

Calculator Application using ActionListener with Frame

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

public class Calculator extends Frame {
    TextField tf1;
    TextField tf2;
    Label l1;
    Button b;

    Calculator() {
        setTitle("Adder");
        tf1 = new TextField();
        tf1.setBounds(100, 50, 85, 20);
        tf2 = new TextField();
        tf2.setBounds(100, 100, 85, 20);
        b = new Button("Add");
        b.setBounds(110, 220, 60, 40);
        l1 = new Label("");
        l1.setBounds(100, 120, 85, 20);
        add(b);
        add(tf1);
        add(tf2);
        add(l1);
        setSize(300, 300);
        setVisible(true);

        b.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int a = Integer.parseInt(tf1.getText());
```

```

        int b = Integer.parseInt(tf2.getText());

        int c = a + b;

        l1.setText("Their sum is = " + String.valueOf(c));

    }

});

}

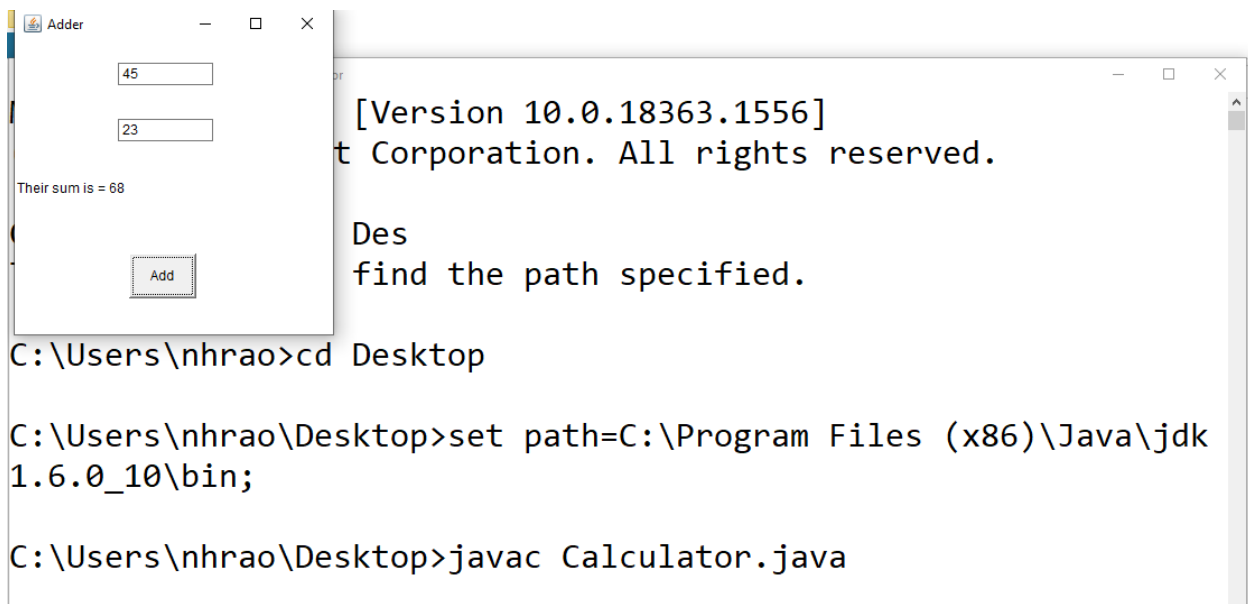
public static void main(String []args) {

    Calculator c=new Calculator();

