PROGRAMMING

Lecture 21

Sushil Paudel

PREVIOUS TOPIC

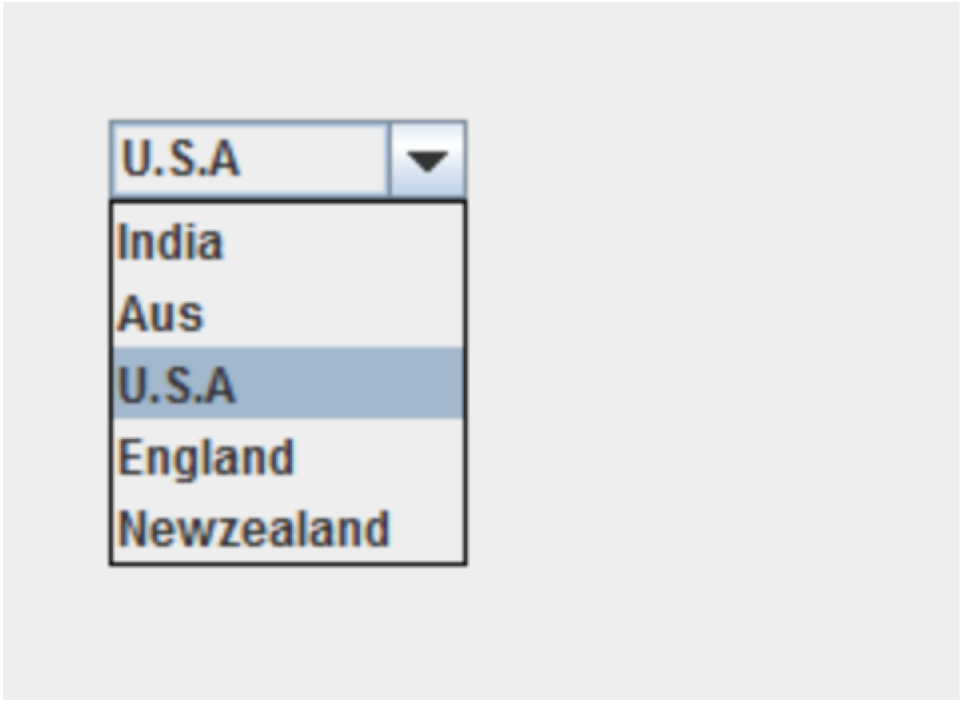
- GridLayout
- GridbagLayout

TODAY'S TOPIC

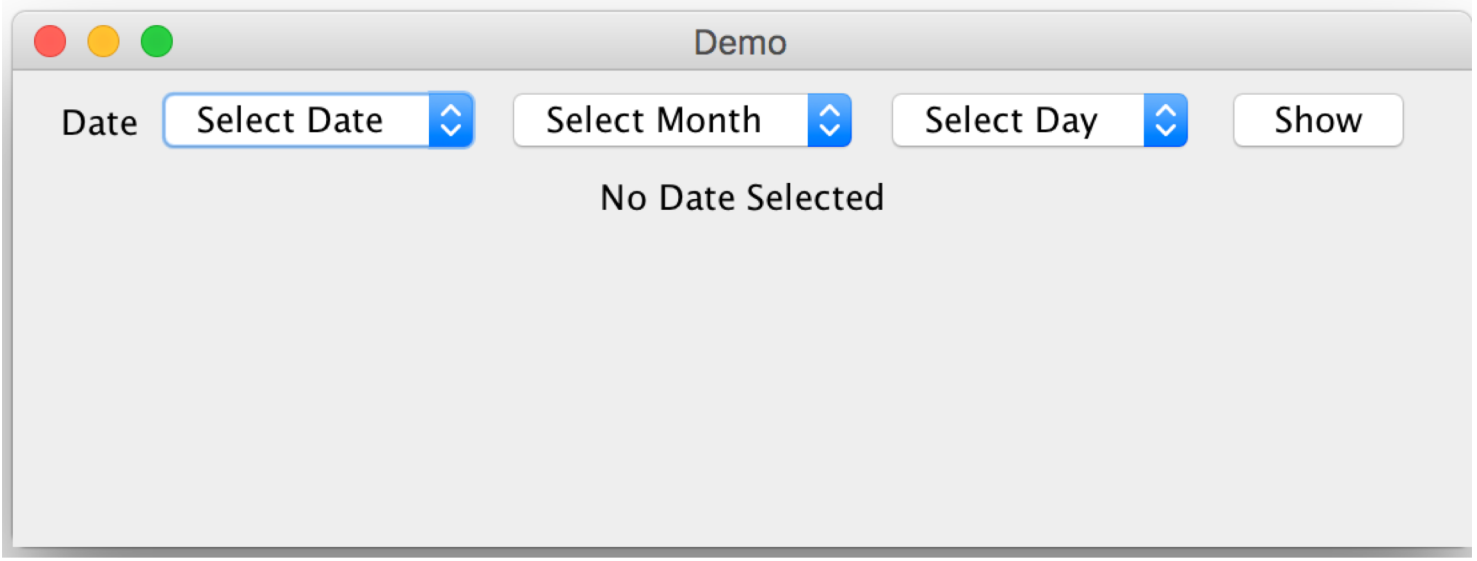
- Combo Box
- Data Table
- GUI designs

COMBO BOX

- JComboBox shows a popup menu that shows a list and the user can select a option from that specified list.



CREATING A COMBO BOX



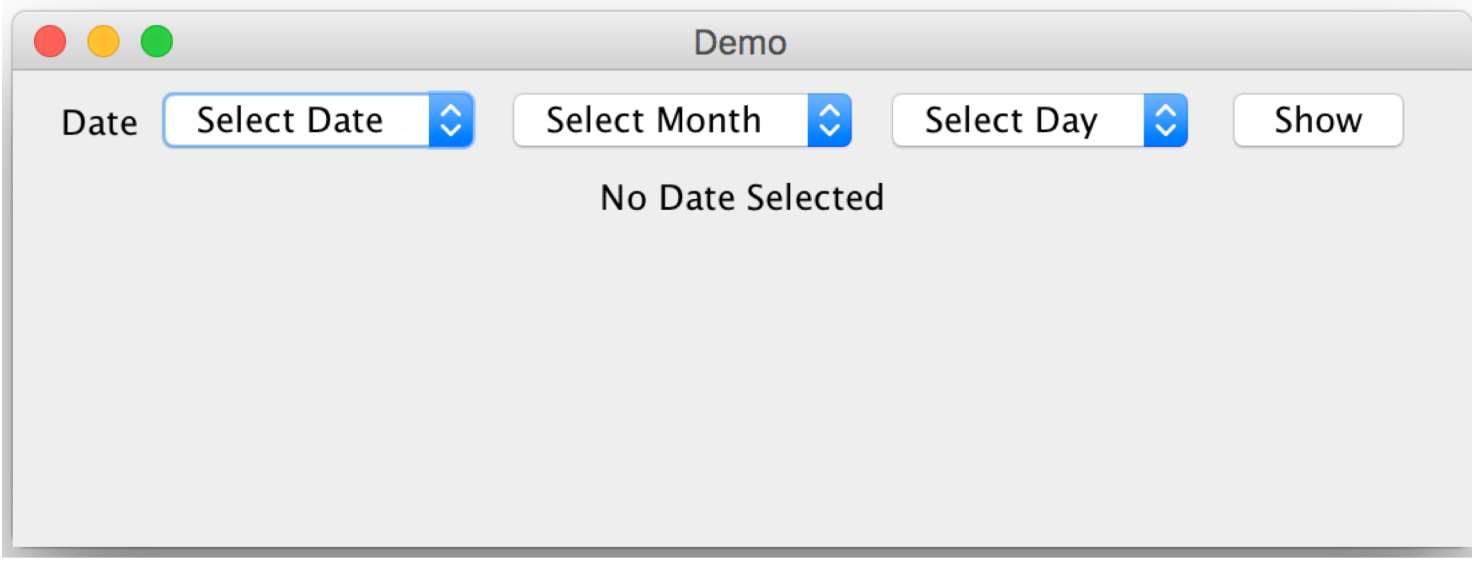
COMBO BOX

```
public static void main(String[] args) {

    JFrame frame = new JFrame("Demo");

    JPanel panel = new JPanel();
    panel.setLayout(new FlowLayout());

    JLabel dateLabel = new JLabel("Date");
    String years[] = {"Select Date", "2022", "2023", "2024", "2025", "2026"};
    String months[] = {"Select Month", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12"};
    String days[] = {"Select Day", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12",
                     "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26",
                          "27", "28", "29", "30", "31"};

    JComboBox yearsComboBox = new JComboBox(years);
    JComboBox monthsComboBox = new JComboBox(months);
    JComboBox daysComboBox = new JComboBox(days);

    JButton button = new JButton("Show");
    JLabel resultLabel = new JLabel("No Date Selected");
```

