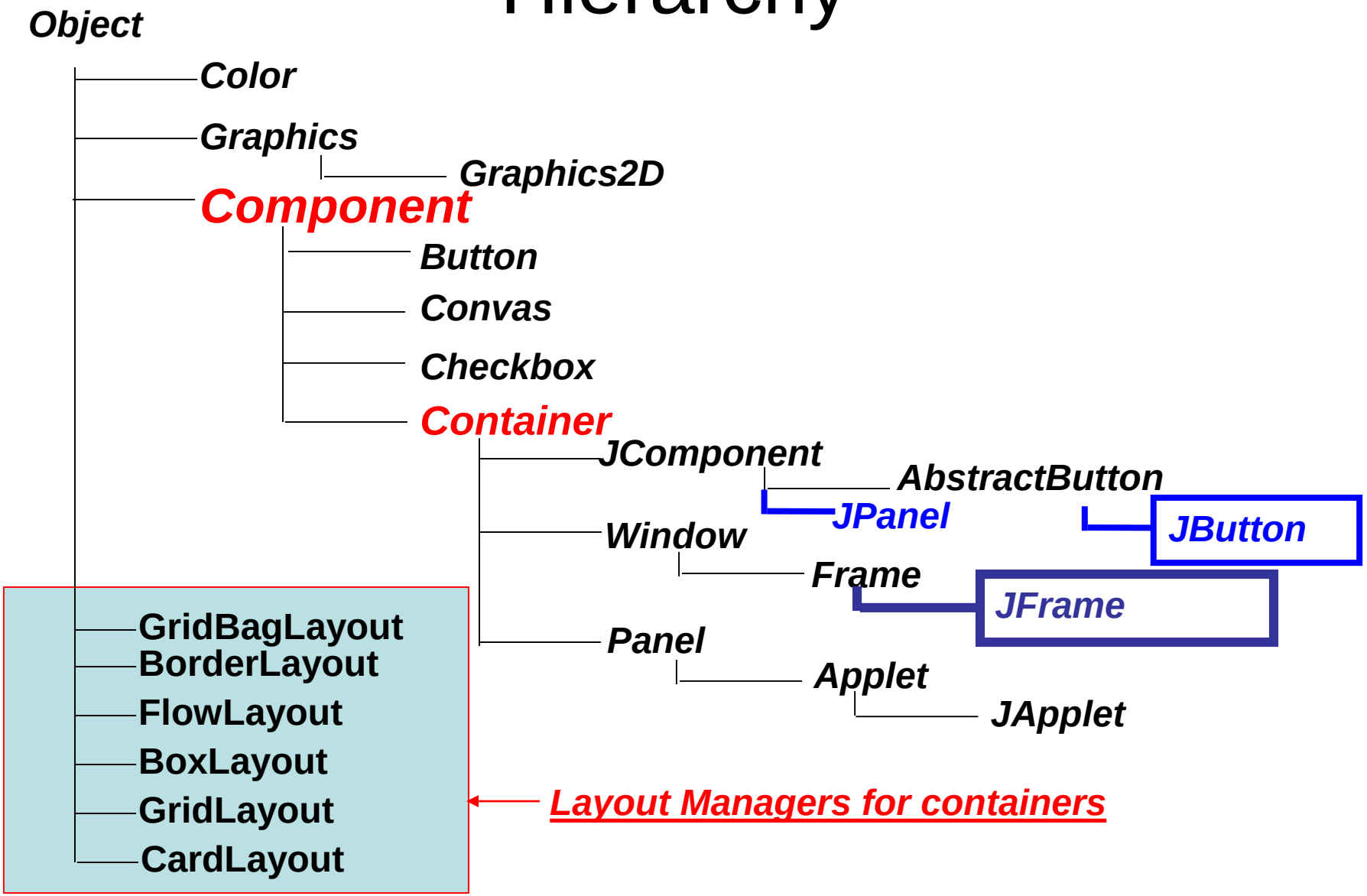
Graphics

- ***Mostly the classes graphics category are in two packages***

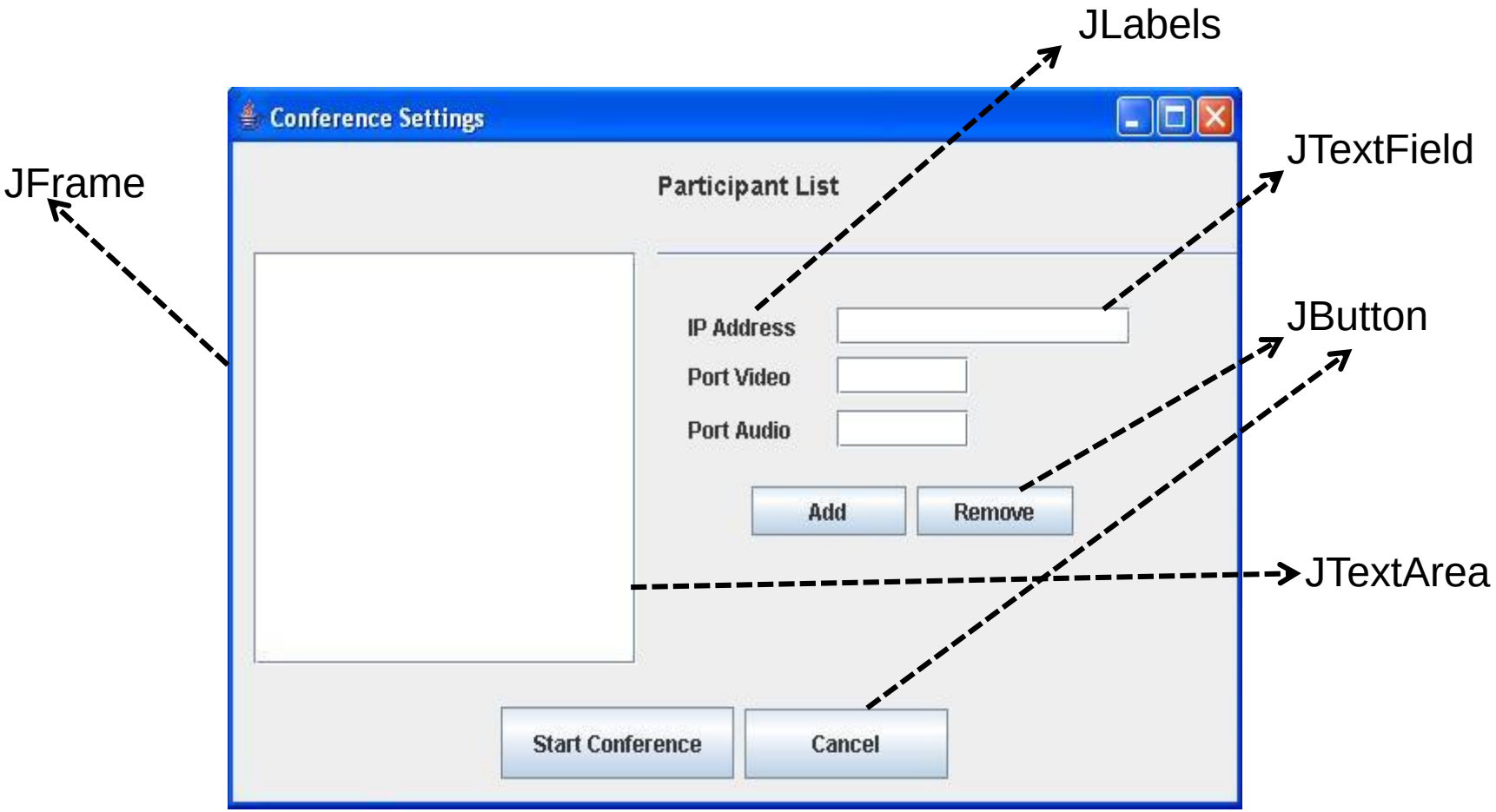
1. ***java.awt.****
2. ***javax.swing.*; [All classes in this package starts with letter J]***

- ***Graphics Programming in Java is Component oriented.***

Partial View of Graphics class Hierarchy



A Simple GUI: Example



JFrame in Java

- **An actual Window having Title Bar, Menu Bar, Borders and resizing corners.**
- **Imporatant Constructors :**

1. **JFrame()**
   Constructs a new frame that is initially invisible.
2. JFrame(String title)
   Creates a new, initially invisible Frame with the specified title.

- **Important Methods :**

1. **Container getContentPane()**
   Returns the contentPane object for this frame.
2. **void setLayout(LayoutManager manager)**
   Sets the LayoutManager.
3. **void setDefaultCloseOperation(int operation)**
4. **void setSize(FRAME_WIDTH,FRAME_HEIGHT)**
5. **void show(); → shows the frame Now Deprecated**
6. **setVisible(true/false); // Actual Method in Java 2 for Displaying Frames.**

JFrame Examples

- JFrame f1 = new JFrame(); // Creates a Frame with no title
- JFrame f2 = new JFrame(“My Frame”); // Frame with Title “MY Frame”
- f1.setSize(400,600); // sets size to 400 by 600 pixels
- f2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  (Program Terminates when Frame Window is closed)
- Adding Components to JFrame (Upto 1.4.2)
  Conatiner c1 = f1.getContentPane();
  c1.setLayout(…….);
  c1.add(<Component>);
- In Java 1.5 (No need to get ContentPane)
  f1.add(<Component>);

JPanel class in Java