}

}

```



Addition of two numbers using ActionListener with Applets

Add.java

```

import java.awt.*;

import java.awt.event.*;

import java.applet.*;

```

```
public class Add extends Applet implements ActionListener
```

```
{
```

```
    TextField t1 = new TextField(10);
```

```
    TextField t2 = new TextField(10);
```

```
    TextField t3 = new TextField(10);
```

```
    Label l1 = new Label("FIRST NO:");
```

```
    Label l2 = new Label("SECOND NO:");
```

```
    Label l3 = new Label("SUM:");
```

```
    Button b = new Button("ADD");
```

```
    public void init()
```

```
{
```

```
        t1.setForeground(Color.red);
```

```
        add(l1);
```

```
        add(t1);
```

```
        add(l2);
```

```
        add(t2);
```

```
        add(l3);
```

```
        add(t3);
```

```
        add(b);
```

```
        b.addActionListener(this);
```

```
}
```

```
    public void actionPerformed(ActionEvent e)
```

```
{
```

```
        if (e.getSource() == b)
```

```
        {
```

```
            int n1 = Integer.parseInt(t1.getText());
```

```
        int n2 = Integer.parseInt(t2.getText());  
        t3.setText(" " + (n1 + n2));  
    }  
}  
}
```

App.html

<HTML>

<HEAD>

<TITLE>WELCOME TO JAVA APPLET</TITLE>

</HEAD>

<BODY>

<CENTER>

<H1>WELCOME TO THE APPLET</H1> </CENTER>

**
**

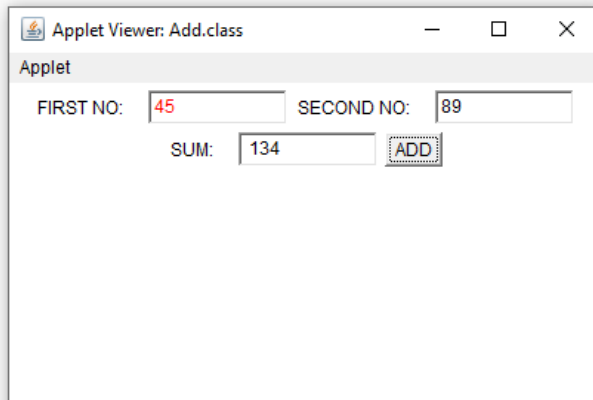
<APPLET CODE=Add.class WIDTH=400 HEIGHT=400> </APPLET>

</BODY>

</HTML>

```
C:\Users\nhrao\Desktop>javac Add.java
```

```
C:\Users\nhrao\Desktop>appletviewer app.html
```



KeyEvents

SimpleKey.java

```
// Demonstrate the key event handlers.
```

```
import java.awt.*;
```

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

```
import java.applet.*;
```

```
/*
```

```
<applet code="SimpleKey" width=300 height=100>
```

```
</applet>
```

```
*/
```

```
public class SimpleKey extends Applet implements KeyListener
```

```
{
```

```
String msg = "";
```

```
int X = 10, Y = 20; // output coordinates
```

```
public void init() {  
    addKeyListener(this);  
}
```

```
public void keyPressed(KeyEvent ke)  
{  
    showStatus("Key Down");  
}
```

```
public void keyReleased(KeyEvent ke)  
{  
    showStatus("Key Up");  
}
```

```
public void keyTyped(KeyEvent ke)  
{  
    msg += ke.getKeyChar();  
    repaint();  
}
```

```
// Display keystrokes.
```

```
public void paint(Graphics g) {  
    g.drawString(msg, X, Y);  
}  
}
```

MouseEvents

MouseEvents.java

```
// Demonstrate the mouse event handlers.  
  
import java.awt.*;  
import java.awt.event.*;  
import java.applet.*;  
  
/*  
    <applet code="MouseEvents" width=300 height=100>  
    </applet>  
*/  
  
    public class MouseEvents extends Applet implements MouseListener,  
    MouseMotionListener  
    {  
        String msg = "";  
  
        int mouseX = 0, mouseY = 0; // coordinates of mouse
```

```

public void init()
{
    addMouseListener(this);
    addMouseMotionListener(this);

}

// Handle mouse clicked.
public void mouseClicked(MouseEvent me)
{
    // save coordinates
        //mouseX = 0;
        //mouseY = 10;
        mouseX = me.getX();
        mouseY = me.getY();

    msg = "Mouse clicked.";
    repaint();
}

// Handle mouse entered.
public void mouseEntered(MouseEvent me)
{
    // save coordinates
        //mouseX = 0;
        //mouseY = 10;

```



```
        mouseX = me.getX();  
        mouseY = me.getY();  
        msg = "Mouse entered."  
        repaint();  
    }
```

// Handle mouse exited.

