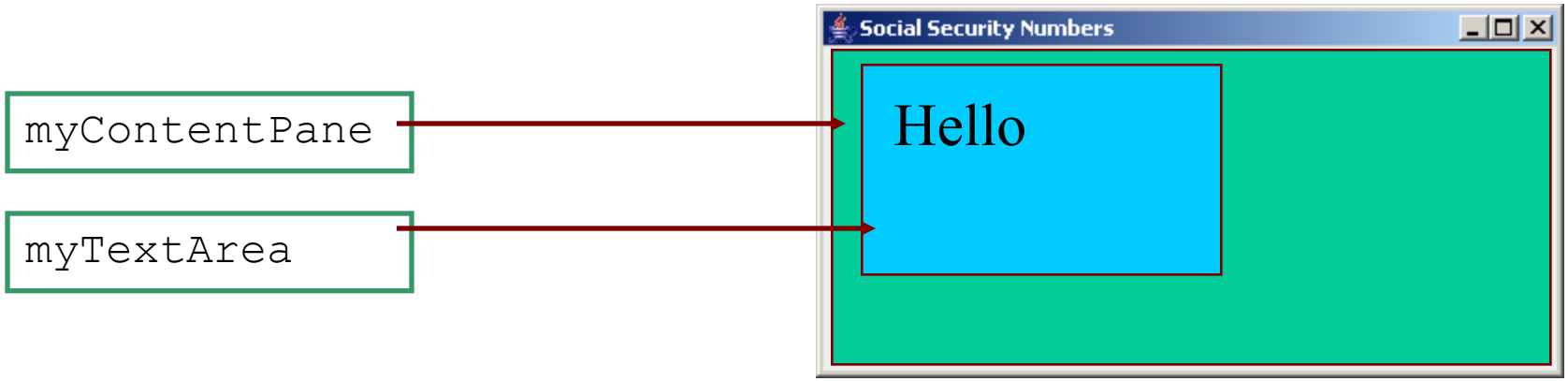

```
TextArea myTextArea = new TextArea();
```



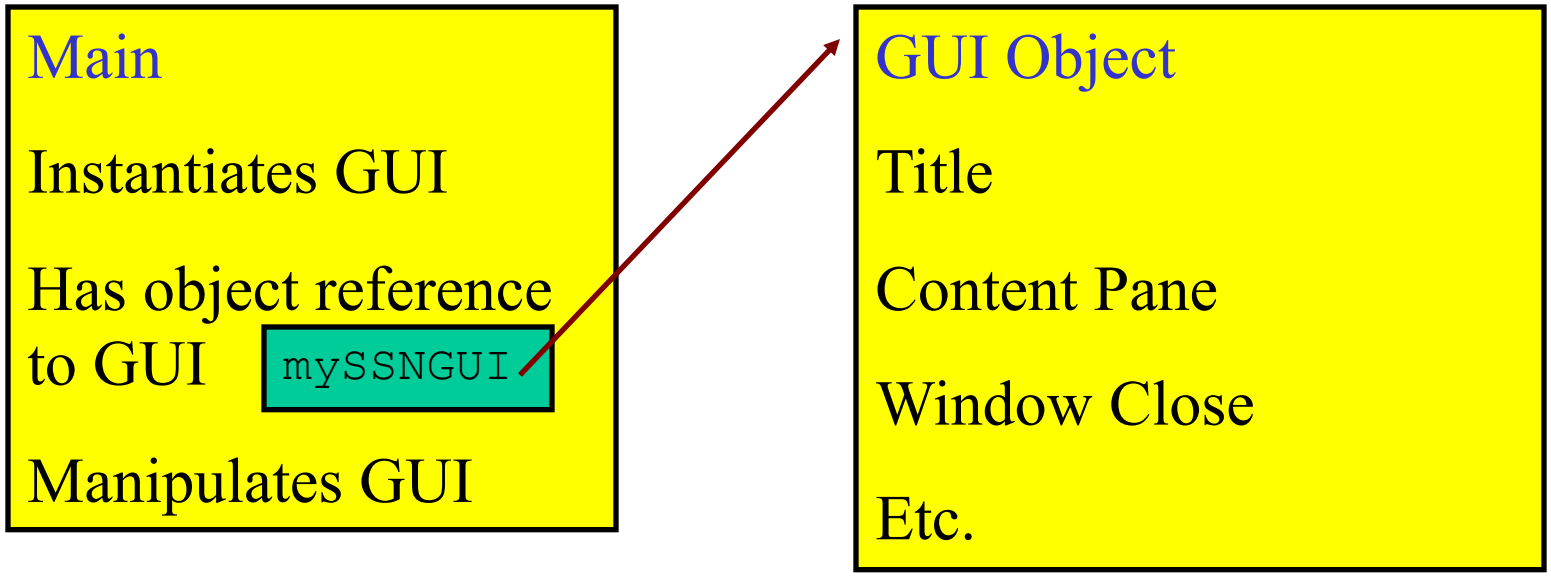
```
myContentPane.add(myTextArea);
```




```
myTextArea.append("Hello\n");
```