- Is just like a Frame window without Borders and a Title.
- Flat panel window acts a light weight container
- Constructors

1. **JPanel**()
   Creates a new JPanel with a double buffer and a flow layout.
2. **JPanel**(LayoutManager layout)
   Create a new buffered JPanel with the specified layout manager

JLabel Objects in Java

1. **Display text or image or both in Java Graphics**
2. **Constructors :**

- **JLabel() throws HeadlessException**
- **JLabel(Icon image)**
- **JLabel(String text) throws HeadlessException**
- **Jlabel(String text, int how) throws HeadlessException [Note : ‘how’ parameter must be any of the three constants: Label.LEFT, Label.RIGHT or Label.CENTER**

3. Important Methods :

- **setIcon(Icon icon);**
- **getIcon()**
- **setText(String text);**
- **getText();**
- **setSize(width,height);**
- **setForeground(Color.red);**

Java.awt.Color class

- **Every Component can set foreground and background colors.**
  1. **public void setBackground(java.awt.Color c)**
  2. **public void setForeground(java.awt.Color c)**
- **Colors can be set by passing arguments as :**

  Color.red

  Color.black

  Color.green

Layout Managers

- **Arrangements of several components within a container is called *layout*.**
- **Conatiner's layout can be set by passing LayoutManager object to setLayout() method.**
- **Some basic Layouts :**

  1. FlowLayout :

  "Lays out components left to right and then starts from new row when there is not enough room in the current one."

  2. BOXLayout :

  " Lays out components horizontally or vertically"

  3. BorderLayout :

  This layout has five areas for laying out. NORTH,SOUTH,EAST,WEST and CENTER.

