Java Swing
Short Notes

Java GUI Application

AONG CHO MARMA

Contents
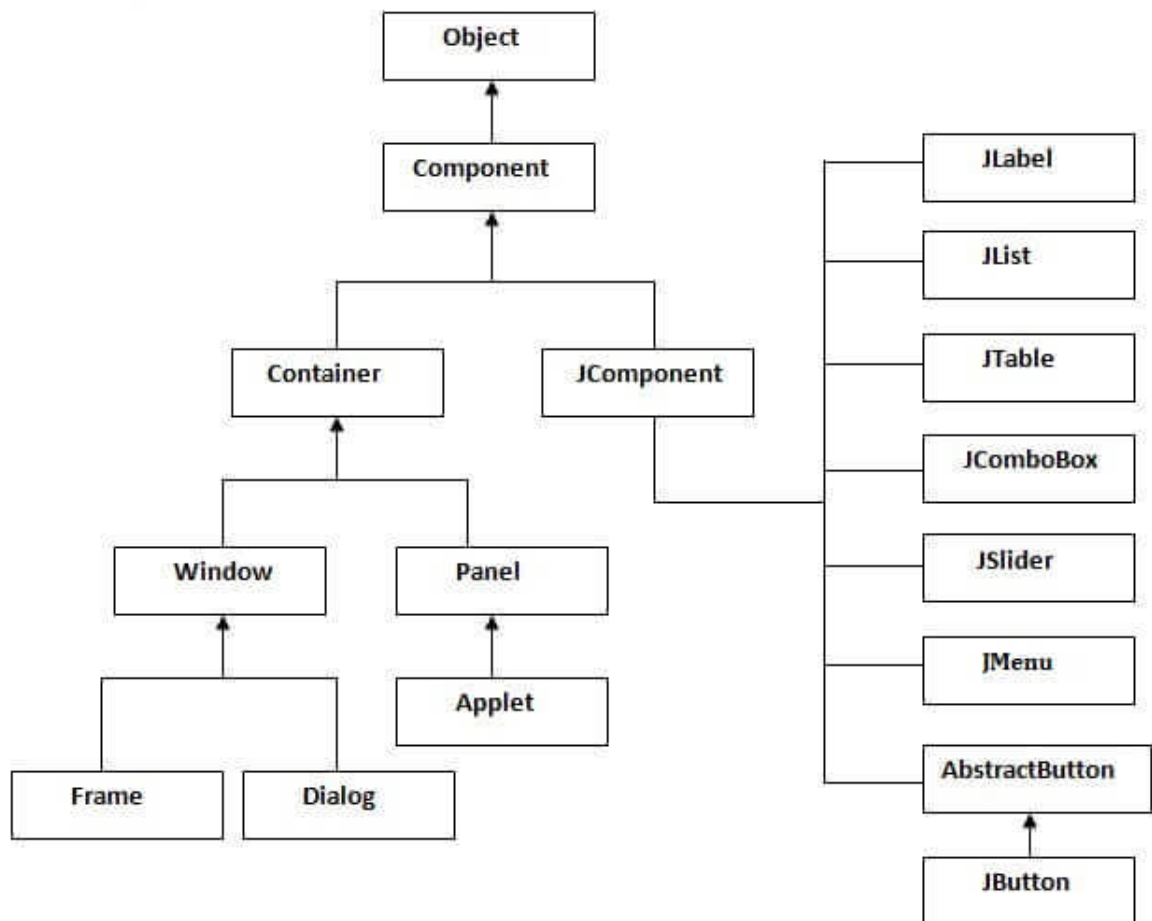
Chapter 1

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:

```
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:

```java
package SwingOne;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import java.awt.Image;

class MessageDialog{

  public static void main(String[] args) {
    ImageIcon icon = new ImageIcon("D:\\Programming\\Java\\PracticeProject\\Swing\\Icons\\
    done.png");
    Image image = icon.getImage();
    image = image.getScaledInstance(50, 50, Image.SCALE_SMOOTH);
    icon = new ImageIcon(image);
    JOptionPane.showMessageDialog(null, "Done","Success",JOptionPane.DEFAULT_OPTION, icon);
  }
}
```

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

CHAPTER 1. SWING BASICS

1.2.3 `showInputDialog()` Method

Example Code:

```java
package SwingOne;

import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

class InputDialog{

  public static void main(String[] args) {
    ImageIcon icon = new ImageIcon("D:\\Programming\\Java\\PracticeProject\\Swing\\Icons\\user_Icon.png");
    Image img = icon.getImage();
    img = img.getScaledInstance(50, 50, Image.SCALE_SMOOTH);
    icon = new ImageIcon(img);

    String name = JOptionPane.showInputDialog(null,"User Name","User Info",JOptionPane.INFORMATION_MESSAGE);
    int age = 0;

    try {
      int birthY = Integer.parseInt(JOptionPane.showInputDialog(null,"Birth Year: ","User Info",JOptionPane.INFORMATION_MESSAGE));
      age = 2023-birthY;
      JOptionPane.showMessageDialog(null,"User Name: "+name+"\nAge: "+age,"User Info",JOptionPane.DEFAULT_OPTION,icon);
    } catch(NumberFormatException e) {
      JOptionPane.showMessageDialog(null,"Invalid Bith Year !!!","Warning",JOptionPane.WARNING_MESSAGE);
    }


  }
}
```

CHAPTER 1. SWING BASICS

1.2.4 `showConfirmDialog()` Method

Example Code:

```java
package SwingOne;

import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

class ConfirmDialog{

  public static void main(String[] args) {
    int choice = JOptionPane.showConfirmDialog(null, "Do you want to Execute this Program ?"
    ,"Confirmation",JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE);

    if(choice == JOptionPane.YES_OPTION) {
      JOptionPane.showMessageDialog(null, "Congratulations !!!","Execute",JOptionPane.
    INFORMATION_MESSAGE);
    } else {
      JOptionPane.showMessageDialog(null, "Exit program","Exit",JOptionPane.WARNING_MESSAGE)
    ;
    }

  }
}
```

CHAPTER 1. SWING BASICS

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 }
```

CHAPTER 1. SWING BASICS

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
```

CHAPTER 1. SWING BASICS

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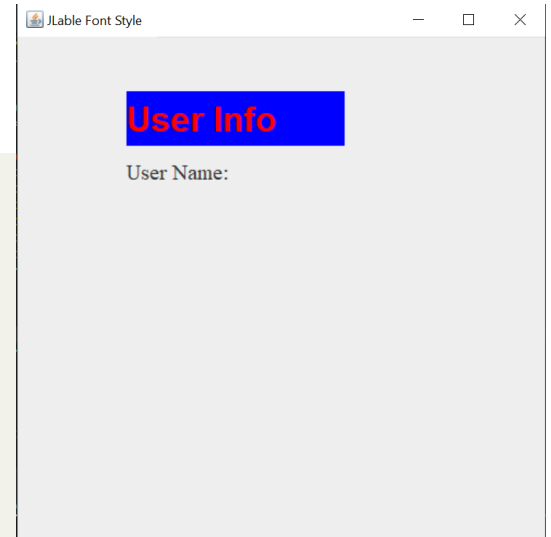
1.4.2   JLabel Font Style

.

