

Java Swing

Short Notes

Java GUI Application

AONG CHO MARMA

The Publisher 

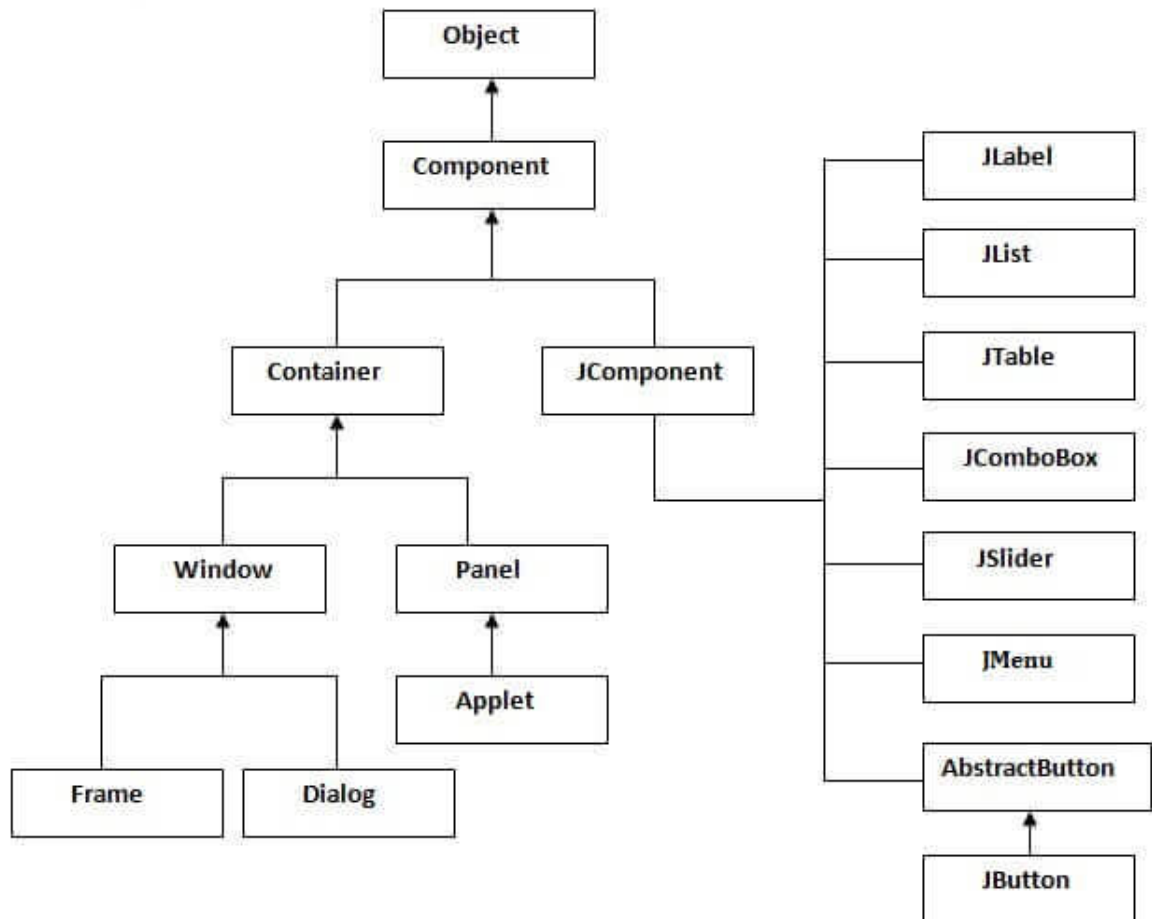
Contents

1	Swing Basics	2
1.1	Hierarchy of Java Swing Classes	2
1.2	Some Important Methods Related to JOptionPane Class	3
1.2.1	showMessageDialog() Method	3
1.2.2	JOptionPane Icons	4
1.2.3	showInputDialog() Method	5
1.2.4	showConfirmDialog() Method	6
1.3	JFrame	7
1.3.1	Basic JFrame	7
1.3.2	Frame Decoration (Icon & Background Color)	8
1.4	JLabel	9
1.4.1	Basic JLabel	9
1.4.2	JLabel Font Style	10
1.4.3	Set Tip Text	11
1.4.4	Image Label	12
1.4.5	Text Field	13
1.5	Event Listener	14
1.5.1	Read text form text field and show in messageDialog()	14
1.5.2	Perform EventListener for JTextField and Show Message Using showMessageDialog	15
1.5.3	Password Field	17
1.5.4	Button Field	18
1.6	Drawing Shapes	19
1.7	Set Cursor	20
2	Latex Ticks	22
2.1	Figure Overlap On The Plain Text	22
2.2	More Control over the Figure	23

Chapter 1

Swing Basics

1.1 Hierarchy of Java Swing Classes



1.2 Some Important Methods Related to JOptionPane Class

Methods Name	Description
showMessageDialog()	Tell the user about something that has happened.
showInputDialog()	Prompt for some input.
showOptionDialog()	The Grand Unification of the above three.
showConfirmDialog()	Asks a confirming question, like yes/no/cancel.