Layout Managers continued…

- **GridLayout :**

  “ Arraganges Components in a rectangular grid. All Components are resized to an identical size”.

- **GridBagLayout:**

  “ Also Arraganges Components in a grid but rows and columns can have different sizes andcomponents can span multiple rows and columns”.

JTextField class

- **A *text field* is a text component that displays editable text.**
- **Standard mechanism for input into graphics program.**
- **Constructors :**

JTextField()

JTextField(String text)

JTextField(int columns)

JTextField(String text, int columns)

Icon Interface

- Used to create and define icons.

```
public interface Icon
{
public int getIconHeight();
public int getIconWidth();
public void paintIcon(Component c, Graphics g, int x, int y)
}
```

•Graphics Parameter carries out the Drawing operation.

•To use more powerful 2D drawing operations we have to type case to Graphics2D

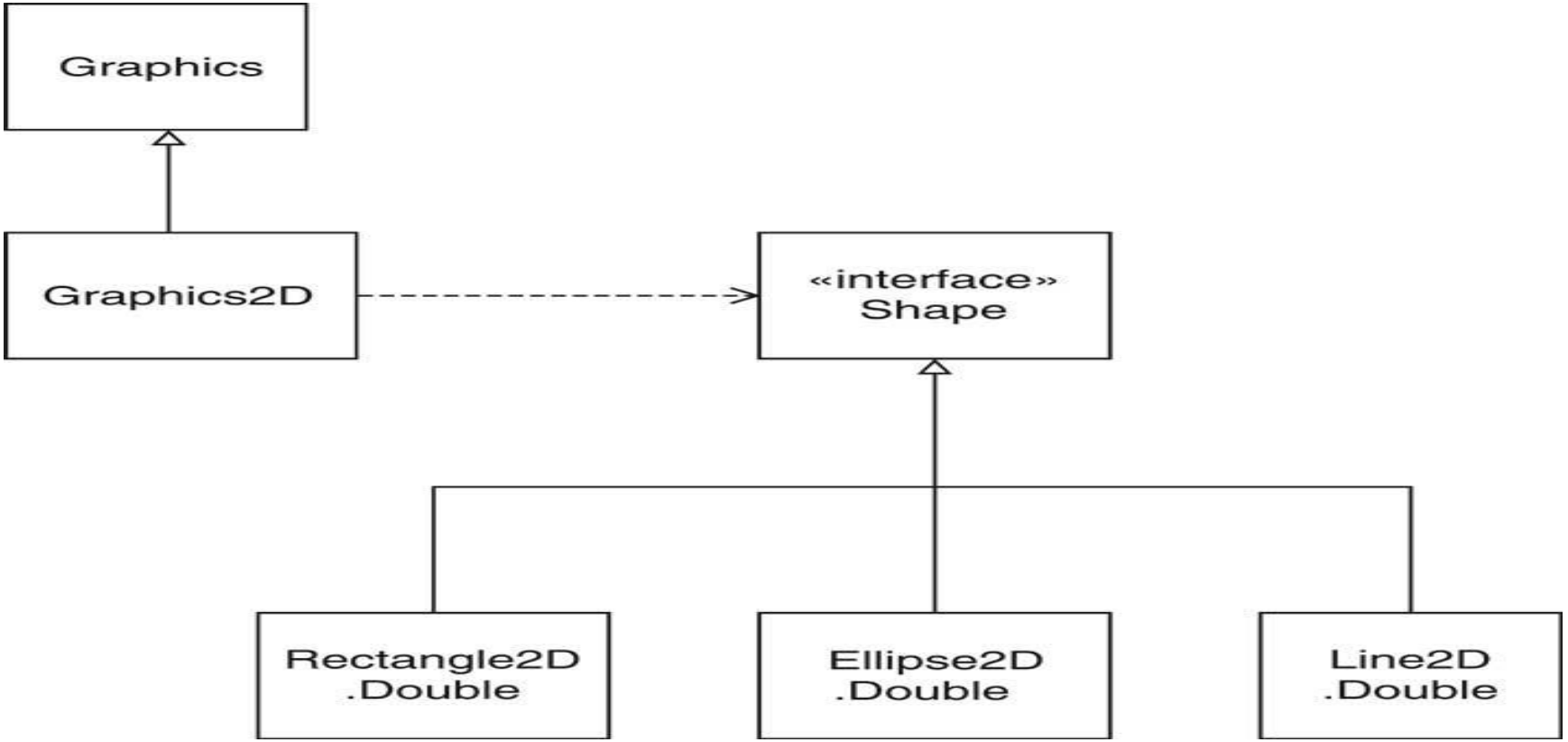
Icon interface ….

- **paintIcon method receives graphics context of type Graphics**
- **Actually a Graphics2D object in modern Java versions**

- **public void paintIcon(Component c, Graphics g, int x, int y)**
  {
  Graphics2D g2 = (Graphics2D)g;
  . . .
  }

Drawing Shapes

- We can draw objects of any class that implements ***shape*** interface



Drawing shapes ….

```
Shape s = …….
g2.draw(s);

Shape s = new Rectangle2D.Double(x,y,width,height);
g2.draw(s);  // Rectangle will be drawn

Shape s = new Ellipse2D.Double(x,y,width,height);
g2.draw(s);  // Ellipse will be drawn

Shape s = new Rectangle2D.Double(x,y,width,width);
g2.draw(s);  // Circle will be drawn
```