```
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,with,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 }
```
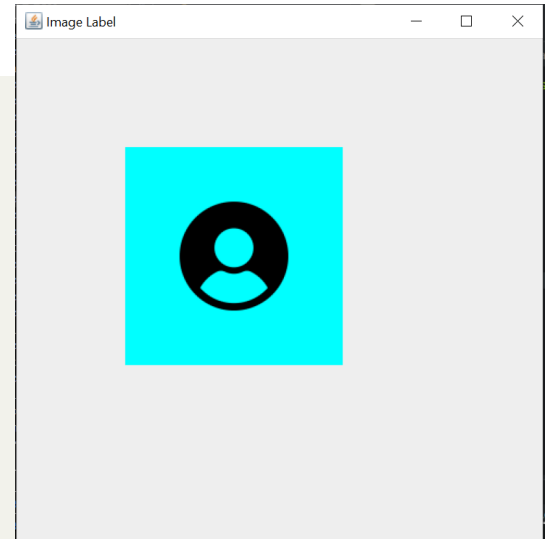
CHAPTER 1. SWING BASICS

1.4.3 Set Tip Text

```java
class MakeFrame extends JFrame {
  private static final long serialVersionUID = 1L;

  private Container container = null;
  private Font robotoFont = null;
  private JLabel title = null;

  MakeFrame() {
    frameDecoration();
  }

  public void frameDecoration() {

    container = this.getContentPane();
    container.setLayout(null);

    this.setVisible(true);
    this.setTitle("JLable Font Style");
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.setSize(500,500);
    this.setLocationRelativeTo(null);

    //------- Define Font Style
    robotoFont = new Font("Roboto",Font.BOLD,20);

    title = new JLabel("User Info");
    title.setToolTipText("Title");    // Tool tip
    title.setBounds(200, 50, 100, 30);
    title.setFont(robotoFont);
    title.setForeground(Color.RED);
    title.setOpaque(true);
    title.setBackground(Color.BLUE);
    container.add(title);

  }
}
```

CHAPTER 1. SWING BASICS

1.4.4 Image Label


```java
class MakeFramee extends JFrame {
  private static final long serialVersionUID = 1L;

  private Container container = null;
  public JLabel imgLabel = null;
  private ImageIcon icon = null;
  private Image img = null;

  MakeFramee(){
    decorateFrame();
  }

  public void decorateFrame() {

    container = this.getContentPane();
    container.setLayout(null);

    this.setDefaultCloseOperation(EXIT_ON_CLOSE);

    //----- Set Image Label
    icon = new ImageIcon(getClass().getResource("user_Icon.png"));
    img = icon.getImage();
    img = img.getScaledInstance(100, 100, Image.SCALE_SMOOTH); // Resize Image
    icon = new ImageIcon(img);           // Re-assign image to icon
    imgLabel = new JLabel(icon);         // Instantiate image Label
    imgLabel.setBounds(100,100,200,200);
    imgLabel.setOpaque(true);
    imgLabel.setBackground(Color.cyan);
    container.add(imgLabel);             // Add image to container
  }
}

public class ImageLabel {

  public static void main(String[] args) {
    MakeFramee frame = new MakeFramee();
    frame.setVisible(true);
    frame.setTitle("Image Label");
    frame.setBounds(100,50,500,500);
    frame.setLocationRelativeTo(null);
  }

}
```

CHAPTER 1. SWING BASICS
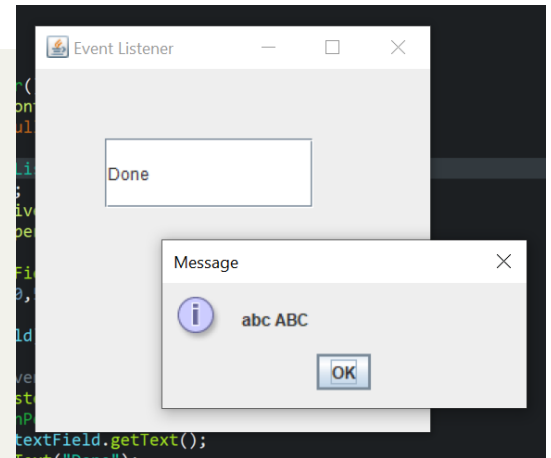
1.4.5   Text Field

.