It is more common for the "main" application and the GUI to be separate classes.




```
public class SSN {…
```

~~mySSNGUI = new Jframe();~~

```
  mySSNGUI = new SSNGUI();
```

```
public class SSNGUI {…
```

```
public class SSNGUI {…

}
```

But do we have to write code here for all things a GUI (JFrame) can do?

NO! We can take advantage of the most powerful Object-Oriented feature…

Inheritance

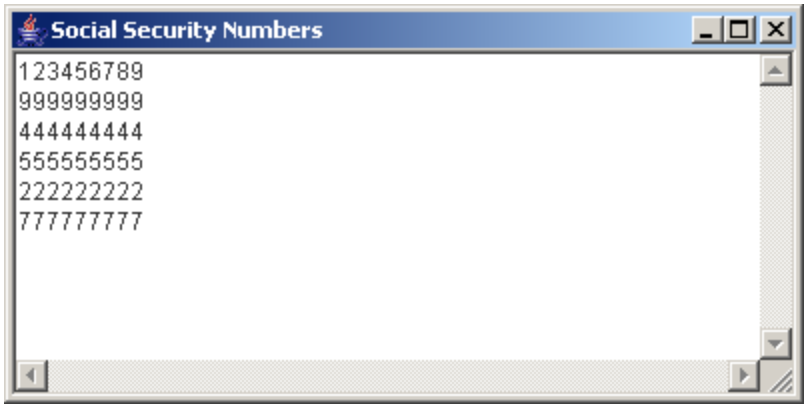
```
public class SSNGUI extends JFrame{…



}
```

```
public class SSNGUI extends JFrame{…



}
```

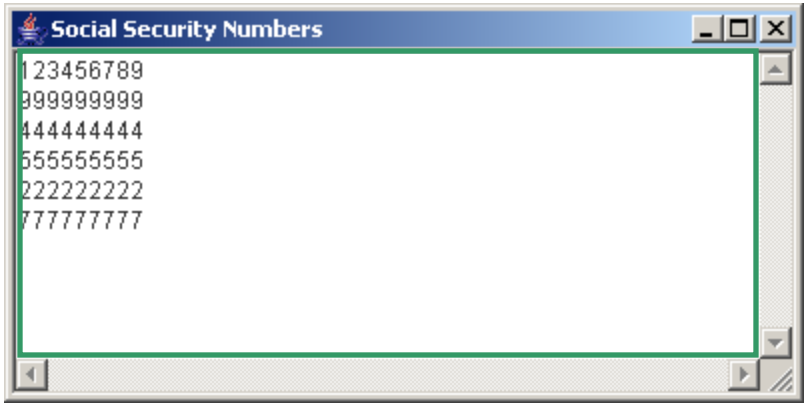
The class SSNGUI can now do whatever a JFrame can do, as well as whatever we add to it.

```
public static void initialize() {
  mySSNGUI=new SSNGUI();
  mySSNGUI.setSize(400, 200);
  mySSNGUI.setLocation(100, 100);
  mySSNGUI.setTitle("Social Security Numbers");
  mySSNGUI.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  mySSNGUI.setVisible(true);
   }
```