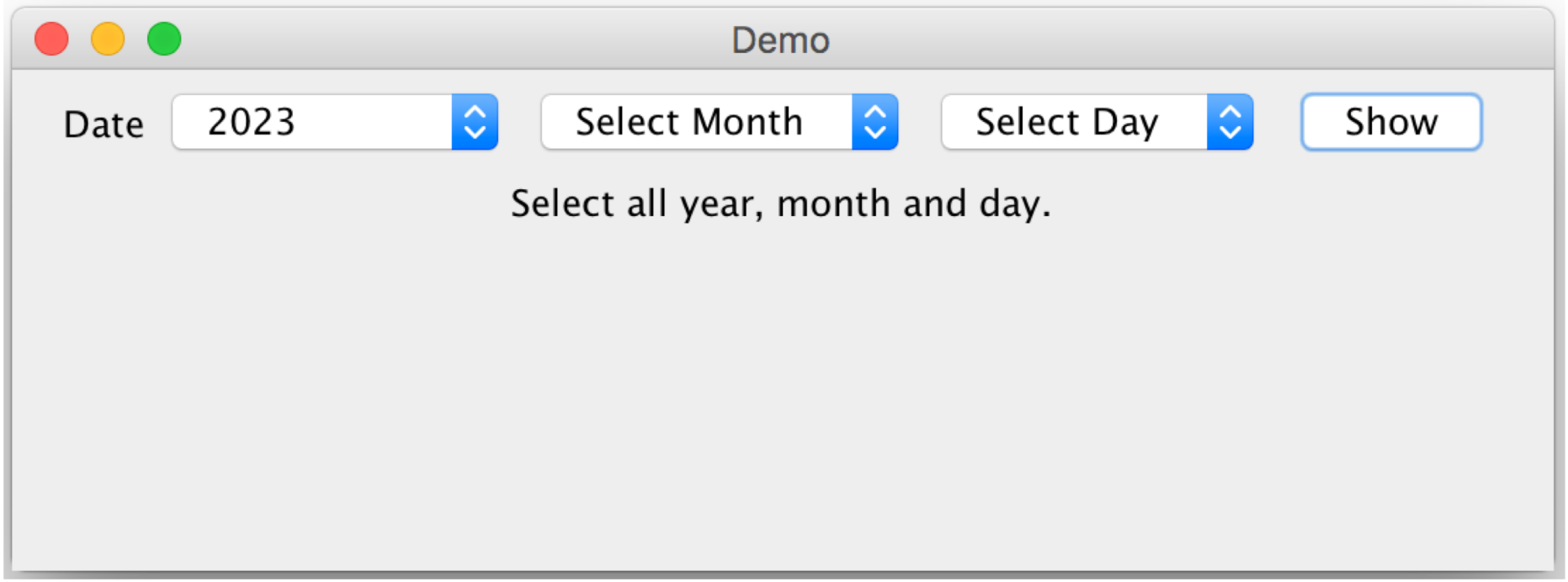
COMBO BOX

```
button.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (yearsComboBox.getSelectedIndex() == 0 || monthsComboBox.getSelectedIndex() == 0
                        || daysComboBox.getSelectedIndex() == 0) {
                resultLabel.setText("Select all year, month and day.");
            } else {
                resultLabel.setText(yearsComboBox.getSelectedItem().toString() + "- " +
                        monthsComboBox.getSelectedItem().toString() + "-" +
                        daysComboBox.getSelectedItem().toString());
            }
        }
    });

    panel.add(dateLabel);
    panel.add(yearsComboBox);
    panel.add(monthsComboBox);
    panel.add(daysComboBox);
    panel.add(button);
    panel.add(resultLabel);

    frame.add(panel);
    frame.setSize(500, 200);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);

}
```

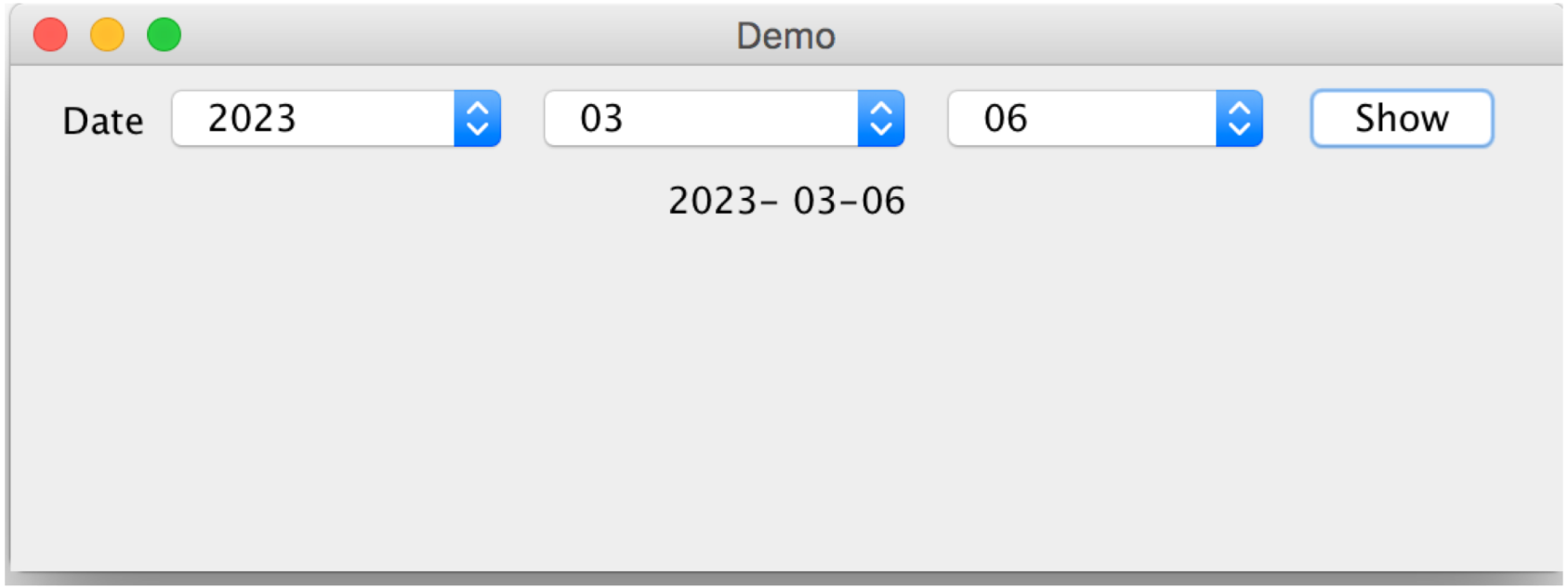
OUTPUT


Demo
Date
Select Date
Select Month
Select Day
Show
No Date Selected

OUTPUT


Demo
Date
2023
Select Month
Select Day
Show
Select all year, month and day.

COMBO BOX



2 DIMENSIONAL ARRAY

- A multidimensional array is an array of arrays.
- To create a two-dimensional array, add each array within its own set of **curly braces**:

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
```

EXAMPLE

- **myNumbers** is now an array with two arrays as its elements.

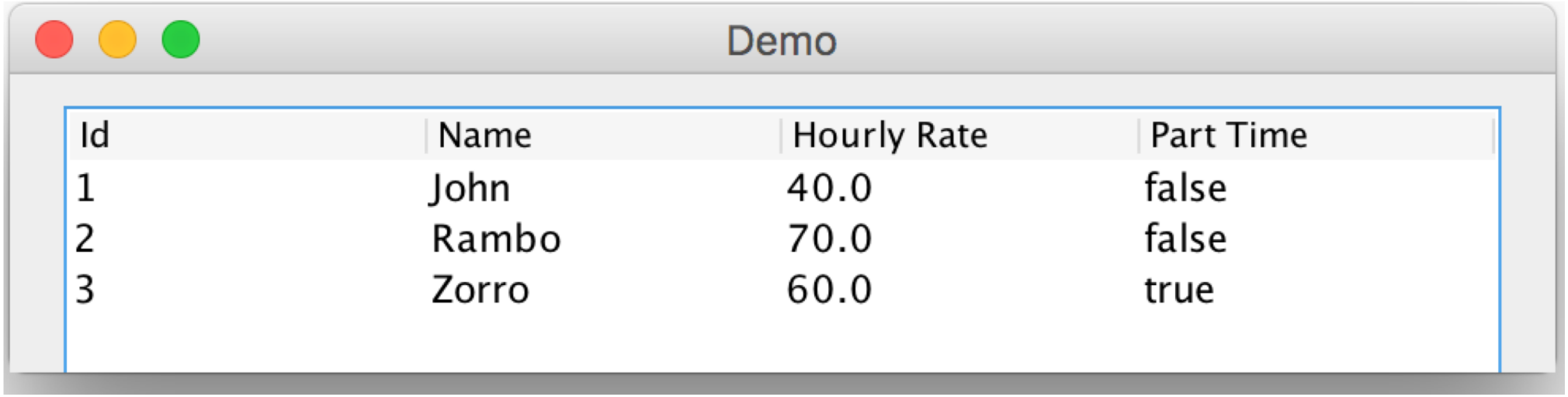
TODAY'S TOPIC

- To access the elements of the **myNumbers** array, specify two indexes: one for the array, and one for the element inside that array. This example accesses the third element (2) in the second array (1) of myNumbers:

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
int x = myNumbers[1][2];
System.out.println(x); // Outputs 7
```

JTABLE

- The JTable class is used to display data in tabular form.
- It is composed of rows and columns