```
public void mouseExited(MouseEvent me)  
{  
    // save coordinates  
        //mouseX = 0;  
        //mouseY = 10;  
        mouseX = me.getX();  
        mouseY = me.getY();  
        msg = "Mouse exited."  
        repaint();  
}
```

// Handle button pressed.

```
public void mousePressed(MouseEvent me)  
{  
    // save coordinates  
        mouseX = me.getX();  
        mouseY = me.getY();  
        msg = "Down";  
        repaint();  
}
```

```
}
```

```
// Handle button released.  
public void mouseReleased(MouseEvent me)  
{  
    // save coordinates  
    mouseX = me.getX();  
    mouseY = me.getY();  
    msg = "Up";  
    repaint();  
}
```

```
// Handle mouse dragged.  
public void mouseDragged(MouseEvent me)  
{  
    // save coordinates  
    mouseX = me.getX();  
    mouseY = me.getY();  
    msg = "*";  
    showStatus("Dragging mouse at " + mouseX + ", " + mouseY);  
    repaint();  
}
```

```
// Handle mouse moved.
```

```
public void mouseMoved(MouseEvent me)
```

```
{
```

```
    showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
```

```
}
```

```
// Display msg in applet window at current X,Y location.
```

```
    public void paint(Graphics g)
```

```
    {
```

```
        g.drawString(msg, mouseX, mouseY);
```

```
    }
```

```
}
```

Layout Managers

BorderLayoutDemo.java

```
// Demonstrate BorderLayout.
```

```
import java.awt.*;
```

```
import java.applet.*;
```

```
import java.util.*;
```

```
/*
```

```
<applet code="BorderLayoutDemo" width=400 height=200>
```

```
</applet>
```

```
*/
```

```
public class BorderLayoutDemo extends Applet
```

```
{
```

```
public void init()
```

```
{
```

```
setLayout(new BorderLayout());
```

```
add(new Button("This is across the top."),BorderLayout.NORTH);
```

```
//add(new Label("The footer message might go here."),
```

```
//BorderLayout.SOUTH);
```

```
add(new Button("Bottom"), BorderLayout.SOUTH);
```

```
add(new Button("Right"), BorderLayout.EAST);
```

```
add(new Button("Left"), BorderLayout.WEST);
```

```
String msg = "RVR & JC"+"\n"+"CSE Department"+"\n"+"II B.Tech CSE-A";
```

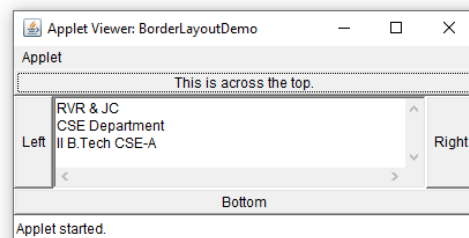
```
add(new TextArea(msg), BorderLayout.CENTER);
```

```
}
```

```
}
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>javac BorderLayoutDemo.java
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>appletviewer BorderLayoutDemo.java
```



GridLayout

```
// Demonstrate GridLayout
```

```
import java.awt.*;
```

```
import java.applet.*;
```

```
/*  
<applet code="GridLayoutDemo" width=300 height=200>  
</applet>  
*/
```