1.2.1 showMessageDialog() Method

Syntax:

```
1 JOptionPane.showMessageDialog(parameters);
```

- component, object
- component, object, String, int
- component, object, String, int, Icon





Parameters Descriptions -

- Component – The first parameter, which determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is used.
- Object – The second parameter can be any objects.
- String – The third parameter is a String placed as the title of the message dialog window.
- int – The int that follows the String is the MessageType. The different MessageTypes for JOptionPane, are:
 - ERROR_MESSAGE
 - INFORMATION_MESSAGE
 - WARNING_MESSAGE
 - QUESTION_MESSAGE
 - PLAIN_MESSAGE
- Icon – The last parameter is an Icon that is displayed inside the dialog and overrides the default MessageType icon.

Example Code:

```
1 package SwingOne;
2 import javax.swing.ImageIcon;
3 import javax.swing.JOptionPane;
4 import java.awt.Image;
5
6 class MessageDialog{
7
8     public static void main(String[] args) {
9         ImageIcon icon = new ImageIcon("D:\\Programming\\Java\\PracticeProject\\Swing\\Icons\\
10         done.png");
11         Image image = icon.getImage();
12         image = image.getScaledInstance(50, 50, Image.SCALE_SMOOTH);
13         icon = new ImageIcon(image);
14         JOptionPane.showMessageDialog(null, "Done", "Success", JOptionPane.DEFAULT_OPTION, icon);
15     }
16 }
```

1.2.2 JOptionPane Icons

Icon	Code	IDE Value
No icon	JOptionPane.PLAIN_MESSAGE	-1
	JOptionPane.ERROR_MESSAGE	0
	JOptionPane.INFORMATION_MESSAGE	1
	JOptionPane.WARNING_MESSAGE	2
	JOptionPane.QUESTION_MESSAGE	3

1.2.3 showInputDialog() Method

Example Code:

```
1 package SwingOne;
2
3 import java.awt.Image;
4 import javax.swing.ImageIcon;
5 import javax.swing.JOptionPane;
6
7 class InputDialog{
8
9     public static void main(String[] args) {
10         ImageIcon icon = new ImageIcon("D:\\Programming\\Java\\PracticeProject\\Swing\\Icons\\
11         user_Icon.png");
12         Image img = icon.getImage();
13         img = img.getScaledInstance(50, 50, Image.SCALE_SMOOTH);
14         icon = new ImageIcon(img);
15
16         String name = JOptionPane.showInputDialog(null, "User Name", "User Info", JOptionPane.
17         INFORMATION_MESSAGE);
18         int age = 0;
19
20         try {
21             int birthY = Integer.parseInt(JOptionPane.showInputDialog(null, "Birth Year: ", "User
22             Info", JOptionPane.INFORMATION_MESSAGE));
23             age = 2023-birthY;
24             JOptionPane.showMessageDialog(null, "User Name: "+name+"\nAge: "+age, "User Info",
25             JOptionPane.DEFAULT_OPTION, icon);
26         } catch (NumberFormatException e) {
27             JOptionPane.showMessageDialog(null, "Invalid Birth Year !!!", "Warning", JOptionPane.
28             WARNING_MESSAGE);
29         }
30     }
31 }
```

1.2.4 showConfirmDialog() Method

Example Code:

```
1 package SwingOne;
2
3 import java.awt.Image;
4 import javax.swing.ImageIcon;
5 import javax.swing.JOptionPane;
6
7 class ConfirmDialog{
8
9     public static void main(String[] args) {
10         int choice = JOptionPane.showConfirmDialog(null, "Do you want to Execute this Program ?"
11             , "Confirmation", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
12
13         if(choice == JOptionPane.YES_OPTION) {
14             JOptionPane.showMessageDialog(null, "Congratulations !!!", "Execute", JOptionPane.
15                 INFORMATION_MESSAGE);
16         } else {
17             JOptionPane.showMessageDialog(null, "Exit program", "Exit", JOptionPane.WARNING_MESSAGE)
18             ;
19         }
20     }
21 }
```

1.3 JFrame

1.3.1 Basic JFrame

Example Code 01:

```
1 package SwingOne;
2
3 import javax.swing.JFrame;
4
5 class JFrameExample{
6
7     public static void main(String[] args) {
8         JFrame frame = new JFrame();
9         frame.setVisible(true);
10        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        frame.setTitle("Frame Demo");
12        frame.setResizable(false);
13
14        //----- Set frame size
15        //    frame.setSize(500,300);
16
17        //----- Set frame at center
18        //    frame.setLocationRelativeTo(null);
19        //----- Custom location
20        //    frame.setLocation(150,70);
21
22        //----- Size and Location combine
23        //frame.setBounds(left, top, with, height);
24
25        frame.setBounds(150,70,500,300);
26    }
27 }
```