Drawing shapes cont….

```
Point2D.Double start =  new Point2D.Double(x1,y1);
Point2D.Double end =  new Point2D.Double(x1,y1);

Shape s1 = new Line2D.Double(start,end);
g2.draw(s1);
```

Example 1 Circle Icon

```
class circleicon implements Icon
{
private int size; // radius
circleicon(int radius) { size = radius;}
public int getIconHeight() { return size;}
public int getIconWidth()  { return size;}
public void paintIcon(Component c, Graphics g, int x,int y)
{
Graphics2D g2 = (Graphics2D) g;
Ellipse2D.Double circle = new Ellipse2D.Double(x,y,size,size);
g2.draw(circle);
}
}
```

Example 2 RectangleIcon

```
class recticon implements Icon
{
private int height;
private int width;
recticon(int height,int witdh)
{
this.height = height;
this.width = width;
}
public int getIconHeight() { return height;}
public int getIconWidth()  { return width;}
public void paintIcon(Component c, Graphics g, int x,int y)
{
Graphics2D g2 = (Graphics2D) g;
Rectangle2D.Double r1 = new Rectangle2D.Double(5,5,x-width,y-
height);
g2.draw(r1);
}
}
```

Adding a Rectangle to Frame

METHOD 1 : Make a Rectangle a Componenet

```
// A simple Frame with Rectangle Inside

import java.awt.*;
import javax.swing.*;
import java.awt.geom.*;  // For Shapes

class rectComponent extends JComponent
{
public void paintComponenet(Graphics g)
{
Graphics2D g2 = (Graphics2D) g;
Rectangle2D.Double rect = new Rectangle2D.Double(0,0,20,30);
//Rectangle rect = new Rectangle(0,0,50,100);
g2.draw(rect);
g2.setColor(Color.red);
//g2.fill(rect);
}
}
```

Cont …..

```
class FrameTest
{
public static void main(String args[])
{
JFrame f1 =new JFrame("My Frame");
f1.setLayout(new BorderLayout());
rectComponent r1 =  new rectComponent();
f1.setSize(300,400);
f1.add(r1,BorderLayout.CENTER);
//f1.show();
f1.setVisible(true);
}
}
```

Adding a Rectangle to Frame

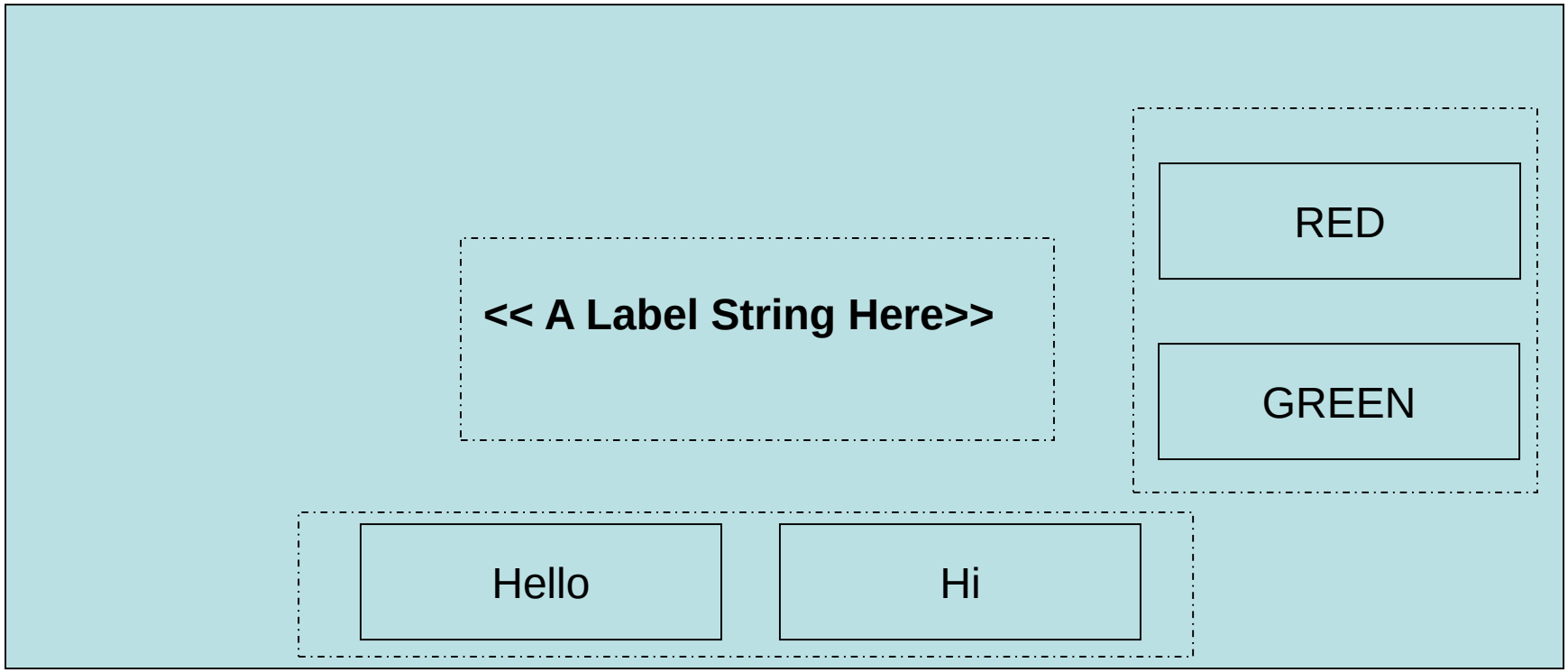
METHOD 2 : Make a class implemeting Icon interface

```
import java.awt.*;
import javax.swing.*;
import java.awt.geom.*;
class rectComponent implements Icon
{
private int height; private int width;
rectComponent(int h,int w)
{ height = h; width = w; }
public int getIconHeight() { return height; }
public int getIconWidth()  { return width; }
public void paintIcon(Component c,Graphics g,int x,int y)
{
Graphics2D g2 = (Graphics2D) g;
Rectangle2D.Double rect = new Rectangle2D.Double(x,y,width,height);
//Rectangle rect = new Rectangle(0,0,50,100);
g2.draw(rect);
g2.setColor(Color.red);
g2.fill(rect);
}
}
```