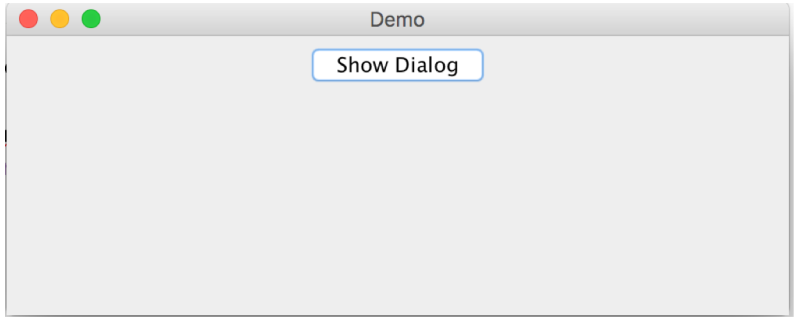
SOURCE CODE

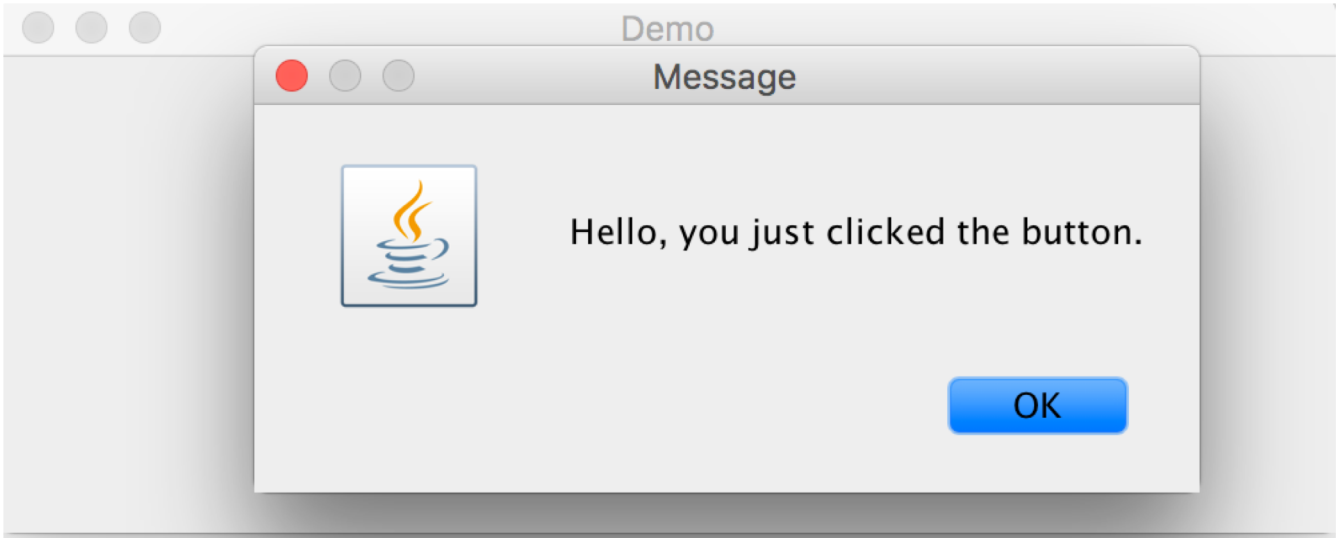
```
JPanel panel = new JPanel();
panel.setLayout(new FlowLayout());

String[] columns = {"Id", "Name", "Hourly Rate", "Part Time"};


 // Object is used to store different types of data in the array
Object[][] data = new Object[][] {
        {1, "John", 40.0, false },
        {2, "Rambo", 70.0, false },
        {3, "Zorro", 60.0, true },
};

JTable jTable = new JTable(data, columns);
panel.add(new JScrollPane(jTable));
```

POP UP DIALOG BOX

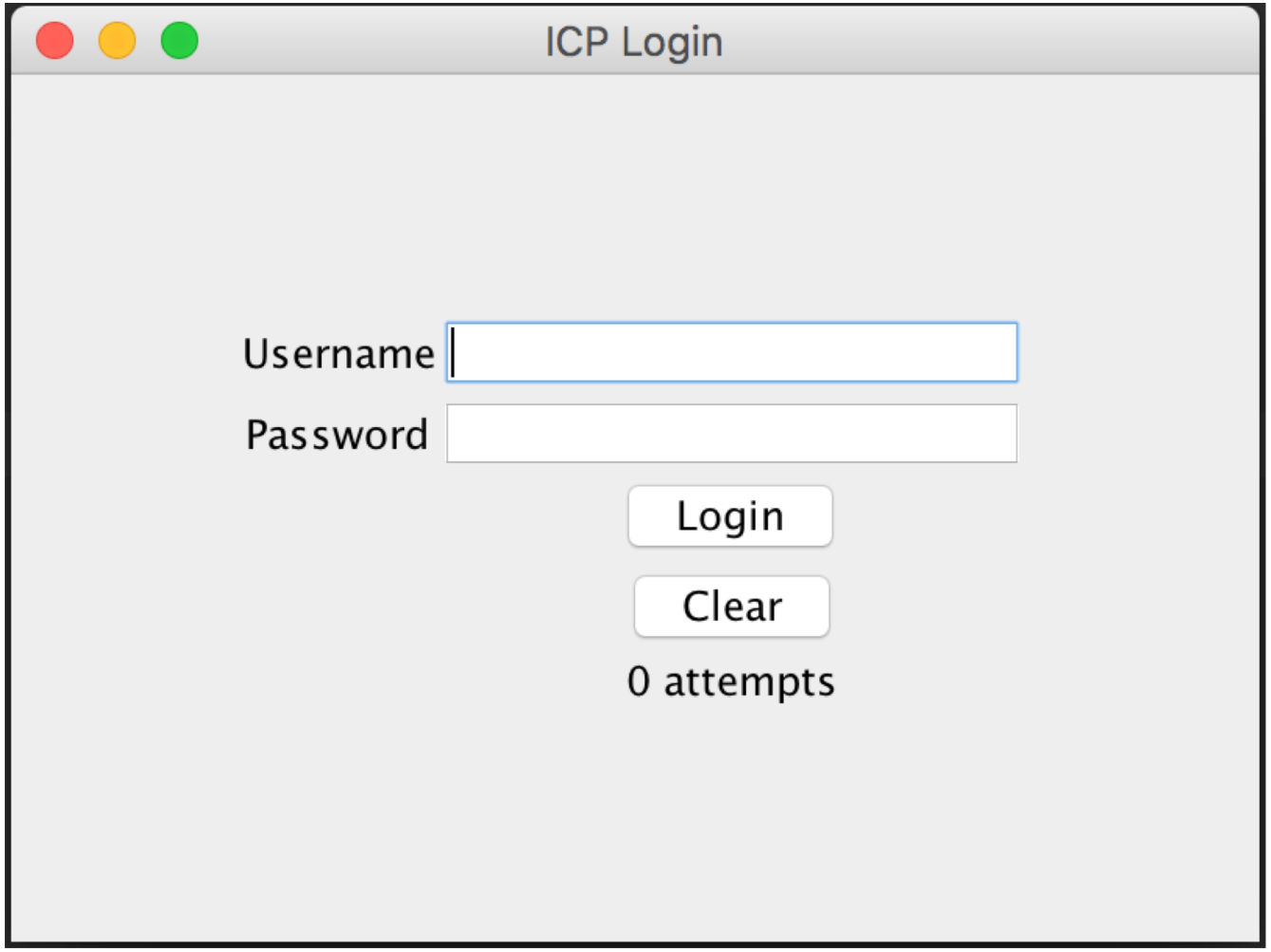
```
JFrame frame = new JFrame("Demo");
frame.setLayout(new FlowLayout());

JButton button = new JButton("Show Dialog");
button.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(frame, "Hello, you just clicked the button.");
    }
});

frame.add(button);
frame.setSize(500, 200);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
```