Example Code 02: Using Constructor.

```
1 package SwingOne;
2
3 import javax.swing.JFrame;
4
5 class JFrameExample extends JFrame{
6
7     JFrameExample() {
8         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9         setTitle("Frame Demo");
10        setBounds(150,100,350,500);
11    }
12
13    public static void main(String[] args) {
14        JFrameExample frame = new JFrameExample();
15        frame.setVisible(true);
16    }
17 }
```


1.3.2 Frame Decoration (Icon & Background Color)

Example Code :

```
1 package SwingOne;
2
3 import java.awt.Color;
4 import java.awt.Container;
5 import javax.swing.ImageIcon;
6 import javax.swing.JFrame;
7
8 class MakeFrame extends JFrame {
9
10     private static final long serialVersionUID = 1L;
11
12     private ImageIcon icon = null;
13     private Container container = null;
14
15     //----- Constructor
16     MakeFrame() {
17         setVisible(true);
18         setTitle("Frame Demo");
19         setDefaultCloseOperation(EXIT_ON_CLOSE);
20         setLocationRelativeTo(null);
21         setSize(350,500);
22         decorateFrame();          // Call decorateFrame() method
23     }
24     public void decorateFrame() {
25
26         //----- Set Frame Icon -----
27         icon = new ImageIcon(getClass().getResource("done.png"));
28         this.setIconImage(icon.getImage());
29
30         //----- Set Frame Background Color -----
31         container = this.getContentPane();
32         // container.setBackground(Color.ORANGE);
33
34         //----- Set Custom Background Color -----
35         Color bgColor = Color.decode("#e6e6fa");
36         container.setBackground(bgColor);
37     }
38 }
39
40 class JFrameExample{
41
42     public static void main(String[] args) {
43         MakeFrame frame = new MakeFrame();
44         frame.setLocationRelativeTo(null);
45     }
46 }
```

1.4 JLabel

1.4.1 Basic JLabel

```
1 package SwingTwo;
2
3 import java.awt.Color;
4 import java.awt.Container;
5 import javax.swing.ImageIcon;
6 import javax.swing.JFrame;
7 import javax.swing.JLabel;
8
9 public class JLabelDemo extends JFrame {
10
11     private static final long serialVersionUID = 1L;
12
13     private ImageIcon icon = null;
14     private Container container = null;
15     private JLabel userLabel = null, passLabel = null;
16
17     JLabelDemo() {
18         decorateFrame();
19     }
20
21     public void decorateFrame() {
22
23         container = this.getContentPane();
24         container.setLayout(null);
25
26         icon = new ImageIcon(getClass().getResource("user_Icon.png"));
27         this.setIconImage(icon.getImage());
28
29         Color bgColor = Color.decode("#e2fbfa");
30         container.setBackground(bgColor);
31
32         this.setVisible(true);
33         this.setTitle("JLabel Demo");
34         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
35         this.setSize(500, 350);
36         this.setLocationRelativeTo(null);
37
38         userLabel = new JLabel();
39         userLabel.setBounds(50, 30, 100, 30);
40         userLabel.setText("User Name: ");
41         container.add(userLabel);
42
43         passLabel = new JLabel("Password: ");
44         passLabel.setBounds(50, 60, 100, 30);
45         container.add(passLabel);
46
47     }
48 }
```

```

49 public static void main(String[] args) {
50     JLabelDemo frame = new JLabelDemo();
51 }
52
53 }

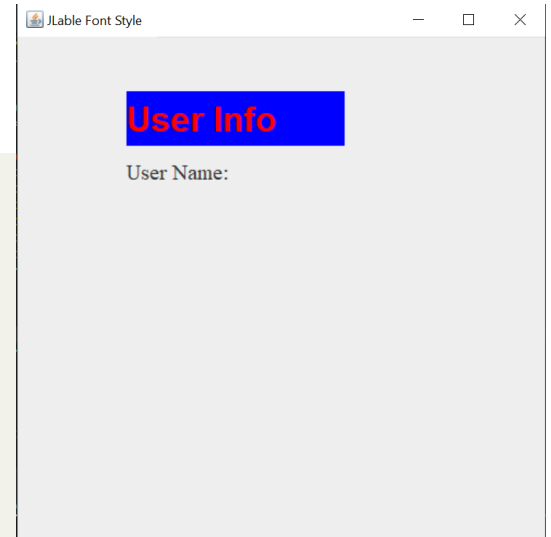
```