```
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 }
```

CHAPTER 1. SWING BASICS

1.5 Event Listener

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

```java
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 EventListener for JTextField and Show Message Using showMessage-Dialog
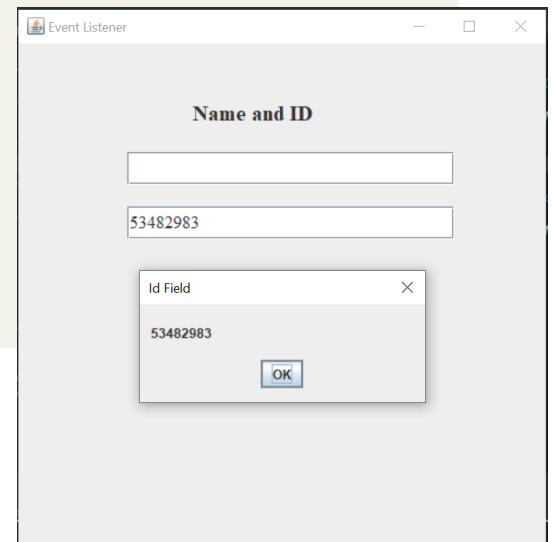
.

```
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
```
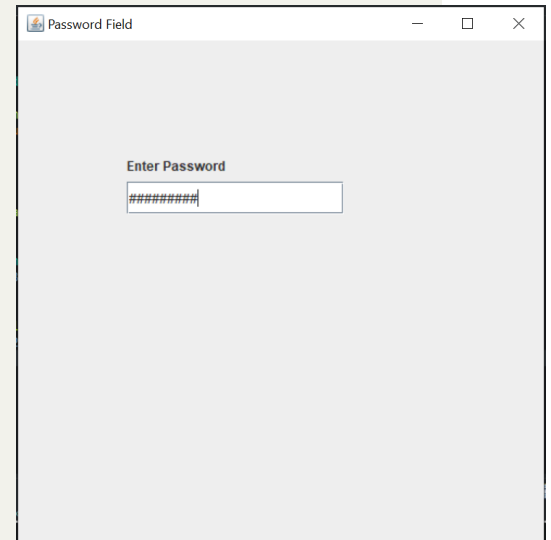
CHAPTER 1. SWING BASICS

```java
50  // Implements the ActionListner 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  }
```
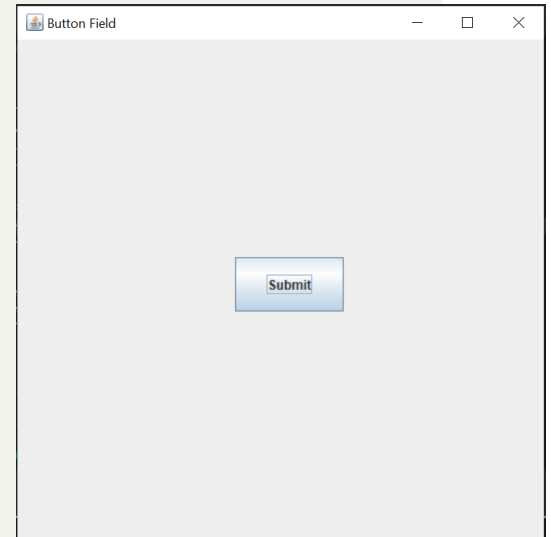
CHAPTER 1. SWING BASICS