Cont …..

```
class FrameTest1
{
public static void main(String args[])
{
JFrame f1 =new JFrame("My Frame");
f1.setLayout(new BorderLayout());
rectComponent r1 =  new rectComponent(50,100);
JLabel l1 = new JLabel(r1);
f1.setSize(300,400);
f1.add(l1,BorderLayout.CENTER);
f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
//f1.show();
f1.setVisible(true);
}
}
```

Design and Implement a Frame as follows



Without Inner Classes

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

class buttonlistener implements ActionListener
{
JLabel l1 ;
buttonlistener(JLabel l1)
{
this.l1 = l1;
}
public void actionPerformed(ActionEvent ae)
{
String str = ae.getActionCommand();
if(str.equals("Hello"))
l1.setText("Hello How are you");
if(str.equals("Hi"))
l1.setText("Hi How are you");
if(str.equals("RED"))
l1.setForeground(Color.red);
if(str.equals("GREEN"))
l1.setForeground(Color.green);
}
} // End of class
```

```
class mainframetest
{
public static void main(String args[])
{
JFrame f1 = new JFrame("Main Frame");
JPanel p1 = new JPanel();
p1.setLayout(new BoxLayout(p1,BoxLayout.Y_AXIS));

JPanel p2 = new JPanel();

JPanel p3 = new JPanel();
p3.setLayout(new BorderLayout());

JButton b1 = new JButton("RED");
JButton b2 = new JButton("GREEN");
JButton b3 = new JButton("Hello");
JButton b4 = new JButton("Hi");

p1.add(b1);
p1.add(b2);
p2.add(b3);
p2.add(b4);
JLabel l1 =new JLabel("Hello How are you");
p3.add(l1,BorderLayout.CENTER);
```

```
b1.addActionListener(new buttonlistener(l1));
b2.addActionListener(new buttonlistener(l1));
b3.addActionListener(new buttonlistener(l1));
b4.addActionListener(new buttonlistener(l1));

f1.add(p1,BorderLayout.EAST);
f1.add(p3,BorderLayout.CENTER);
f1.add(p2,BorderLayout.SOUTH);

f1.setSize(400,600);
f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f1.show();
}
}
```

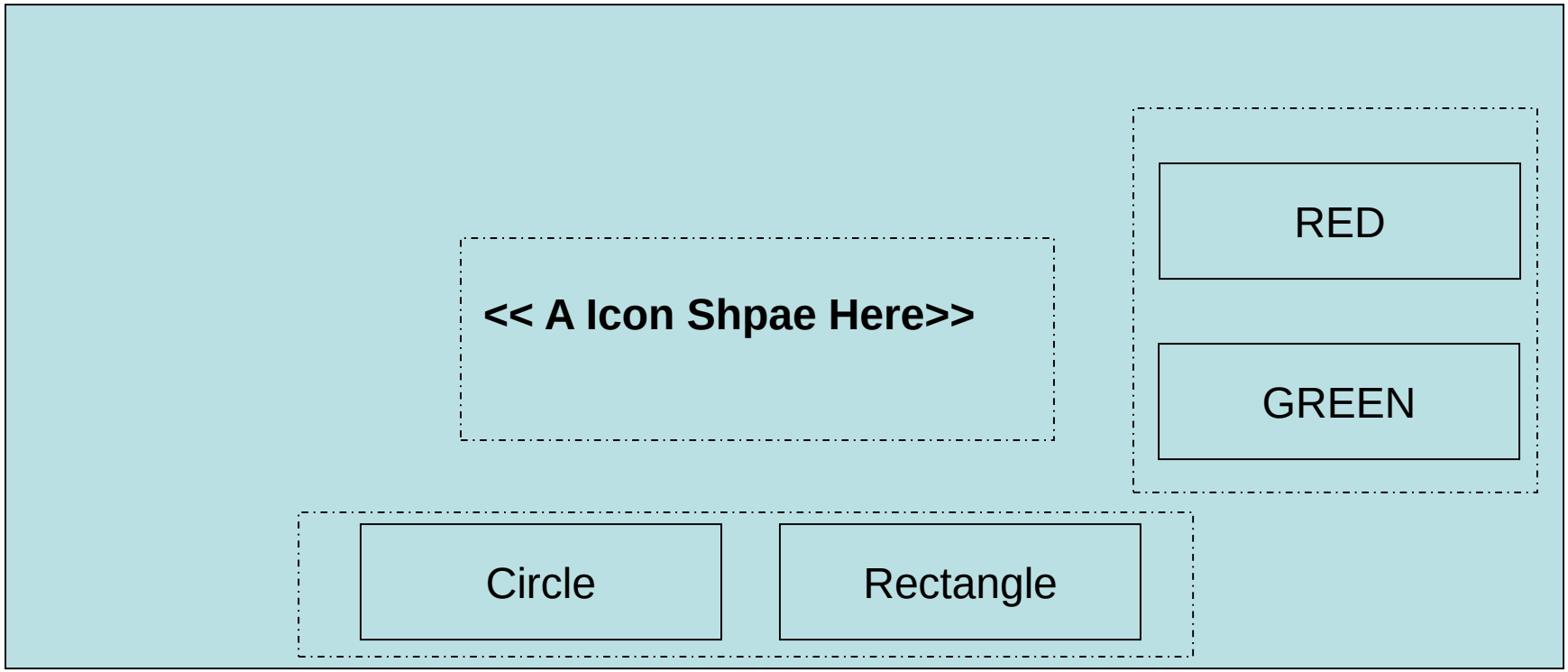
Use Inner classes

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

class mainframetest
{
public static void main(String args[])
{
JFrame f1 = new JFrame("Main Frame");
JPanel p1 = new JPanel();
p1.setLayout(new BoxLayout(p1,BoxLayout.Y_AXIS));

JPanel p2 = new JPanel();
//p2.setLayout(new BorderLayout());

JPanel p3 = new JPanel();
p3.setLayout(new BorderLayout());

JButton b1 = new JButton("RED");
JButton b2 = new JButton("GREEN");
JButton b3 = new JButton("Hello");
JButton b4 = new JButton("Hi");
```

```
p1.add(b1);
p1.add(b2);
p2.add(b3);
p2.add(b4);
final JLabel l1 =new JLabel("Hello How are you");
p3.add(l1,BorderLayout.CENTER);

b1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setForeground(Color.red);
}
});

b2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setForeground(Color.green);
}
});
```

```
b3.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setText("Hello How are you");
}
});
b4.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setText("Hi How are you");
}
});

f1.add(p1,BorderLayout.EAST);
f1.add(p3,BorderLayout.CENTER);
f1.add(p2,BorderLayout.SOUTH);

f1.setSize(400,600);
f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f1.show();
}
}
```