OUTPUT


Demo
Show Dialog

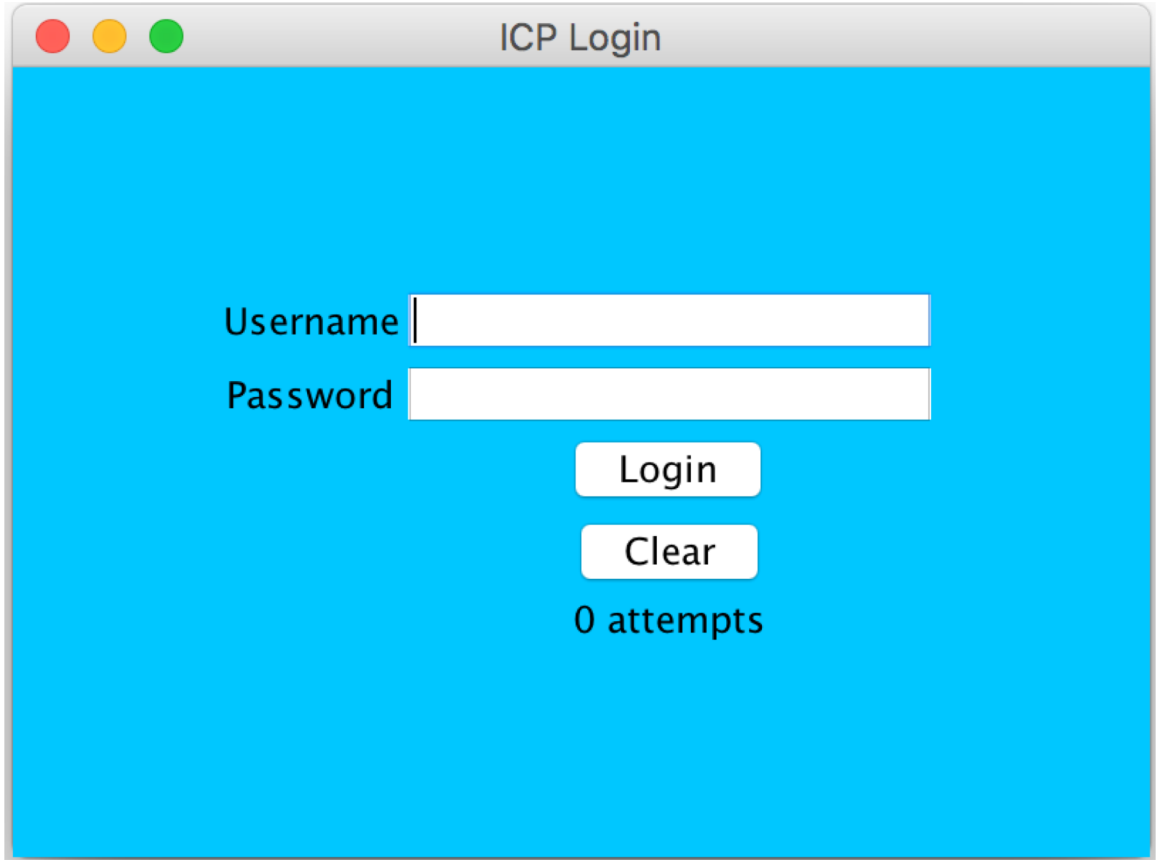

Demo
Message
Hello, you just clicked the button.
OK

PREVIOUS GUI



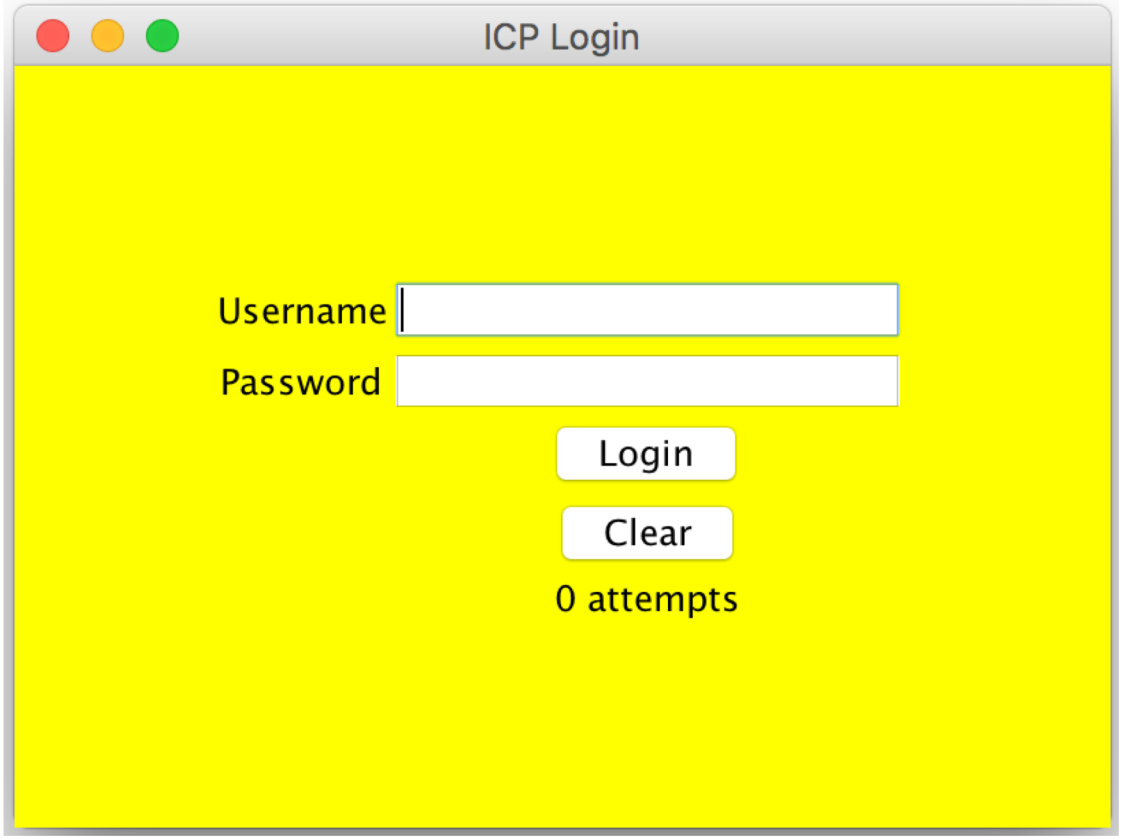
CHANGING BACKGROUND COLOR USING RGB

```
JPanel panel = new JPanel();
panel.setBackground(new Color( r: 10, g: 200, b: 255));
```



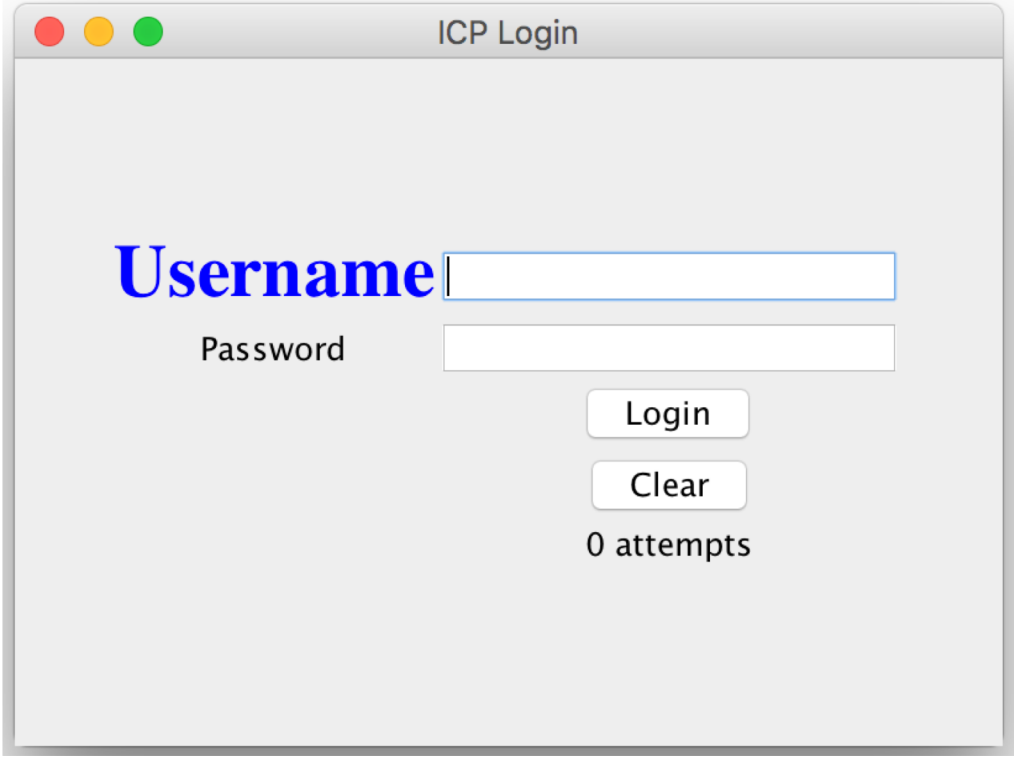
CHANGING BACKGROUND COLOR USING COLOR

```
JPanel panel = new JPanel();
panel.setBackground(Color.YELLOW);
```



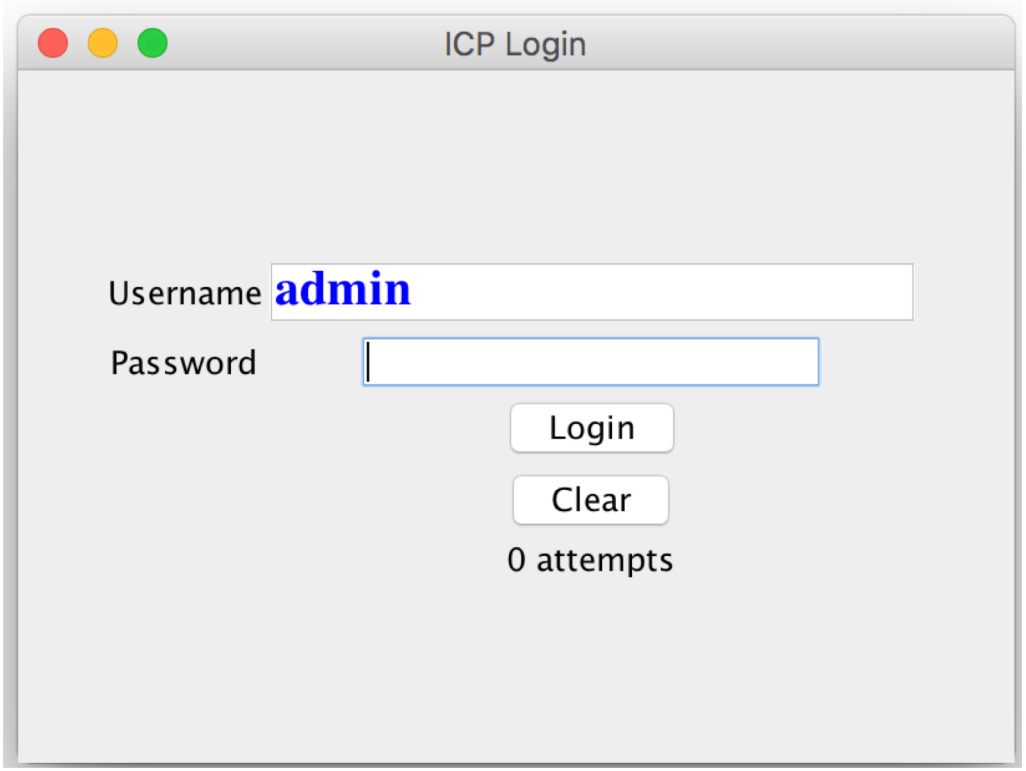
CHANGING LABEL'S FONT STYLE

```
JLabel usernameLabel = new JLabel( text: "Username");
usernameLabel.setFont(new Font( name: "Serif",Font.BOLD, size: 30));
usernameLabel.setForeground(Color.BLUE);
```