1.5.3 Password Field

.

```java
class CreatFrame extends JFrame{
  private static final long serialVersionUID = 1L;

  private Container container = null;
  private JPasswordField passField = null;

  CreatFrame() {
    this.setTitle("Password Field");
    this.setSize(500,500);
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.setLocationRelativeTo(null);
    frameActions();
  }

  public void frameActions() {
    container = this.getContentPane();
    container.setLayout(null);

    JLabel label = new JLabel("Enter Password");
    label.setBounds(100,100,200,30);
    container.add(label);

    passField = new JPasswordField();
    passField.setBounds(100,130,200,30);
    passField.setEchoChar('#');
    container.add(passField);

  }
}

public class PasswordField {
  public static void main(String[] args) {
    CreatFrame frame = new CreatFrame();
    frame.setVisible(true);
  }
}
```
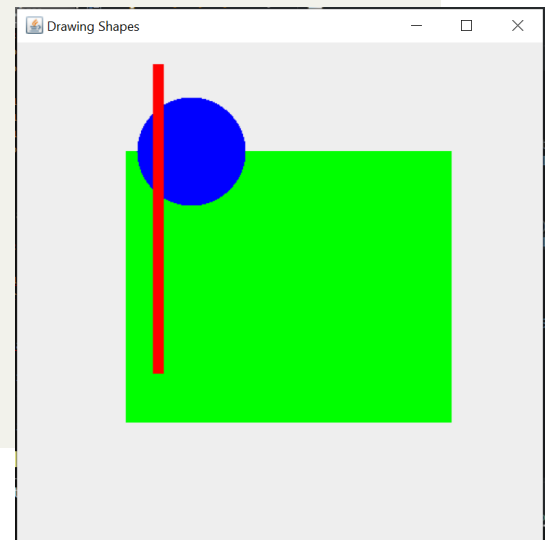
CHAPTER 1. SWING BASICS

1.5.4 Button Field

.

```java
class BuildFrame extends JFrame {
  private static final long serialVersionUID = 1L;

  private Container container = null;
  private JButton button = null;

  BuildFrame () {
    this.setTitle("Button Field");
    this.setSize(500,500);
    this.setLocationRelativeTo(null);
    frameActions();
  }

  public void frameActions() {
    container = this.getContentPane();
    container.setLayout(null);

    button = new JButton("Submit");
    button.setSize(100,50);
    button.setLocation(200,200);
    container.add(button);
  }
}

public class ButtonField {

  public static void main(String[] args) {

    BuildFrame frame = new BuildFrame();
    frame.setVisible(true);
  }
}
```

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 }
```
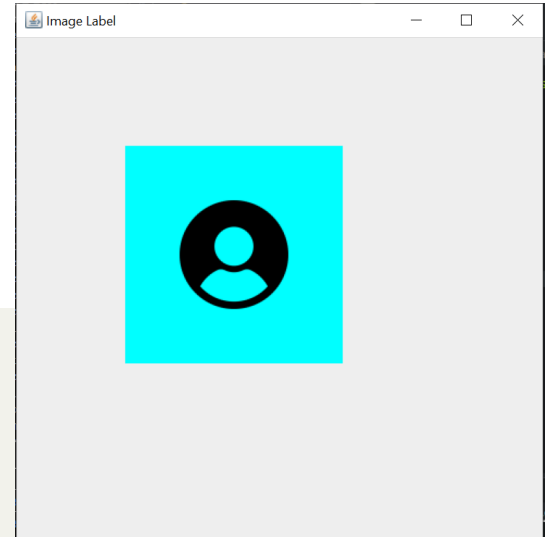
CHAPTER 1. SWING BASICS

Chapter 2

Latex Ticks

2.1   Figure Overlap On The Plain Text

.

```java
class MakeFramee extends JFrame {

  private Container container = null;
  public JLabel imgLabel = null;
  private ImageIcon icon = null;
  private Image img = null;

  MakeFramee(){
    decorateFrame();
  }

  public void decorateFrame() {

    container = this.getContentPane();
    container.setLayout(null);

    this.setDefaultCloseOperation(EXIT_ON_CLOSE);

    //----- Set Image Label
    icon = new ImageIcon(getClass().getResource("user_Icon.png"));
    img = icon.getImage();
    img = img.getScaledInstance(100, 100, Image.SCALE_SMOOTH); // Resize Image
    icon = new ImageIcon(img);            // Re-assign image to icon
    imgLabel = new JLabel(icon);         // Instantiate image Label
    imgLabel.setBounds(100,100,200,200);
    imgLabel.setOpaque(true);
    imgLabel.setBackground(Color.cyan);
    container.add(imgLabel);            // Add image to container
  }
}
```

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 }
```

CHAPTER 2. LATEX TICKS