Icon Frames



```
import java.awt.*;
import java.awt.geom.*;
import javax.swing.*;
import java.awt.event.*;

 class circleicon implements Icon
 {
 private int size;
 circleicon(int radius) { size = radius;}
 public int getIconHeight() { return size;}
 public int getIconWidth()  { return size;}
 public void paintIcon(Component c, Graphics g, int x,int y)
 {
 Graphics2D g2 = (Graphics2D) g;
 Ellipse2D.Double circle = new Ellipse2D.Double(x,y,size,size);
 g2.draw(circle);
 }
 }
```

```
class recticon implements Icon
{
private int height;
private int width;
recticon(int height,int witdh)
{
this.height = height;
this.width = width;
}
public int getIconHeight() { return height;}
public int getIconWidth()  { return width;}
public void paintIcon(Component c, Graphics g, int x,int y)
{
Graphics2D g2 = (Graphics2D) g;
Rectangle2D.Double r1 = new Rectangle2D.Double(5,5,x-width,y-
height);
g2.draw(r1);
}
}
```

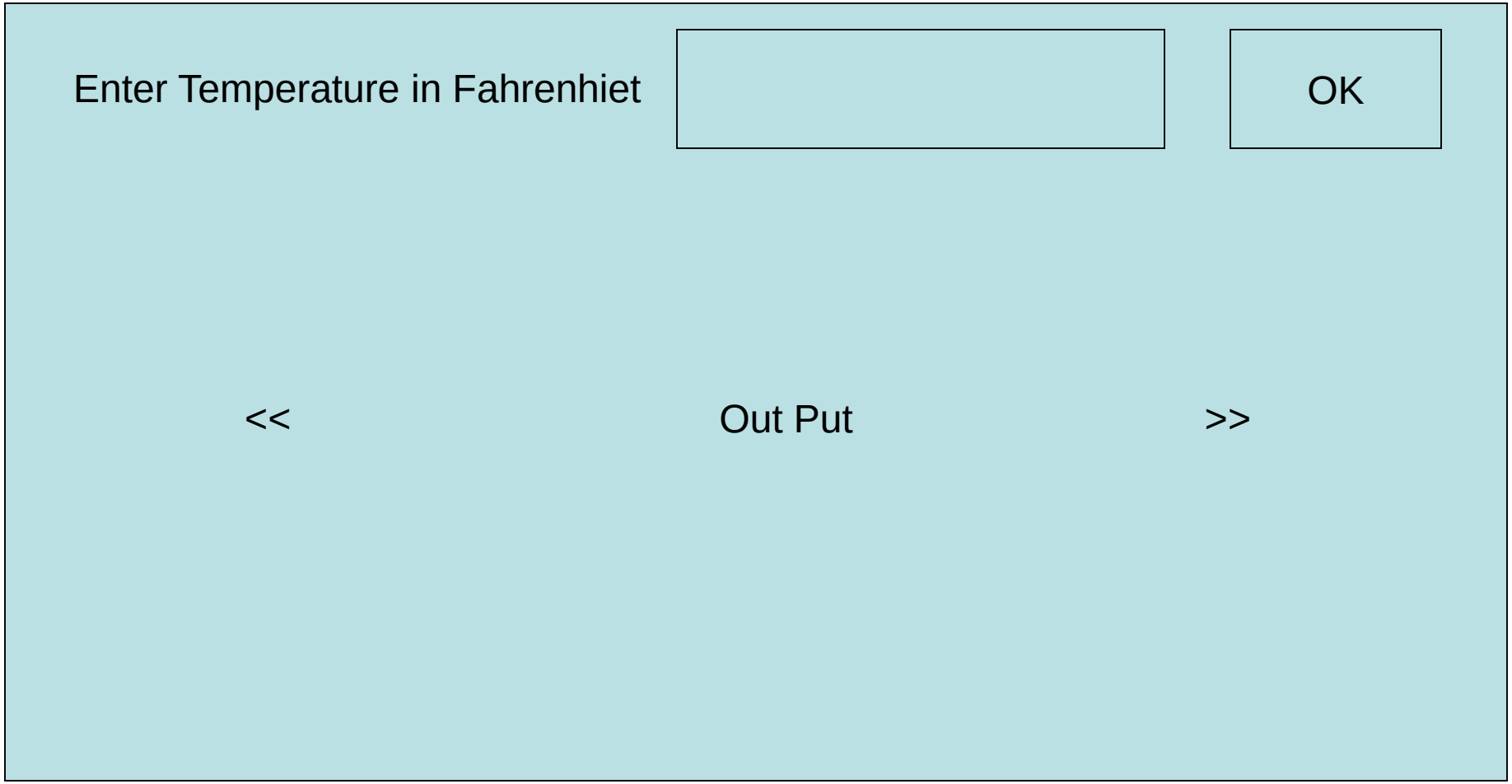
```
class maintest
{
public static void main(String args[])
{
JFrame f1 = new JFrame("Main Frame");

JPanel p1 = new JPanel();
p1.setLayout(new BoxLayout(p1,BoxLayout.Y_AXIS));

JPanel p2 = new JPanel();
//p2.setLayout(new BorderLayout());

JPanel p3 = new JPanel();
p3.setLayout(new BorderLayout());

JButton b1 = new JButton("RED");
JButton b2 = new JButton("GREEN");

JButton b3 = new JButton("Circle Icon");
JButton b4 = new JButton("Rectangle Icon");
```

```
p1.add(b1);
p1.add(b2);
p2.add(b3,BorderLayout.CENTER);
p2.add(b4,BorderLayout.CENTER);
final circleicon cir = new circleicon(50);
final recticon   rect = new recticon(10,20);
final JLabel l1 =new JLabel(cir);
b1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setForeground(Color.red);
}});
b2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setForeground(Color.green);
}});
```

```
b3.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setIcon(cir);
}});

b4.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
l1.setIcon(rect);
}});
```

```
p3.add(l1,BorderLayout.CENTER);

Container c1 = f1.getContentPane();
c1.setLayout(new BorderLayout());

c1.add(p1,BorderLayout.EAST);
c1.add(p3,BorderLayout.CENTER);
c1.add(p2,BorderLayout.SOUTH);

f1.setSize(400,600);
f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f1.show();
}
}
```