```
public class GridLayoutDemo extends Applet
```

```
{
```

```
static final int n = 4;
```

```
public void init()
```

```
{
```

```
setLayout(new GridLayout(n, n));
```

```
setFont(new Font("SansSerif", Font.BOLD, 24));
```

```
for(int i = 0; i < n; i++)
```

```
{
```

```
for(int j = 0; j < n; j++)
```

```
{
```

```
int k = i * n + j;
```

```
if(k > 0)
```

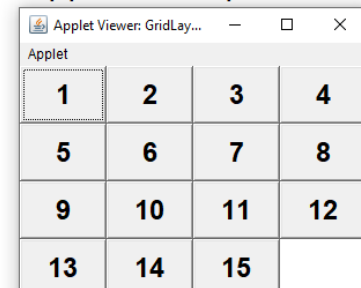
```
add(new Button("" + k));
```

```
}
```

```
}  
}  
}
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>javac GridLayoutDemo.java
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>appletviewer GridLayoutDemo.java
```



FlowLayout

```
// Use left-aligned flow layout.
```

```
import java.awt.*;
```

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

```
import java.applet.*;
```

```
/*
```

```
<applet code="FlowLayoutDemo" width=250 height=200>
```

```
</applet>
```

***/**

public class FlowLayoutDemo extends Applet implements ItemListener

{

String msg = "";

Checkbox winXP, winVista, solaris, mac;

public void init()

{

// set left-aligned flow layout

setLayout(new FlowLayout(FlowLayout.CENTER));

winXP = new Checkbox("Windows XP", null, true);

winVista = new Checkbox("Windows Vista");

solaris = new Checkbox("Solaris");

mac = new Checkbox("Mac OS");

add(winXP);

add(winVista);

add(solaris);

add(mac);


```
// register to receive item events
```

```
winXP.addItemListener(this);
```

```
winVista.addItemListener(this);
```

```
solaris.addItemListener(this);
```

```
mac.addItemListener(this);
```

```
}
```

```
// Repaint when status of a check box changes.
```

```
public void itemStateChanged(ItemEvent ie)
```

```
{
```

```
repaint();
```

```
}
```

```
// Display current state of the check boxes.
```

```
public void paint(Graphics g)
```

```
{
```

```
msg = "Current state: ";
```

```
g.drawString(msg, 6, 80);
```

```
msg = " Windows XP: " + winXP.getState();
```

```
g.drawString(msg, 6, 100);
```

```
msg = " Windows Vista: " + winVista.getState();
```

```
g.drawString(msg, 6, 120);
```

```
msg = " Solaris: " + solaris.getState();
```

```
g.drawString(msg, 6, 140);
```

```
msg = " Mac: " + mac.getState();
```

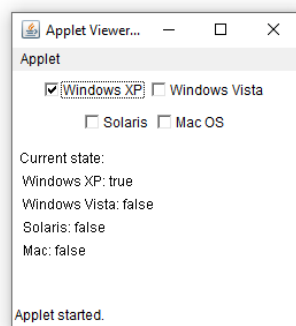
```
g.drawString(msg, 6, 160);
```

```
}
```

```
}
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>javac FlowLayoutDemo.java
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>appletviewer FlowLayoutDemo.java
```



CardLayout Manager

```
// Demonstrate CardLayout.
```

```
import java.awt.*;
```

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

```
import java.applet.*;
```

```
/*  
<applet code="CardLayoutDemo" width=300 height=100>  
</applet>  
*/
```

```
public class CardLayoutDemo extends Applet implements ActionListener,  
MouseListener
```

```
{
```

```
Checkbox winXP, winVista, solaris, mac;
```

```
Panel osCards;
```

```
CardLayout cardLO;
```

```
Button Win, Other;
```

```
public void init()
```

```
{
```

```
Win = new Button("Windows");
```

```
Other = new Button("Other");
```

```
add(Win);
```

```
add(Other);
```

```
cardLO = new CardLayout();
```

```
osCards = new Panel();  
osCards.setLayout(cardLO); // set panel layout to card layout
```