1.4.2 JLabel Font Style

```

1 class MakeFrame extends JFrame {
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     private Font robotoFont = null, tnrFont=null;
6     private JLabel title = null,userName = null;
7
8     MakeFrame() {
9         frameDecoration();
10    }
11
12    public void frameDecoration() {
13
14        container = this.getContentPane();
15        container.setLayout(null);
16
17        this.setVisible(true);
18        this.setTitle("JLable Font Style");
19        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
20        this.setSize(500,500);
21        this.setLocationRelativeTo(null);
22
23        //----- Define Font Style
24        robotoFont = new Font("Roboto",Font.BOLD,32);
25        tnrFont = new Font("Times New Roman",Font.PLAIN,20);
26
27        title = new JLabel("User Info");
28        title.setBounds(200, 50, 100, 30); // x,y,width,height
29        title.setFont(robotoFont); // Set Font Style
30        title.setForeground(Color.RED); // Change Foreground Color
31        title.setOpaque(true); // Enable to change Background Color
32        title.setBackground(Color.BLUE); // Change Background Color
33        container.add(title); // Add JLabel to Container
34
35        userName = new JLabel("User Name: ");
36        userName.setFont(tnrFont);
37        userName.setBounds(100,100,100,30);
38        container.add(userName);
39
40    }
41 }

```

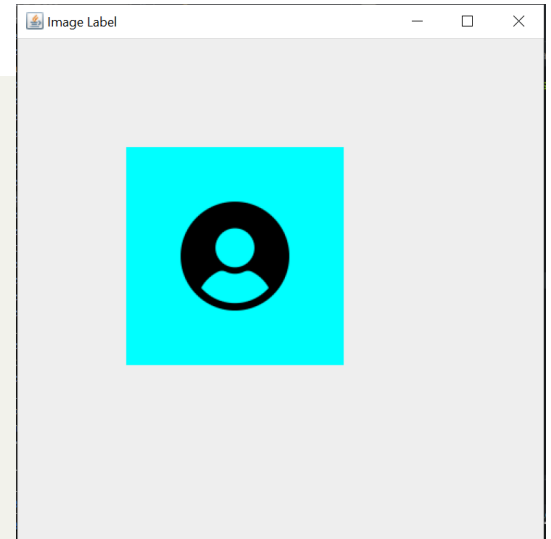


1.4.3 Set Tip Text

```
1 class MakeFrame extends JFrame {
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     private Font robotoFont = null;
6     private JLabel title = null;
7
8     MakeFrame() {
9         frameDecoration();
10    }
11
12    public void frameDecoration() {
13
14        container = this.getContentPane();
15        container.setLayout(null);
16
17        this.setVisible(true);
18        this.setTitle("JLabel Font Style");
19        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
20        this.setSize(500, 500);
21        this.setLocationRelativeTo(null);
22
23        //----- Define Font Style
24        robotoFont = new Font("Roboto", Font.BOLD, 20);
25
26        title = new JLabel("User Info");
27        title.setToolTipText("Title");    // Tool tip
28        title.setBounds(200, 50, 100, 30);
29        title.setFont(robotoFont);
30        title.setForeground(Color.RED);
31        title.setOpaque(true);
32        title.setBackground(Color.BLUE);
33        container.add(title);
34
35    }
36 }
```

1.4.4 Image Label

```
1 class MakeFramee extends JFrame {
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     public JLabel imgLabel = null;
6     private ImageIcon icon = null;
7     private Image img = null;
8
9     MakeFramee() {
10         decorateFrame();
11     }
12
13     public void decorateFrame() {
14
15         container = this.getContentPane();
16         container.setLayout(null);
17
18         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
19
20         //----- Set Image Label
21         icon = new ImageIcon(getClass().getResource("user_Icon.png"));
22         img = icon.getImage();
23         img = img.getScaledInstance(100, 100, Image.SCALE_SMOOTH); // Resize Image
24         icon = new ImageIcon(img); // Re-assign image to icon
25         imgLabel = new JLabel(icon); // Instantiate image Label
26         imgLabel.setBounds(100,100,200,200);
27         imgLabel.setOpaque(true);
28         imgLabel.setBackground(Color.cyan);
29         container.add(imgLabel); // Add image to container
30     }
31 }
32
33 public class ImageLabel {
34
35     public static void main(String[] args) {
36         MakeFramee frame = new MakeFramee();
37         frame.setVisible(true);
38         frame.setTitle("Image Label");
39         frame.setBounds(100,50,500,500);
40         frame.setLocationRelativeTo(null);
41     }
42
43 }
```