Example TextField Demo



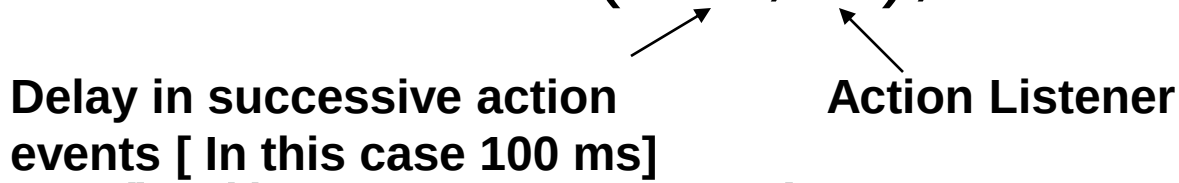
```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

class textfielddemo
{
public static void main(String args[])
{
JFrame f1 = new JFrame("Temprature Converiosn Demo");
f1.setLayout(new BorderLayout());
JLabel l1 = new JLabel("Enter Temprature: ");
final JLabel l2 = new JLabel();

final JTextField txt = new JTextField(10);
JButton btn = new JButton("OK");

JPanel p1 =new JPanel();
p1.add(l1);
p1.add(txt);
p1.add(btn);

f1.add(p1,BorderLayout.NORTH);
f1.add(l2,BorderLayout.CENTER);
```

```
btn.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
double f = Double.parseDouble(txt.getText());
int c = (int) Math.round(5*(f-32)/9);
txt.setText("");
l2.setText("Temprature in Centigrade is :"+c);
}
});

f1.setSize(400,400);
f1.setLocation(100,200);
f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f1.show();
}
}
```

Timer class in Java

- Used for Animation Pupose
- Defined in javax.swing. Package
- Generates action events spaced at equal intervals.
- Syntax :

  Timer T1 = new Time(100,L1);