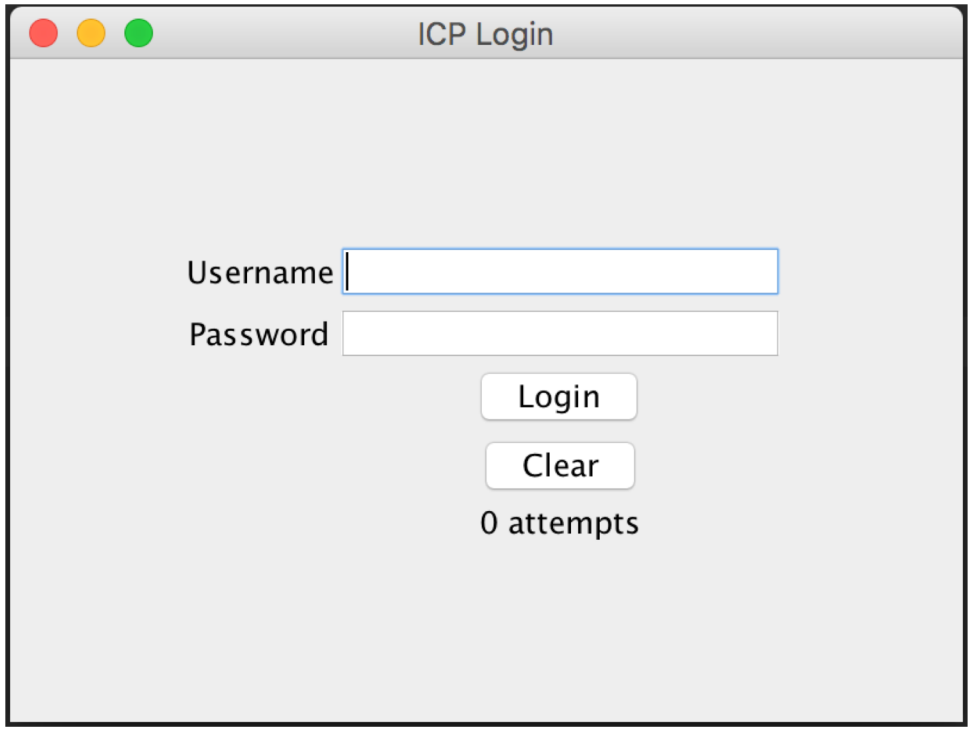
OUTPUT

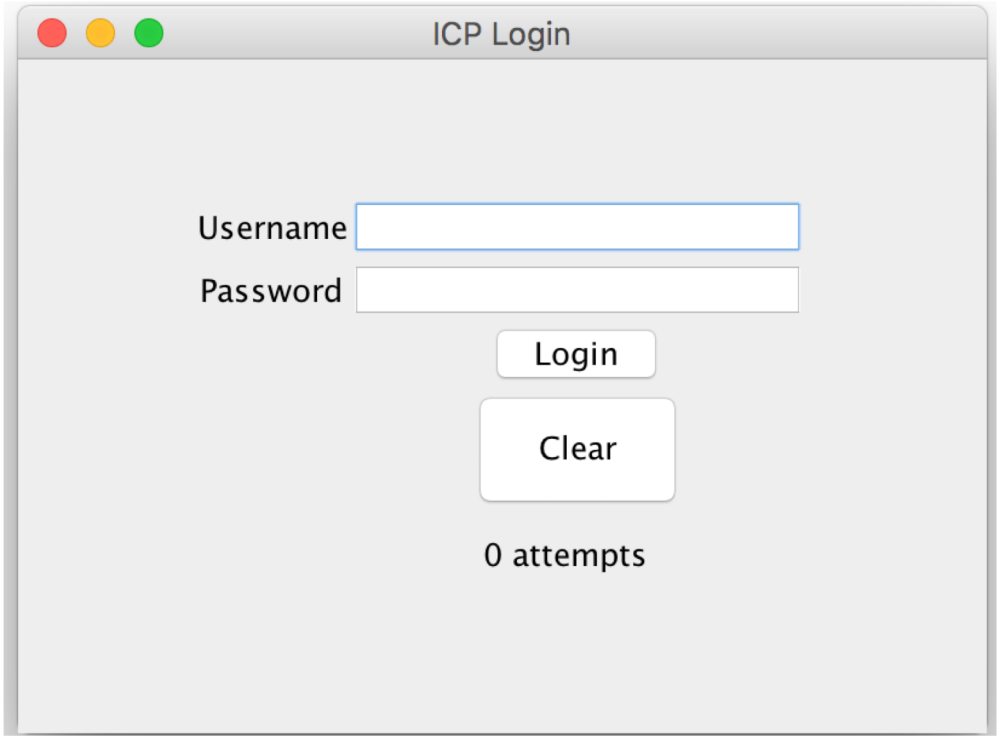
```
JTextField usernameTf = new JTextField( columns: 15);
usernameTf.setFont(new Font( name: "Serif",Font.BOLD, size: 20));
usernameTf.setForeground(Color.BLUE);
```



PADDING

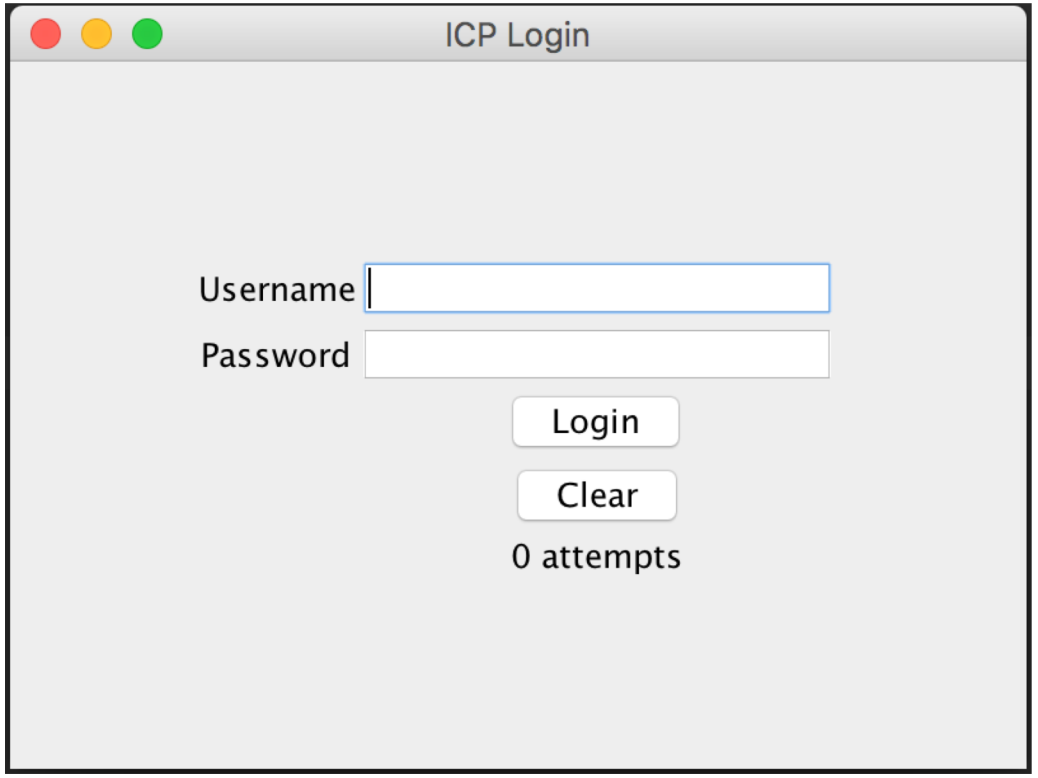
```
gbc.ipadx = 10;
gbc.ipady = 20;
panel.add(clearButton, gbc);
```