1.4.5 Text Field

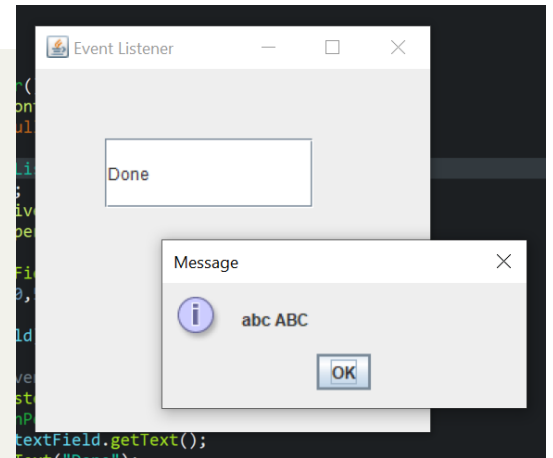
```
1 class MKFrame extends JFrame {
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     private JTextField textField = null;
6
7     MKFrame() {
8         decorateFrame();
9     }
10
11    public void decorateFrame() {
12        container = this.getContentPane();
13        container.setLayout(null);
14
15        this.setTitle("Text Field");
16        this.setSize(300,300);
17        this.setLocationRelativeTo(null);
18        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
19
20        font = new Font("Times New Roman",Font.BOLD + Font.ITALIC,18);
21
22        textField = new JTextField();
23        textField.setFont(font);
24        textField.setBounds(50,50,150,50);
25        textField.setBackground(Color.cyan);
26        textField.setForeground(Color.blue);
27        textField.setHorizontalAlignment(JTextField.CENTER); // Writing start form center
28
29        container.add(textField);
30    }
31 }
32
33 public class TextField {
34
35     public static void main(String[] args) {
36
37         MKFrame frame = new MKFrame();
38         frame.setVisible(true);
39     }
40 }
41
42 }
```



1.5 Event Listener

1.5.1 Read text form text field and show in messageDialog()

```
1 class MKFrame extends JFrame {
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     private JTextField textField = null;
6
7     MKFrame() {
8         actionListener();
9     }
10
11    public void actionListener() {
12        container = this.getContentPane();
13        container.setLayout(null);
14
15        this.setTitle("Event Listener");
16        this.setSize(300,300);
17        this.setLocationRelativeTo(null);
18        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
19
20        textField = new JTextField();
21        textField.setBounds(50,50,150,50);
22
23        container.add(textField);
24
25        //----- Event Listener
26        textField.addActionListener(new ActionListener() {
27            public void actionPerformed(ActionEvent e) {
28                String msg = textField.getText(); // Get message from textField
29                textField.setText("Done");        // Replace text with "Done" msg
30                if(!msg.isEmpty()) {
31                    JOptionPane.showMessageDialog(null, msg);
32                } else {
33                    JOptionPane.showMessageDialog(null, "Empty Message !");
34                }
35            }
36        });
37    }
38 }
39
40 public class TextField {
41     public static void main(String[] args) {
42
43         MKFrame frame = new MKFrame();
44         frame.setVisible(true);
45     }
46 }
```