The program SSNFrame3 produces this SSNGUI:



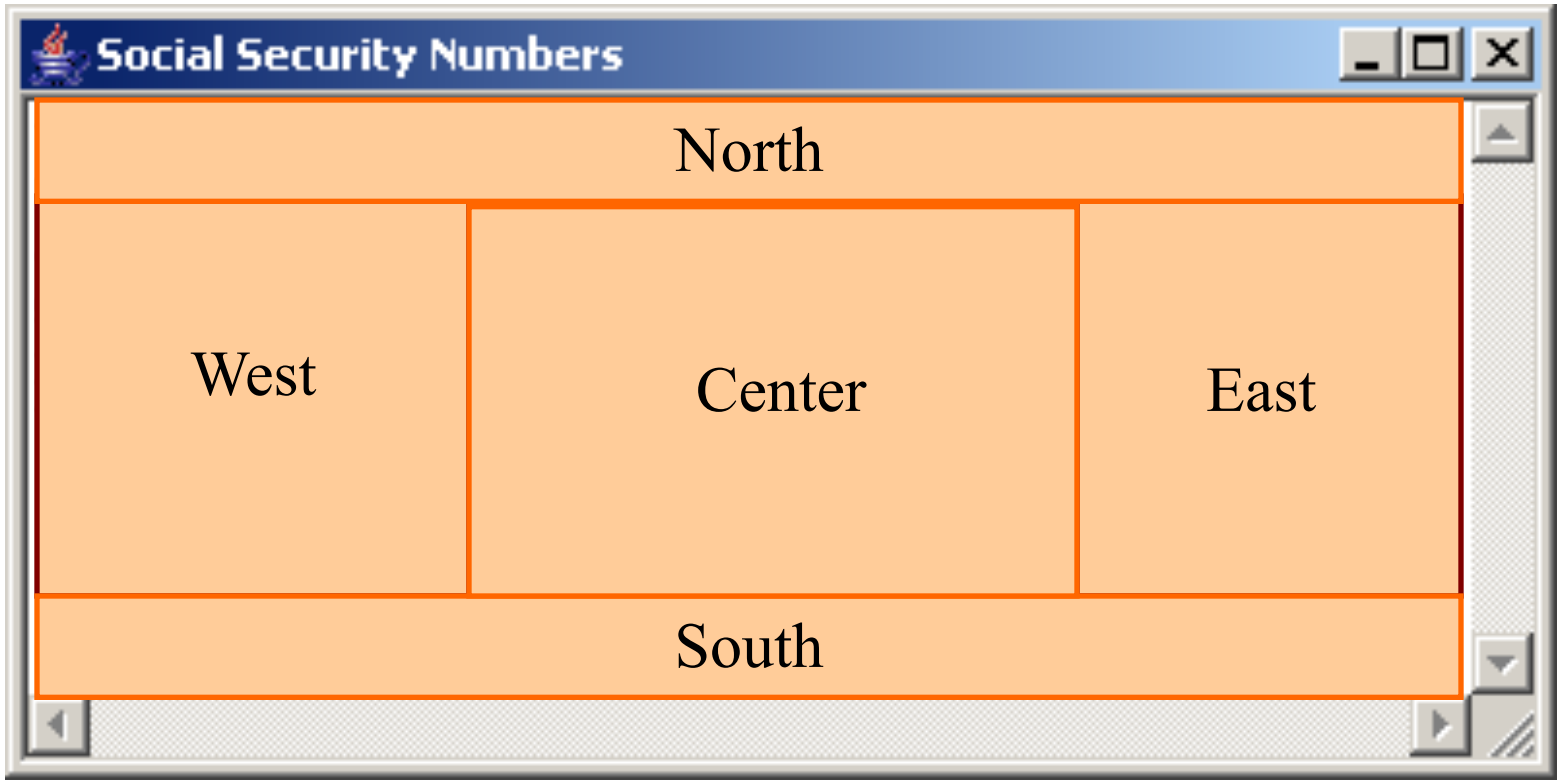
The entire ContentPane is one TextArea