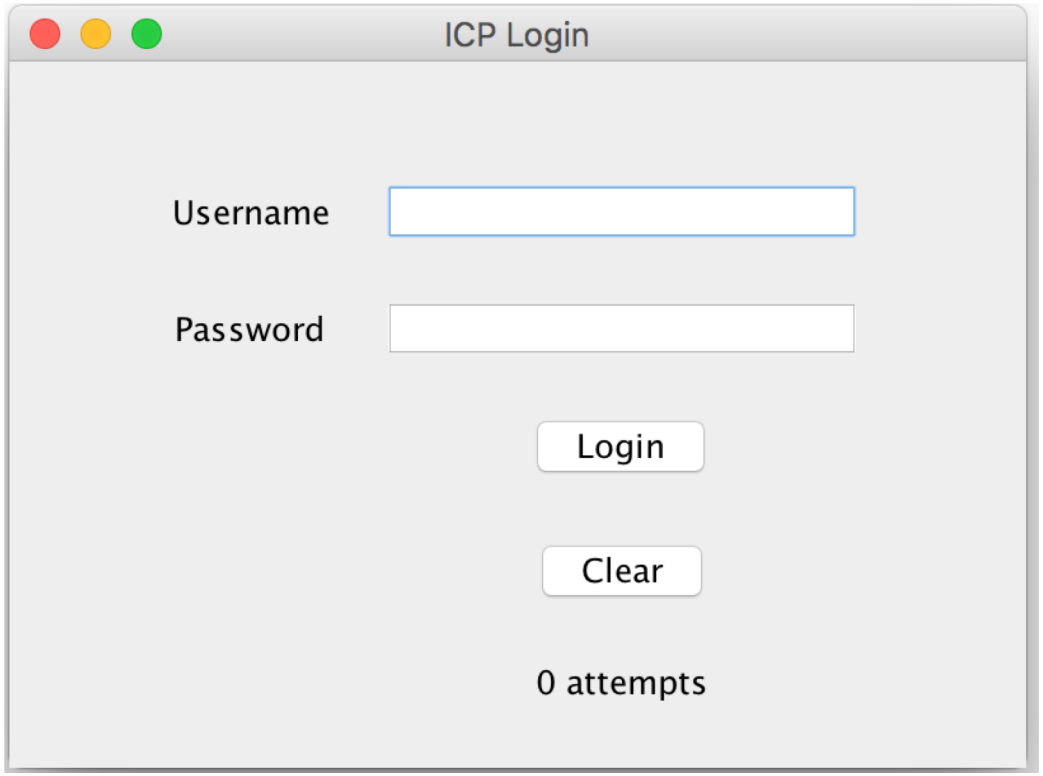




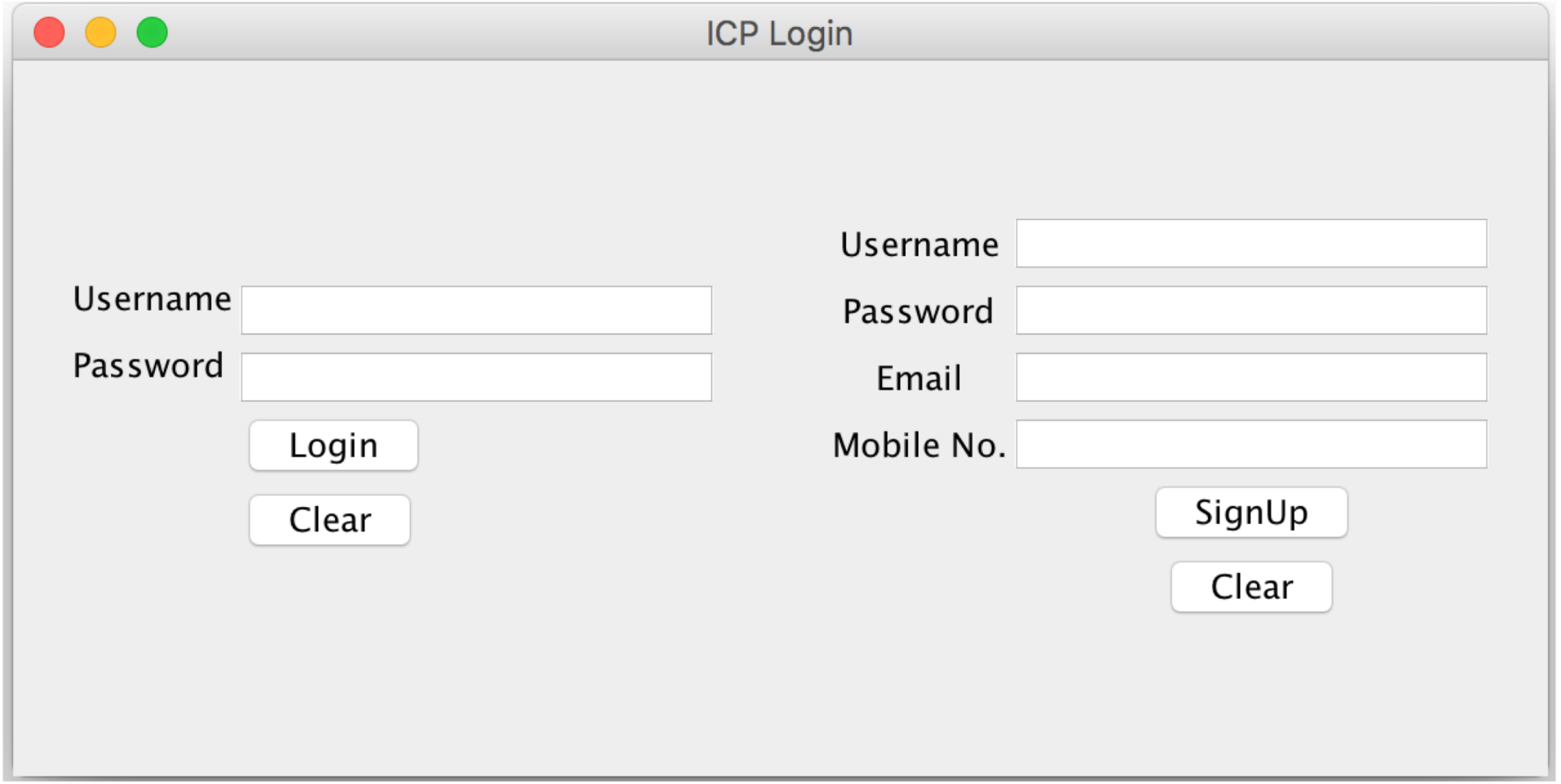
OUTPUT

```
gbc.insets = new Insets( top: 10,  left: 10,  bottom: 10,  right: 10);
```

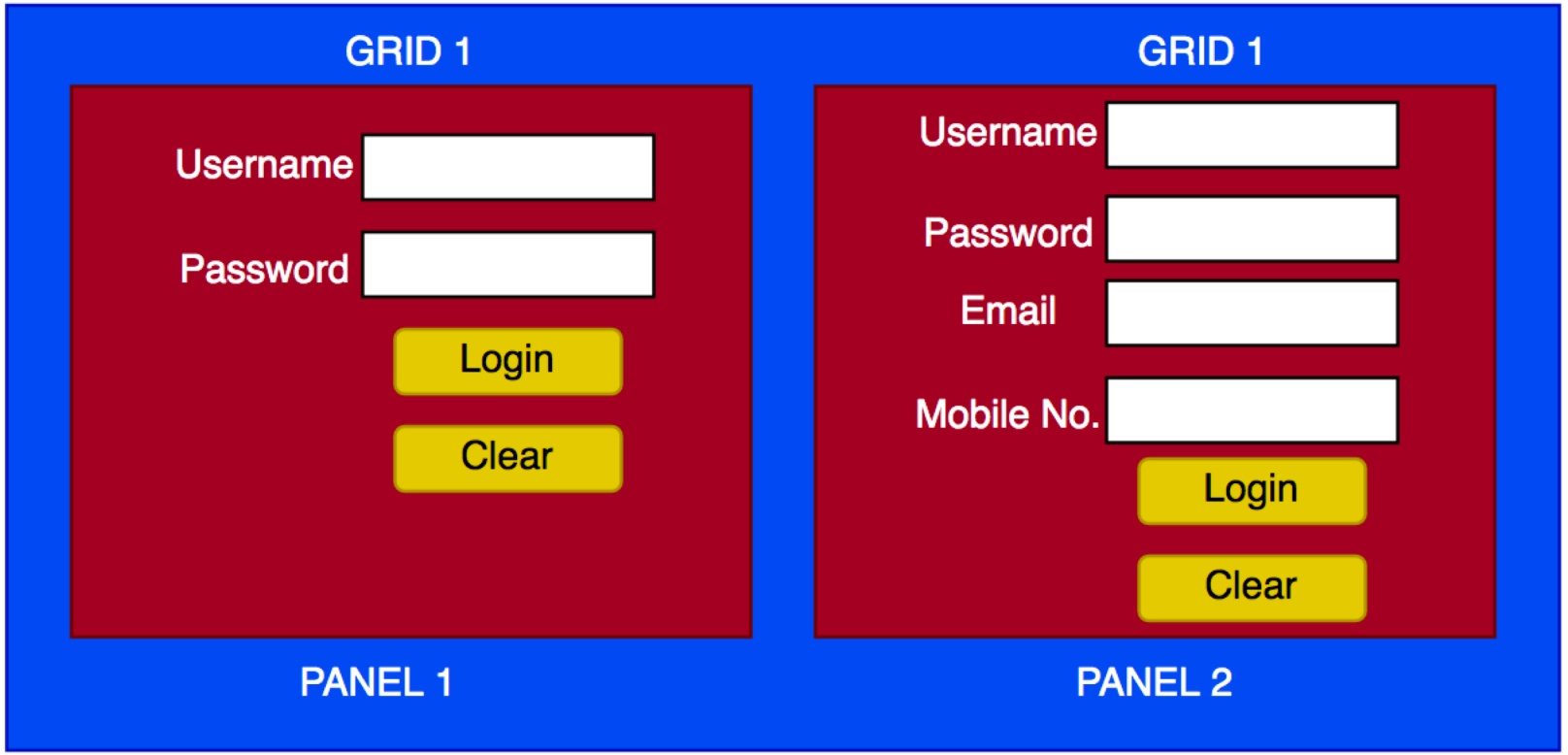




CREATING TWO PANELS



LAYOUT STRUCTURE



- Frame uses GridLayout with 2 Grids i.e. 1 row with 2 columns
- Panel 1 and Panel 2 are added in the frame (in two grids)
- Panel 1 and panel 2 are arranged using GridBagLayout