1.5.2 Perform ActionListener for JTextField and Show Message Using showMessage-Dialog

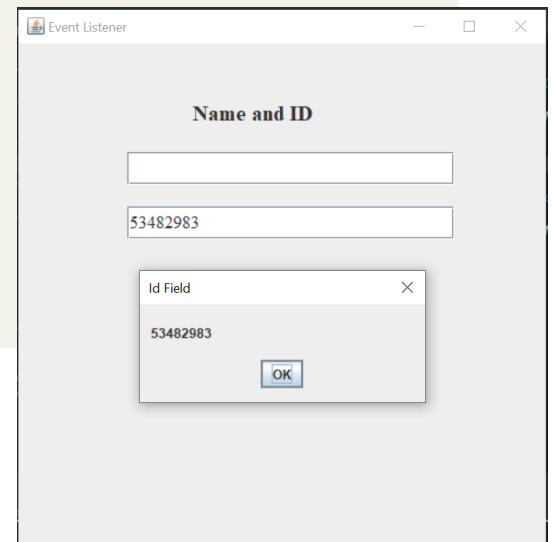
```
1 class MakeFrame extends JFrame {
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     private JLabel heading = null;
6     private JTextField nameField = null, IdField = null;
7     private Font font = null;
8
9     MakeFrame(int width, int height) {
10         this.setTitle("Event Listener");
11         this.setSize(width,height);
12         this.setLocationRelativeTo(null);
13         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
14         frameActions();
15     }
16
17     public void frameActions() {
18         container = this.getContentPane();
19         container.setLayout(null);
20
21         font = new Font("Times New Roman",Font.BOLD,20);
22
23         // Crate and add JLabel to the container
24         heading = new JLabel("Name and ID");
25         heading.setFont(font);
26         heading.setBounds(160,50,150,30);
27         container.add(heading);
28
29         font = new Font("Times New Roman",Font.PLAIN,16);
30
31         // Configure and add JTextField to the Container
32         nameField = new JTextField();
33         nameField.setBounds(100,100,300,30);
34         nameField.setFont(font);
35         container.add(nameField);
36
37         IdField = new JTextField();
38         IdField.setBounds(100,150,300,30);
39         IdField.setFont(font);
40         container.add(IdField);
41
42         // Instantiate EventHandler Class and pass the JTextFields as parameter
43         EventHandler eventHandler = new EventHandler(nameField, IdField);
44
45         nameField.addActionListener(eventHandler);
46         IdField.addActionListener(eventHandler);
47     }
48 }
49
```



```

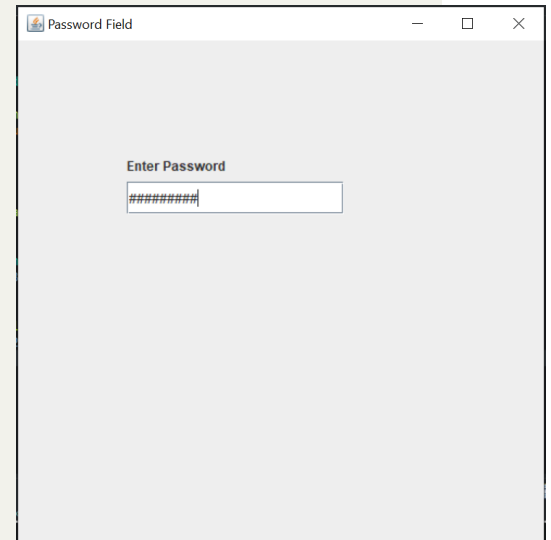
50 // Implements the ActionListener interface
51 class EventHandler implements ActionListener {
52
53     private JTextField nameField = null, idField = null;
54
55     EventHandler(JTextField name, JTextField Id) {
56         nameField = name;
57         idField = Id;
58     }
59
60     public void actionPerformed(ActionEvent e) {
61         String str = null;
62         if(e.getSource() == nameField) {
63             str = nameField.getText();
64             if(str.isEmpty()) {
65                 JOptionPane.showMessageDialog(null, "Empty Name Field", "Name Field", JOptionPane.
WARNING_MESSAGE);
66             } else {
67                 JOptionPane.showMessageDialog(null, str, "Name Field", JOptionPane.PLAIN_MESSAGE);
68             }
69         } else {
70             str = idField.getText();
71             if(str.isEmpty()) {
72                 JOptionPane.showMessageDialog(null, "Empty Id Field", "Id Field", JOptionPane.
WARNING_MESSAGE);
73             } else {
74                 JOptionPane.showMessageDialog(null, str, "Id Field", JOptionPane.PLAIN_MESSAGE);
75             }
76         }
77     }
78 }
79
80 public class EventListener1 {
81
82     public static void main(String[] args) {
83         MakeFrame frame = new MakeFrame(500,500);
84         frame.setVisible(true);
85     }
86
87 }

```



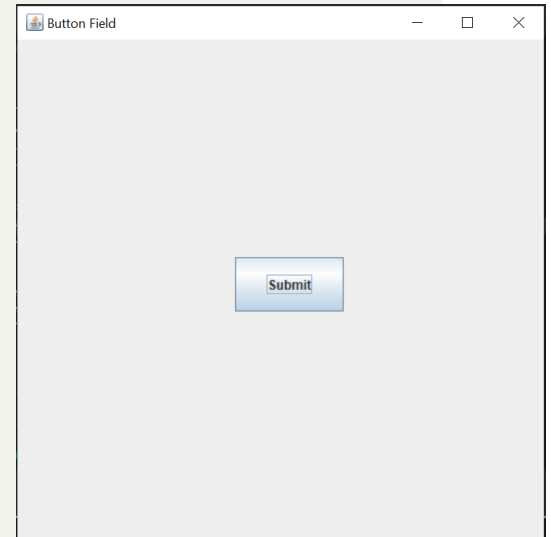
1.5.3 Password Field

```
1 class CreatFrame extends JFrame{
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     private JPasswordField passField = null;
6
7     CreatFrame() {
8         this.setTitle("Password Field");
9         this.setSize(500,500);
10        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
11        this.setLocationRelativeTo(null);
12        frameActions();
13    }
14
15    public void frameActions() {
16        container = this.getContentPane();
17        container.setLayout(null);
18
19        JLabel label = new JLabel("Enter Password");
20        label.setBounds(100,100,200,30);
21        container.add(label);
22
23        passField = new JPasswordField();
24        passField.setBounds(100,130,200,30);
25        passField.setEchoChar('#');
26        container.add(passField);
27
28    }
29 }
30
31 public class PasswordField {
32     public static void main(String[] args) {
33         CreatFrame frame = new CreatFrame();
34         frame.setVisible(true);
35     }
36 }
```