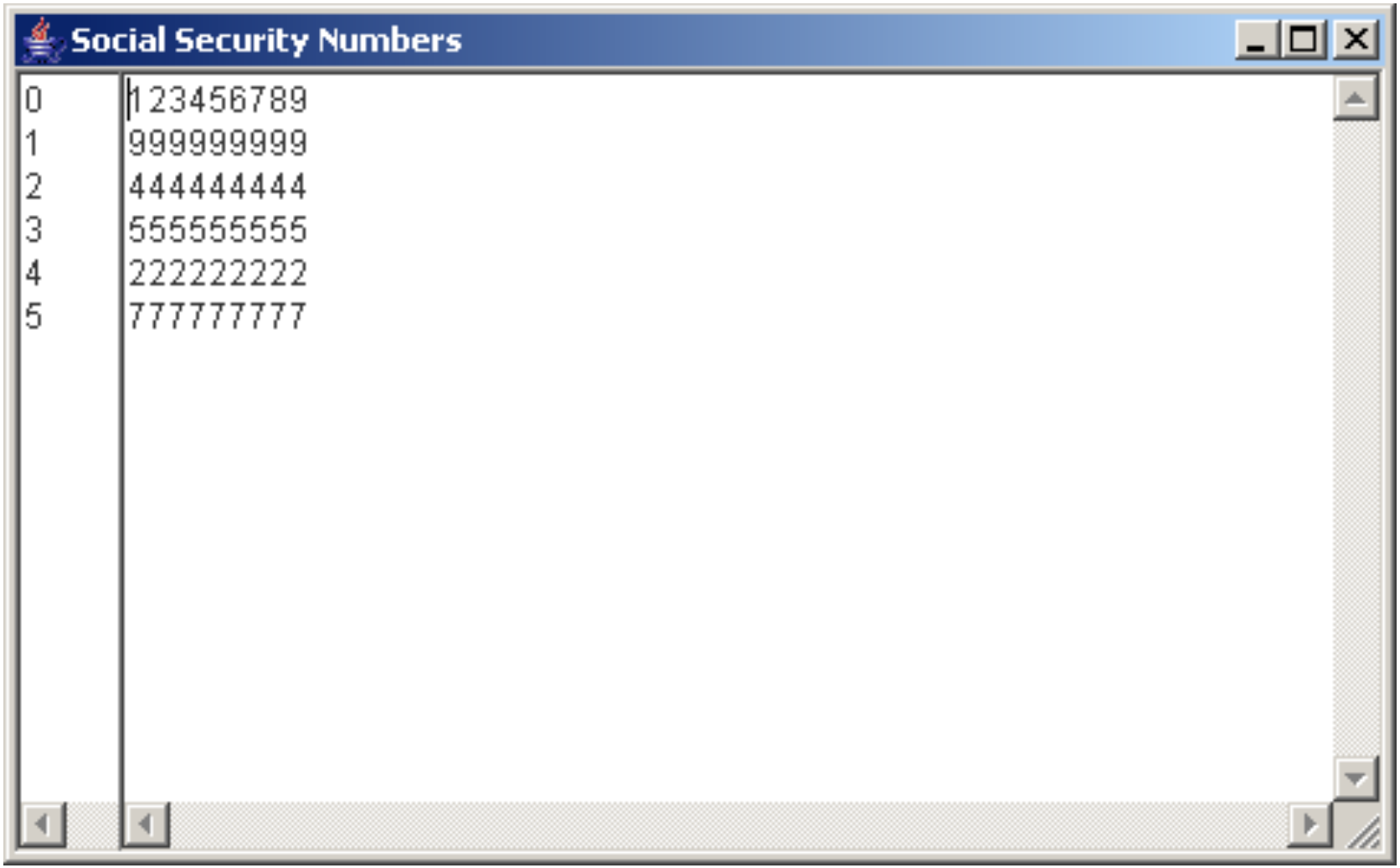
To "divide" the ContentPane into different areas, we can use a LayoutManager