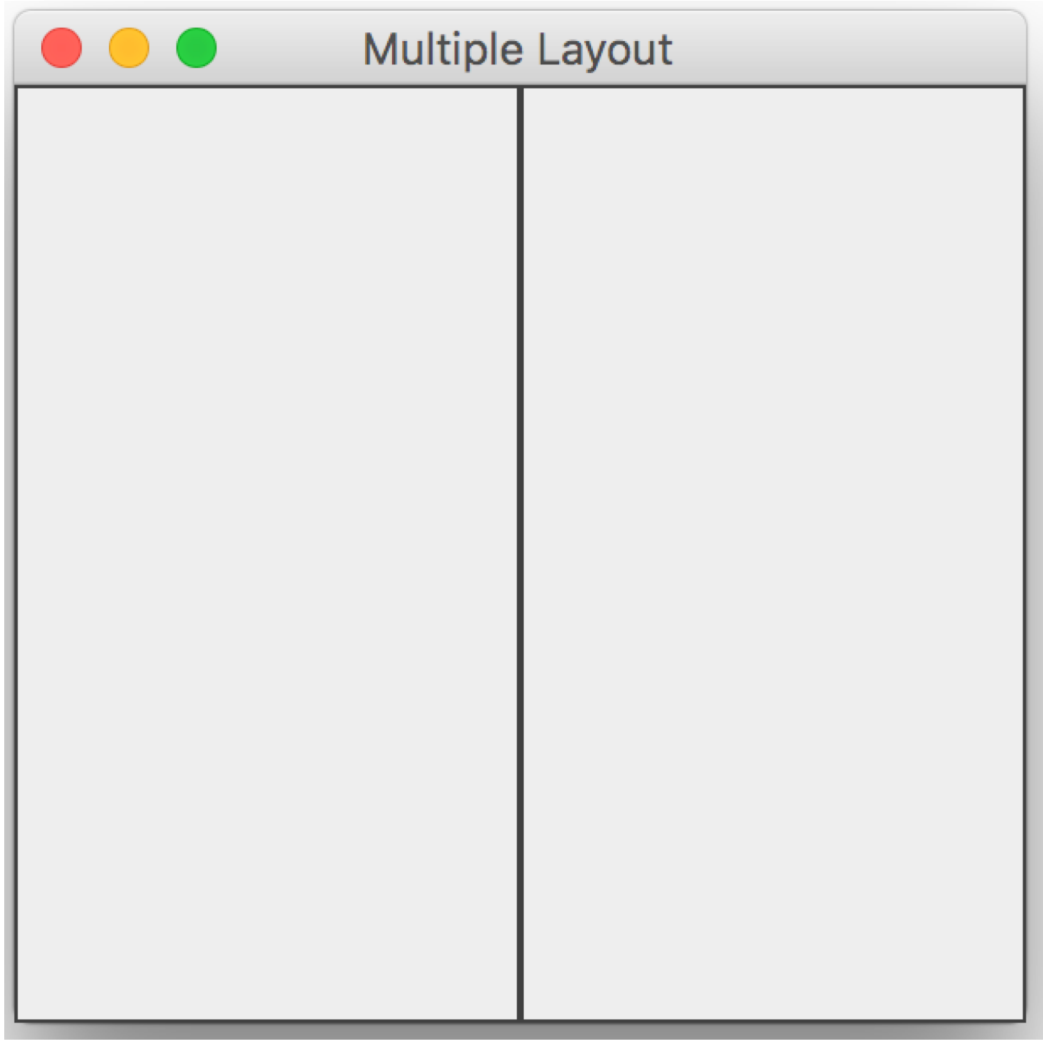
Frame

Panel

SAMPLE CODE

```
JFrame frame = new JFrame( title: "Multiple Layout");
frame.setLayout(new GridLayout( rows: 1,  cols: 2));

JPanel panel1 = new JPanel();
panel1.setLayout(new GridBagLayout());
panel1.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY));

JPanel panel2 = new JPanel();
panel2.setLayout(new GridBagLayout());
panel2.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY));

frame.add(panel1);
frame.add(panel2);

frame.setSize( width: 300,  height: 300);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
```

OUTPUT

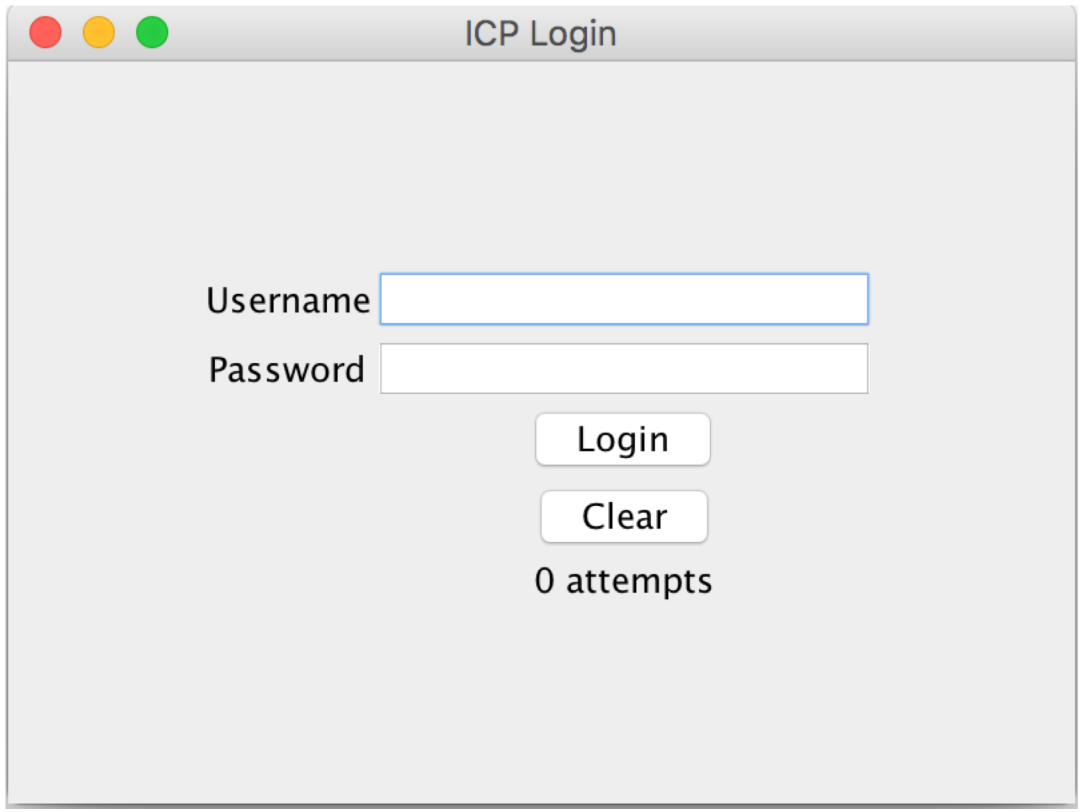


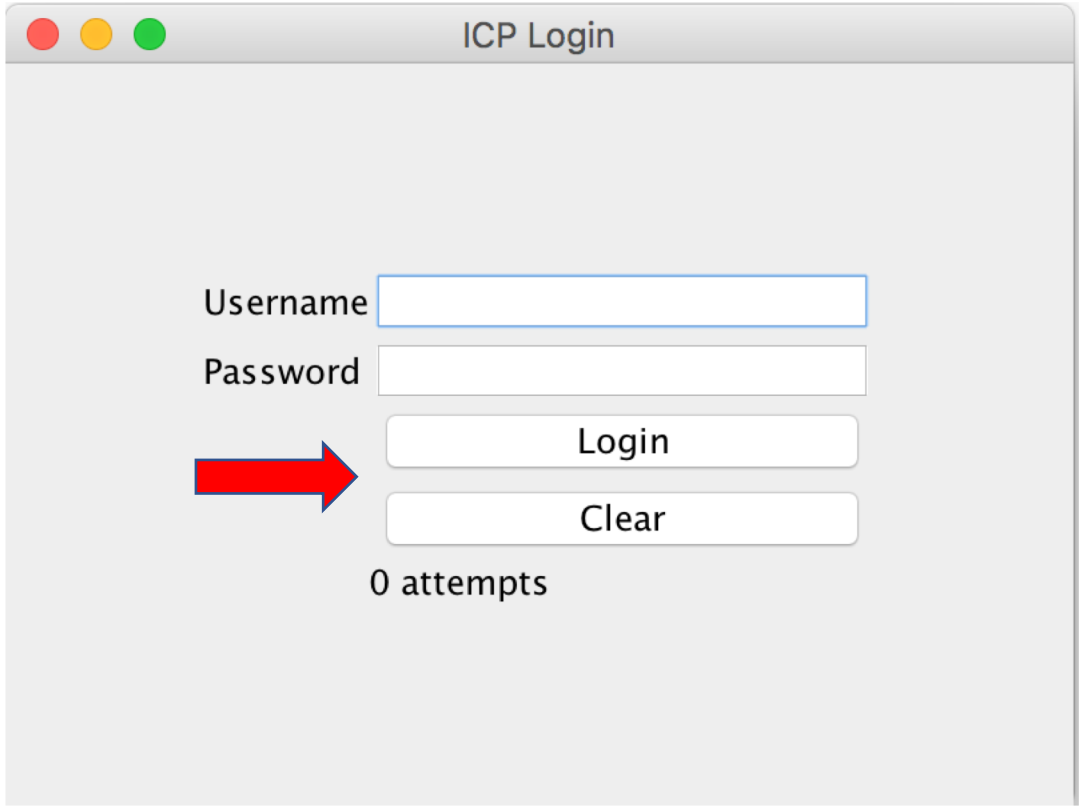
BORDER COLOR

```
JPanel panel = new JPanel();
panel.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY));
```