T1.start(); //starts generating events

T1.stop(); // stops generating events

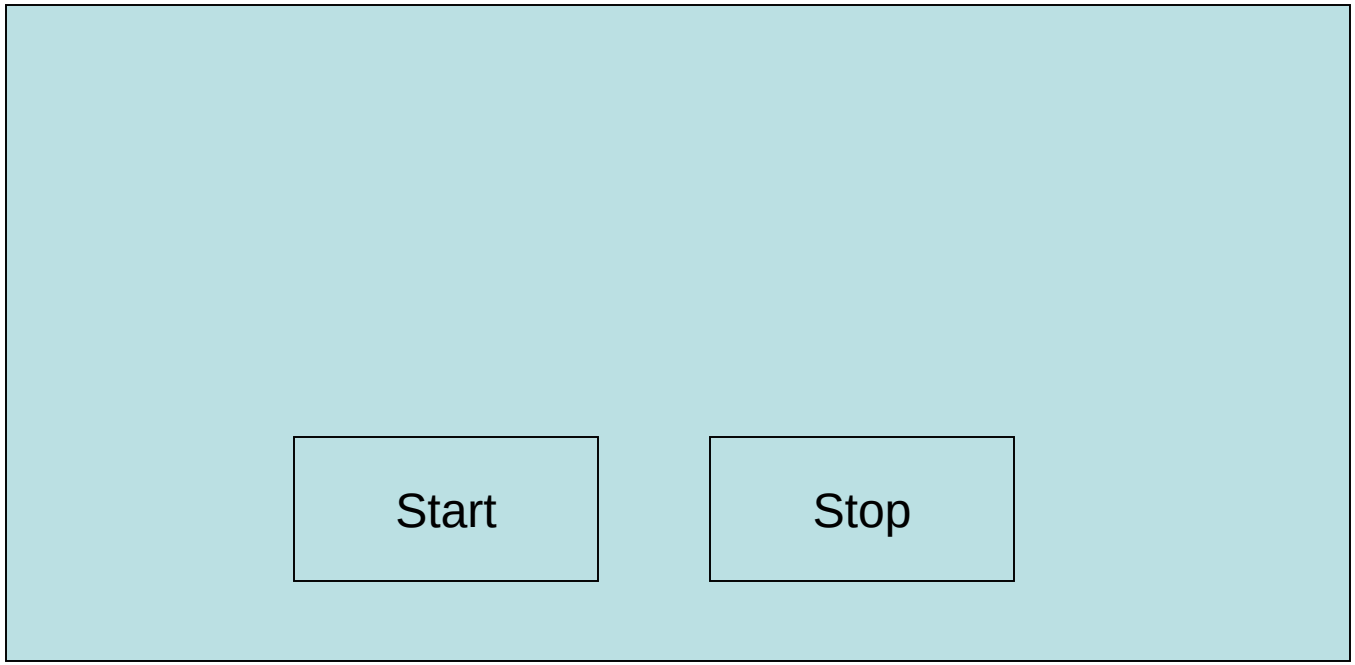
Example

- **Show a Frame which displays texts “Hello” and “Hi” alternatively after 10 ms interval.**

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class TimerTest
{
static boolean isHello = false;
public static void main(String args[])
{
JFrame f1 = new JFrame();
f1.setSize(400,600);
final JLabel label = new JLabel("Hi");
f1.add(label,BorderLayout.CENTER);
```

```
ActionListener l1 =new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(isHello)
{
label.setText("Hi");
isHello = false;
}
else
{
label.setText("Hello");
isHello = true;
}
}
};

Timer t1 = new Timer(100,l1);
t1.start();
f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f1.show();
}
}
```

Animation Effects using Threads



<< Start>> will display displaying texts “Hello” and “Hi” alternatively

<<Stop>> will stop alternating the text.

USING suspend and resume (Java 1.1)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class LabelThread extends Thread
{
 JLabel lbl;
 boolean helloFlag;
 LabelThread(JLabel lbl)
 {
 this.lbl = lbl;
 helloFlag = true;
 }
 public void run()
 {
 try
 {
 if(helloFlag)
 {
 lbl.setText("Hi");
 helloFlag = false;
 }
 else
 {
 lbl.setText("Hello");
 helloFlag = true;
 }
 Thread.sleep(10);
 run();
 }
 catch(InterruptedException ae){}
 }
}
```

```
class MYFRAME
{
static boolean started =false;
public static void main(String args[])
{
JFrame f1 = new JFrame();
final JLabel lbl = new JLabel("Hello");
final LabelThread T1 = new LabelThread(lbl);

JButton b1 = new JButton("Start");
JButton b2 = new JButton("Stop");
JPanel btnPanel = new JPanel();
btnPanel.add(b1);
btnPanel.add(b2);

f1.add(lbl,BorderLayout.CENTER);
f1.add(btnPanel,BorderLayout.SOUTH);
```

```
b1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(!started)
{
T1.start();
started = true;
return;
}
T1.resume();
}
});
b2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
T1.suspend();
}
});
f1.setSize(400,600);
f1.show(); } }
```

Suspend and Resume in Java 2 (Page 308 complete reference)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class LabelThread extends Thread
{
 JLabel lbl;
 boolean helloFlag;
 boolean suspendflag =false;
 LabelThread(JLabel lbl)
 {
 this.lbl = lbl;
 helloFlag = true;
 }
```

```
public void run()
 {
 try
 {
 synchronized(this)
 {
 while(suspendflag)
 wait();
 }
 if(helloFlag)
 {
 lbl.setText("Hi");
 helloFlag = false;
 }
 else
 {
 lbl.setText("Hello");
 helloFlag = true;
 }
 Thread.sleep(10);
 run();
 }
 catch(InterruptedException ae){}
 }
```

```
void mysuspend() { suspendflag = true;}
synchronized void myresume()
{
suspendflag = false;
notify();
}
}
```

```
class MYFRAME15
{
static boolean started =false;
public static void main(String args[])
{
JFrame f1 = new JFrame();
final JLabel lbl = new JLabel("Hello");
final LabelThread T1 = new LabelThread(lbl);

JButton b1 = new JButton("Start");
JButton b2 = new JButton("Stop");
JPanel btnPanel = new JPanel();
btnPanel.add(b1);
btnPanel.add(b2);
f1.add(lbl,BorderLayout.CENTER);
f1.add(btnPanel,BorderLayout.SOUTH);
```

```
b1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
if(!started)
{
T1.start();
started = true;
return;
}
T1.myresume();
}
});
b2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent ae)
{
T1.mysuspend();
}
});

f1.setSize(400,600);
f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f1.show();
} }
```