- There are several LayoutManagers available, including
- BorderLayout
- GridLayout

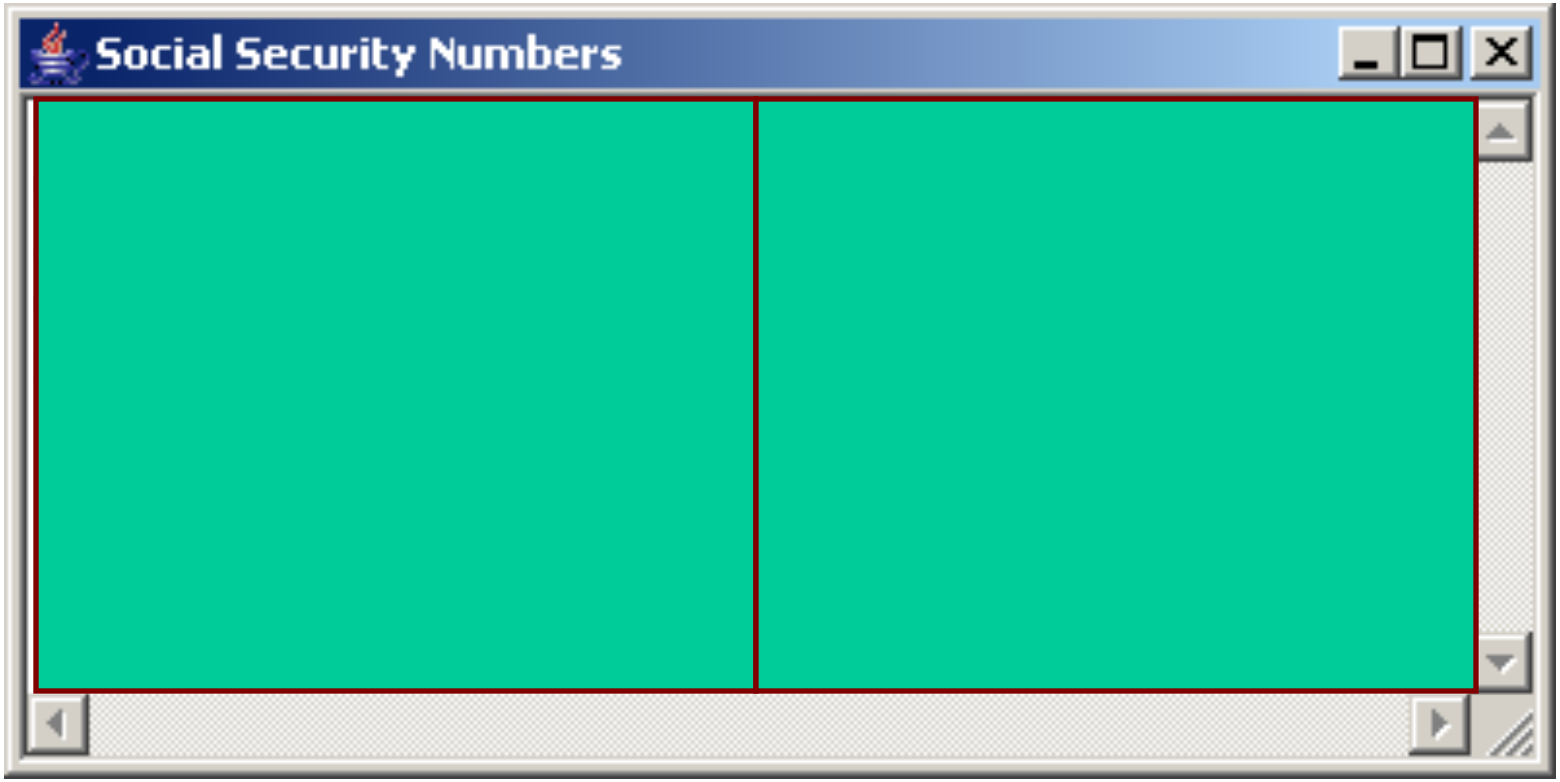
BorderLayout



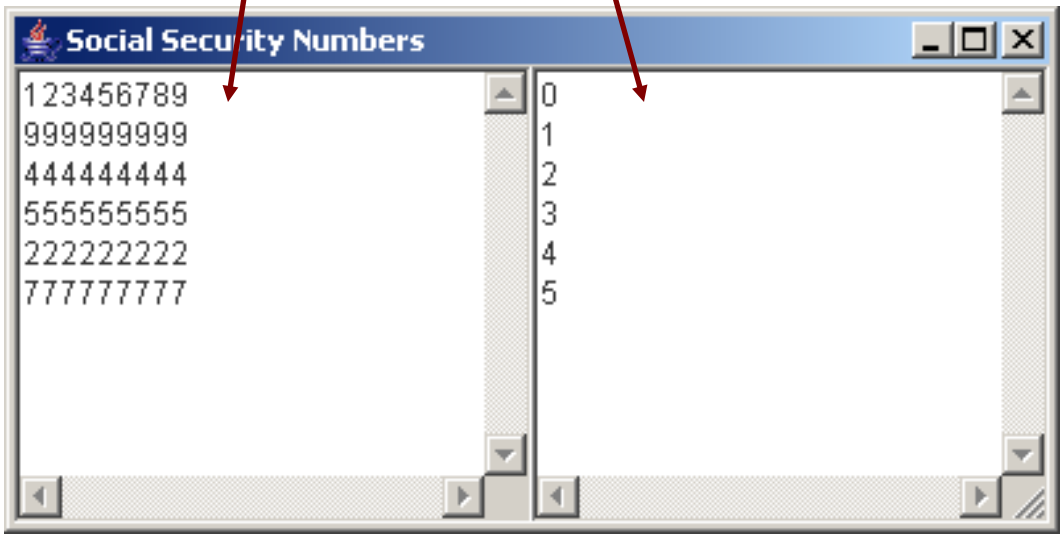
```
public static void printSSNtoSSNGUI
                (SSNGUI ssnGUI,  String[] list, int size) {
  Container myContentPane = ssnGUI.getContentPane();
  TextArea myTextArea = new TextArea();
  TextArea mySubscripts = new TextArea();
  myContentPane.add(myTextArea, BorderLayout.EAST);
  myContentPane.add(mySubscripts, BorderLayout.WEST);
    for (int i=0;i<size;i++) {
      mySubscripts.append(Integer.toString(i)+"\n");
      if (!isValidSSN(list[i]))
        myTextArea.append("Invalid SSN: "+list[i]+"\n");
      else
        myTextArea.append(list[i]+"\n");
    }
    ssnGUI.setVisible(true);
```


Social Security Numbers
0 123456789
1 999999999
2 444444444
3 555555555
4 222222222
5 777777777

```
mySSNGUI.setLayout(new GridLayout(1,2));
```



```
myContentPane.add(myTextArea);
myContentPane.add(mySubscripts);
  for (int i=0;i<size;i++) {
    mySubscripts.append(Integer.toString(i)+"\n");
      if (!isValidSSN(list[i]))
        myTextArea.append("Invalid SSN: "+list[i]+"\n");
      else
        myTextArea.append(list[i]+"\n");
}
```



Non-application Classes (no "main" method)

- Classes without "main" methods are true objects
- They cannot exist without being instantiated
- They may inherit from other classes so little extra code must be written
- When an object is instantiated, a special method called a "constructor" is automatically executed.
- The name of the constructor is the same as the name of the class.

```
import javax.swing.*;

public class SSNGUI extends JFrame {

  public SSNGUI () {


  }


}
```

- The constructor has the same name as the class
- The constructor has no "return" attributes
- The constructor is the initialization method for the object
- The constructor may take parameters from the instantiating method for initial values

```java
import javax.swing.*;
public class SSNGUI extends JFrame {
  public SSNGUI(String title) {
    setTitle(title);
}
```

```java
import javax.swing.*;
public class SSNGUI extends JFrame {
  public SSNGUI(String title, int height, int width) {
    setTitle(title);
    setSize(height, width);
}
```