GRIDBAGCONSTRAINTS FILL

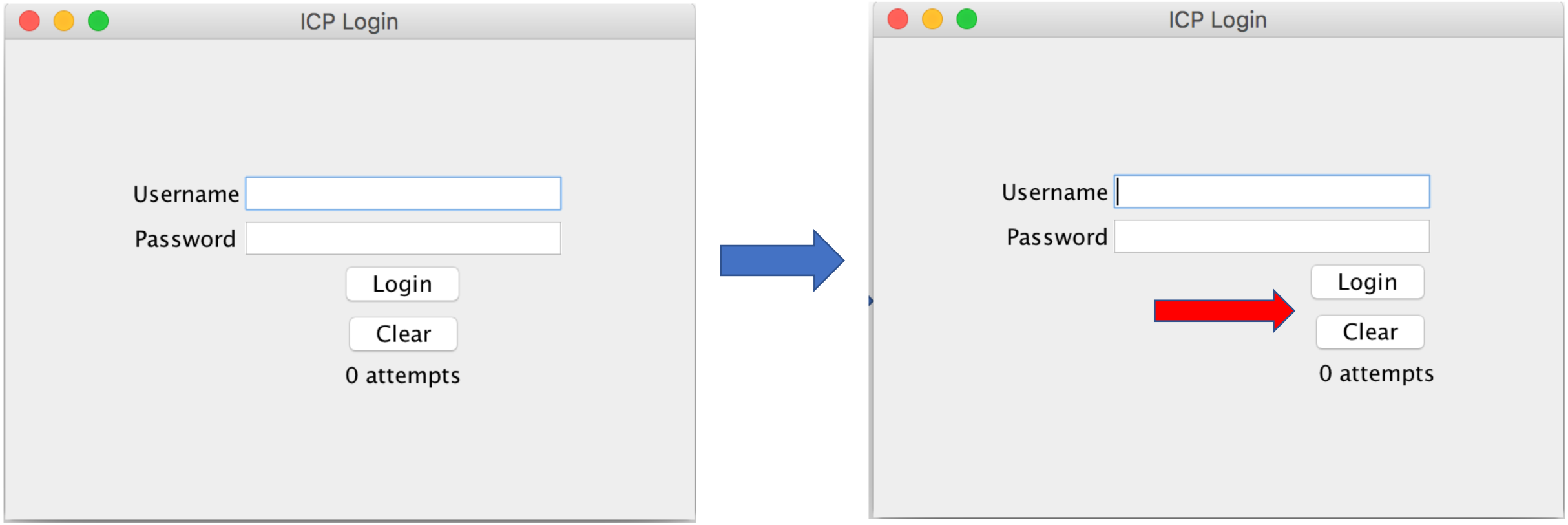
```
gbc.fill = GridBagConstraints.HORIZONTAL;
```





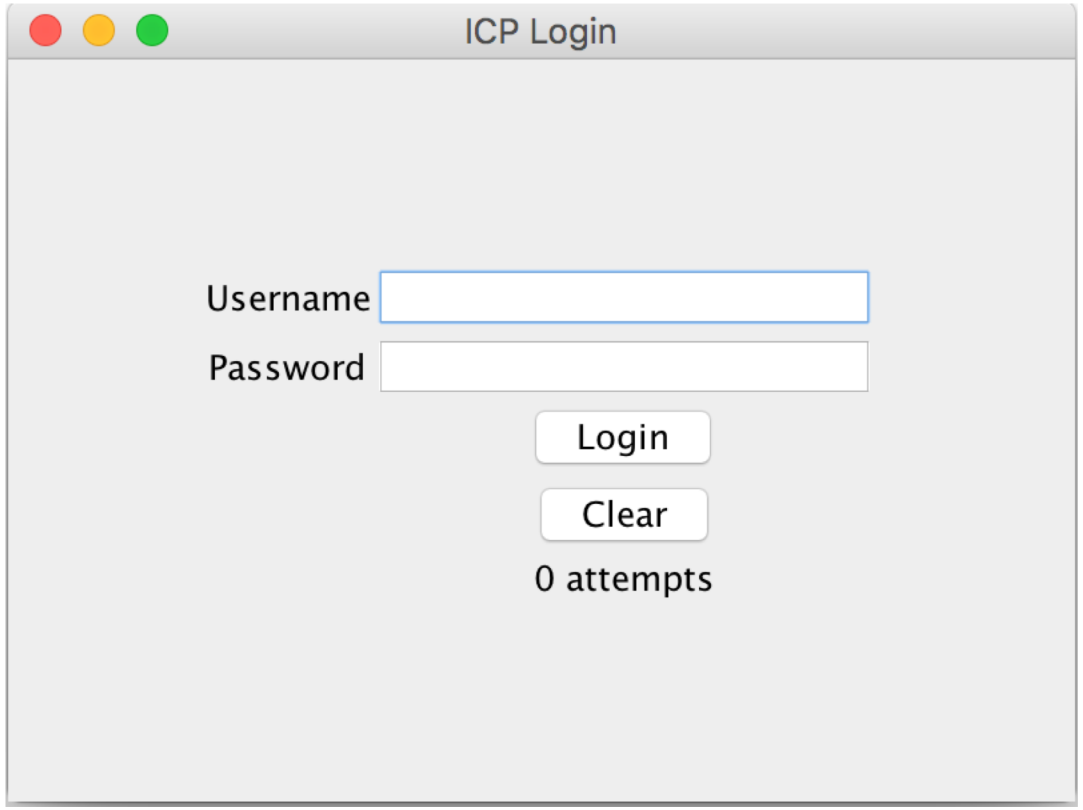
GRIDBAGCONSTRAINTS ANCHOR

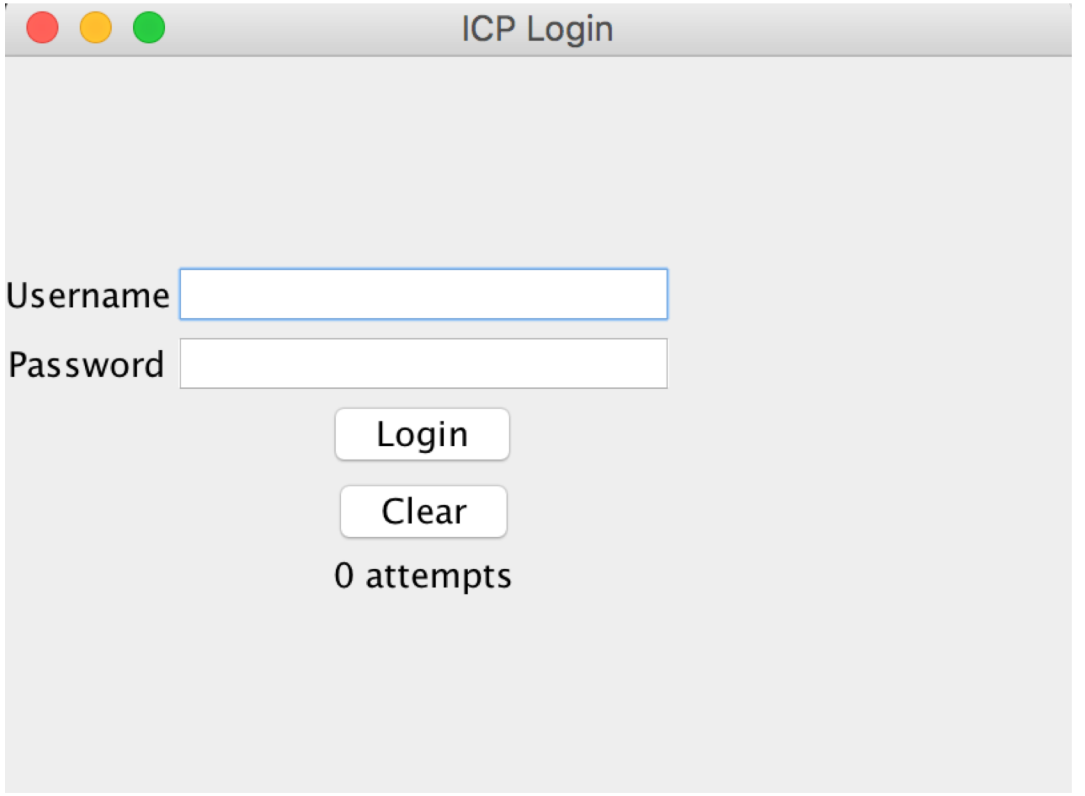
```
gbc.anchor = GridBagConstraints.EAST;
```



POSITIONING PANEL

```
frame.add(panel, BorderLayout.LINE_START);
```





THANK YOU!

Any questions?