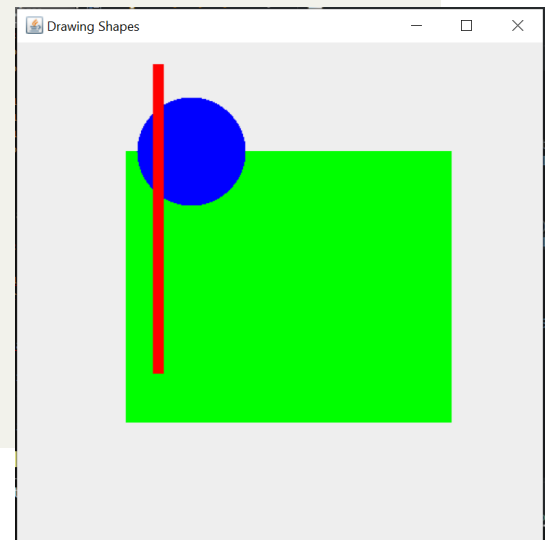
1.5.4 Button Field

```
1 class BuildFrame extends JFrame {
2     private static final long serialVersionUID = 1L;
3
4     private Container container = null;
5     private JButton button = null;
6
7     BuildFrame () {
8         this.setTitle("Button Field");
9         this.setSize(500,500);
10        this.setLocationRelativeTo(null);
11        frameActions();
12    }
13
14    public void frameActions() {
15        container = this.getContentPane();
16        container.setLayout(null);
17
18        button = new JButton("Submit");
19        button.setSize(100,50);
20        button.setLocation(200,200);
21        container.add(button);
22    }
23 }
24
25 public class ButtonField {
26
27     public static void main(String[] args) {
28
29         BuildFrame frame = new BuildFrame();
30         frame.setVisible(true);
31     }
32 }
```



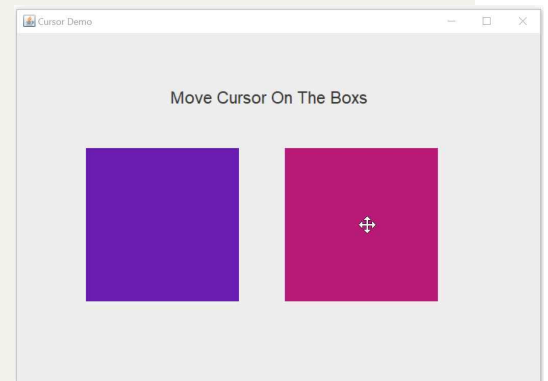
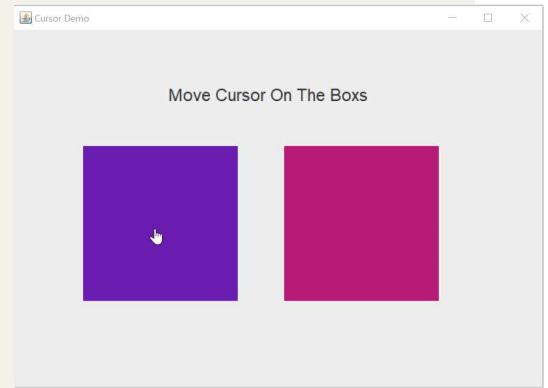
1.6 Drawing Shapes

```
1 DrawingPanel() {
2
3     this.setTitle("Drawing Shapes");
4     this.setSize(500,500);
5     this.setLocationRelativeTo(null);
6     this.setDefaultCloseOperation(EXIT_ON_CLOSE);
7
8     JPanel drawintPanel = new JPanel() {
9         public void paintComponent(Graphics g) {
10             Graphics2D g2 = (Graphics2D) g;
11
12             //Rectangle2D.Double(x,y,width,height);
13             Shape rect = new Rectangle2D.Double(100,100,300,250);
14             g2.setColor(Color.GREEN);
15             g2.fill(rect);
16
17             // Ellipse2D.Double(x,y,width,height);
18             Shape circle = new Ellipse2D.Double(110,50,100,100);
19             g2.setColor(Color.blue);
20             g2.fill(circle);
21
22             // Draw line form (x1,y2) to (x2,y2)
23             Shape line = new Line2D.Double(130,25,130,300);
24             g2.setStroke(new BasicStroke(10));
25             g2.setColor(Color.red);
26             g2.draw(line);
27         }
28     };
29
30     this.getContentPane().add(drawintPanel);
31     this.setVisible(true);
32 }
33
34
35 public class DrawingApp {
36     public static void main(String[] args) {
37         DrawingPanel frame = new DrawingPanel();
38     }
39 }
```