```
winXP = new Checkbox("Windows XP", null, true);  
winVista = new Checkbox("Windows Vista");  
solaris = new Checkbox("Solaris");  
mac = new Checkbox("Mac OS");
```

```
// add Windows check boxes to a panel
```

```
Panel winPan = new Panel();  
winPan.add(winXP);  
winPan.add(winVista);
```

```
// add other OS check boxes to a panel
```

```
Panel otherPan = new Panel();  
otherPan.add(solaris);  
otherPan.add(mac);
```

```
// add panels to card deck panel
```

```
osCards.add(winPan, "Windows");  
osCards.add(otherPan, "Other");
```

```
// add cards to main applet panel
```

```
add(osCards);
```

```
// register to receive action events
```

```
Win.addActionListener(this);
```

```
Other.addActionListener(this);
```

```
// register mouse events
```

```
addMouseListener(this);
```

```
}
```

```
// Cycle through panels.
```

```
public void mousePressed(MouseEvent me)
```

```
{
```

```
cardLO.next(osCards);
```

```
}
```

```
// Provide empty implementations for the other MouseListener methods.
```

```
public void mouseClicked(MouseEvent me)
```

```
{
```

```
}
```

```
public void mouseEntered(MouseEvent me)
{
}

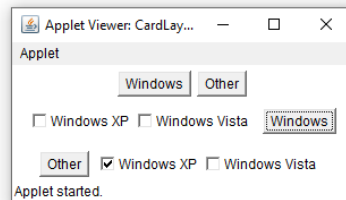
public void mouseExited(MouseEvent me)
{
}

public void mouseReleased(MouseEvent me)
{
}

public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource() == Win)
    {
        cardLO.show(osCards, "Windows");
    }
    else
    {
        cardLO.show(osCards, "Other");
    }
}
}
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>javac CardLayoutDemo.java
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>appletviewer CardLayoutDemo.java
```



MouseAdapter Example

/*

*** Java provides a special feature, called an adapter class,
that can simplify the creation of event handlers in certain situations.**

***An adapter class provides an empty implementation of all methods
in an event listener interface.**

***Adapter classes are useful when you want to receive and process
only some of the events that are handled by a particular event listener interface.**

***You can define a new class to act as an event listener by extending one of the adapter classes and implementing only those events in which you are interested.**

***/**

// Demonstrate an adapter.

import java.awt.*;

import java.awt.event.*;

import java.applet.*;

/*

<applet code="AdapterDemo" width=300 height=100>

</applet>

***/**

public class AdapterDemo extends Applet

{

public void init()

{

addMouseListener(new MyMouseAdapter(this));


```
addMouseListener(new MyMouseMotionAdapter(this));  
}
```

```
}  
  
class MyMouseAdapter extends MouseAdapter
```

```
{  
    AdapterDemo adapterDemo;
```

```
  
    public MyMouseAdapter(AdapterDemo adapterDemo)
```

```
{  
    this.adapterDemo = adapterDemo;  
}
```

```
  
    // Handle mouse clicked.
```

```
  
    public void mouseClicked(MouseEvent me)
```

```
{  
    adapterDemo.showStatus("Mouse clicked");  
}  
}
```

```
  
class MyMouseMotionAdapter extends MouseMotionAdapter
```

```
{
```

```
AdapterDemo adapterDemo;
```

```
public MyMouseMotionAdapter(AdapterDemo adapterDemo)
```

```
{
```

```
this.adapterDemo = adapterDemo;
```

```
}
```

```
// Handle mouse dragged.
```

```
public void mouseDragged(MouseEvent me)
```

```
{
```

```
adapterDemo.showStatus("Mouse dragged");
```

```
}
```

```
}
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>javac AdapterDemo.java
```

```
C:\Users\nhrao\Desktop\CS215_OOP(JAVA)\AppletExamples\JavaPrograms>appletviewer AdapterDemo.java
```