1.7 Set Cursor

```
1
2 package SwingTwo;
3
4 import java.awt.*;
5
6 import javax.swing.*;
7
8 public class CursorDemo extends JFrame {
9     private static final long serialVersionUID = 1L;
10
11     private Container container;
12     private JLabel heading, label1, label2;
13     private Font font;
14     private Cursor cursor1, cursor2;
15
16     CursorDemo() {
17         this.setTitle("Cursor Demo");
18         this.setSize(700,500);
19         this.setLocationRelativeTo(null);
20         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         initComponents();
22     }
23
24     public void initComponents() {
25
26         font = new Font("Arial",Font.PLAIN,22);
27         cursor1 = new Cursor(Cursor.HAND_CURSOR);
28         cursor2 = new Cursor(Cursor.MOVE_CURSOR);
29
30         container = this.getContentPane();
31         container.setLayout(null);
32
33         //----- Heading Label
34         heading = new JLabel("Move Cursor On The Boxs");
35         heading.setFont(font);
36         heading.setBounds(200,70,300,30);
37         container.add(heading);
38
39         //--- Label 1
40         label1 = new JLabel();
41         label1.setFont(font);
42         label1.setBounds(90,150,200,200);
43         label1.setOpaque(true);
44         label1.setBackground(Color.decode("#6a1cb0"));
45         label1.setCursor(cursor1); // Add cursor
46         container.add(label1);
47
48
49
```



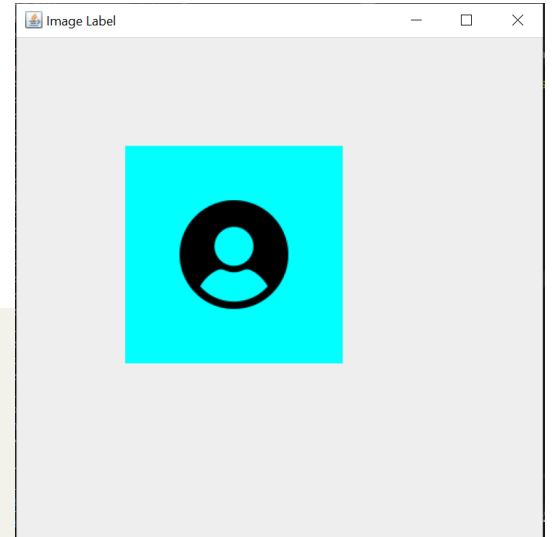
```
50
51     //--- Label 2
52     label2 = new JLabel();
53     label2.setFont(font);
54     label2.setBounds(350,150,200,200);
55     label2.setOpaque(true);
56     label2.setBackground(Color.decode("#b71b77"));
57     label2.setCursor(cursor2); // Add cursor
58     container.add(label2);
59
60 }
61
62 public static void main(String[] args) {
63
64     CursorDemo frame = new CursorDemo();
65     frame.setVisible(true);
66 }
67
68 }
```

Chapter 2

Latex Ticks

2.1 Figure Overlap On The Plain Text

```
1 class MakeFramee extends JFrame {
2
3     private Container container = null;
4     public JLabel imgLabel = null;
5     private ImageIcon icon = null;
6     private Image img = null;
7
8     MakeFramee() {
9         decorateFrame();
10    }
11
12    public void decorateFrame() {
13
14        container = this.getContentPane();
15        container.setLayout(null);
16
17        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
18
19        //----- Set Image Label
20        icon = new ImageIcon(getClass().getResource("user_Icon.png"));
21        img = icon.getImage();
22        img = img.getScaledInstance(100, 100, Image.SCALE_SMOOTH); // Resize Image
23        icon = new ImageIcon(img); // Re-assign image to icon
24        imgLabel = new JLabel(icon); // Instantiate image Label
25        imgLabel.setBounds(100,100,200,200);
26        imgLabel.setOpaque(true);
27        imgLabel.setBackground(Color.cyan);
28        container.add(imgLabel); // Add image to container
29    }
30 }
```



2.2 More Control over the Figure

```
1 DrawingPanel() {
2
3     this.setTitle("Drawing Shapes");
4     this.setSize(500,500);
5     this.setLocationRelativeTo(null);
6     this.setDefaultCloseOperation(EXIT_ON_CLOSE);
7
8     JPanel drawintPanel = new JPanel() {
9         public void paintComponent(Graphics g) {
10             Graphics2D g2 = (Graphics2D) g;
11
12             //Rectangle2D.Double(x,y,width,height);
13             Shape rect = new Rectangle2D.Double(100,100,300,250);
14             g2.setColor(Color.GREEN);
15             g2.fill(rect);
16
17             // Ellipse2D.Double(x,y,width,height);
18             Shape circle = new Ellipse2D.Double(110,50,100,100);
19             g2.setColor(Color.blue);
20             g2.fill(circle);
21
22             // Draw line form (x1,y2) to (x2,y2)
23             Shape line = new Line2D.Double(130,25,130,300);
24             g2.setStroke(new BasicStroke(10));
25             g2.setColor(Color.red);
26             g2.draw(line);
27         }
28     };
29
30     this.getContentPane().add(drawintPanel);
31     this.setVisible(true);
32 }
33 }
34
35 public class DrawingApp {
36     public static void main(String[] args) {
37         DrawingPanel frame = new DrawingPanel();
38     }
39 }
```

