

```
import java.awt.*;
import java.awt.event.*;
class awt10 extends Frame implements ActionListener
{
    private Button b1,b2;
    private TextField t1;
    awt10()
    {
        t1=new TextField(20);
        b1=new Button("One");
        b2=new Button("Two");
        b1.addActionListener(this);
        b2.addActionListener(this);
        setLayout(new FlowLayout());
        add(t1);
        add(b1);
        add(b2);
        setLocation(10,10);
        setSize(400,400);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b1)
        {
            t1.setText("One clicked");
        }
        if(e.getSource()==b2)
        {
            t1.setText("Two Clicked");
        }
    }
}

class example28psp
{
    public static void main(String gg[])
    {
        awt10 a=new awt10();
    }
}

import java.awt.*;
import java.awt.event.*;
class awt11 extends Frame implements ItemListener
{
    private Checkbox c1,c2;
    private TextField t1;
```

---

```
awt11()
{
c1=new Checkbox("Swimming");
c2=new Checkbox("Reading");
t1=new TextField(20);
c1.addItemListener(this);
c2.addItemListener(this);
setLayout(new FlowLayout());
add(c1);
add(c2);
add(t1);
setLocation(10,10);
setSize(400,400);
setVisible(true);
}
public void itemStateChanged(ItemEvent e)
{
if(e.getItemSelectable()==c1)
{
if(c1.getState()==true)
{
t1.setText("Swimming Checked");
}
else
{
t1.setText("Swimming Unchecked");
}
}
if(e.getItemSelectable()==c2)
{
if(c2.getState())
{
t1.setText("Reading checked");
}
else
{
t1.setText("Reading unchecked");
}
}
}
}
class example29psp
{
public static void main(String g[])
{
awt11 a=new awt11();
}
```

```
}  
import java.awt.*;  
import java.awt.event.*;  
class awt12 extends Frame implements WindowListener  
{  
    awt12()  
    {  
        addWindowListener(this);  
        setLocation(10,10);  
        setSize(400,400);  
        setVisible(true);  
    }  
    public void windowClosing(WindowEvent e)  
    {  
        //setVisible(false);  
        System.exit(0);  
    }  
    public void windowClosed(WindowEvent e){}  
    public void windowOpened(WindowEvent e){}  
    public void windowActivated(WindowEvent e){}  
    public void windowDeactivated(WindowEvent e){}  
    public void windowIconified(WindowEvent e){}  
    public void windowDeiconified(WindowEvent e){}  
    }  
    class example30psp  
    {  
        public static void main(String gg[])  
        {  
            awt12 a=new awt12();  
        }  
    }  
}  
import java.awt.*;  
import java.awt.event.*;  
class Toy extends WindowAdapter  
{  
    public void windowClosing(WindowEvent e)  
    {  
        System.exit(0);  
    }  
}  
class awt13 extends Frame  
{  
    awt13()  
    {  
        Toy t;  
        t=new Toy();  
    }  
}
```

```

addWindowListener(t);
setLocation(10,10);
setSize(400,400);
setVisible(true);
}
}
class example31psp
{
public static void main(String gg[])
{
awt13 a=new awt13();
}
}

```

---

```

import java.awt.*;
import java.awt.event.*;
class awt14 extends Frame implements MouseListener
{
private TextField t1;
private TextArea ta;
private Button b1;
awt14()
{
t1=new TextField(40);
b1=new Button("Mouse Events");
ta=new TextArea();
setLayout(new FlowLayout());
add(t1);
add(ta);
add(b1);
b1.addMouseListener(this);
setLocation(10,10);
setSize(600,400);
setVisible(true);
}
public void displayDetails(MouseEvent ev)
{
ta.setText("Clicks : "+ev.getClickCount()+"\n");
ta.append("Location on component - x location "+ev.getX()+"\n");
ta.append("Location on component - y location "+ev.getY()+"\n");
ta.append("Location on screen - x location "+ev.getXOnScreen()+"\n");
ta.append("Location on screen - y location "+ev.getYOnScreen()+"\n");
if(ev.getButton()==ev.BUTTON1)
{
ta.append("Button Clicked : LEFT\n");
}
if(ev.getButton()==ev.BUTTON2)

```

```
{
ta.append("Button Clicked : CENTER\n");
}
if(ev.getButton()==ev.BUTTON3)
{
ta.append("Button Clicked : RIGHT\n");
}
}
public void mouseClicked(MouseEvent ev)
{
t1.setText("Mouse Clicked Event Fired");
displayDetails(ev);
}
public void mouseEntered(MouseEvent ev)
{
t1.setText("Mouse Entered Event Fired");
}
public void mouseExited(MouseEvent ev)
{
t1.setText("Mouse Exited Event Fired");
}
public void mousePressed(MouseEvent ev)
{
t1.setText("Mouse Pressed Event Fired");
displayDetails(ev);
}
public void mouseReleased(MouseEvent ev)
{
t1.setText("Mouse Released Event Fired");
}
}
class example32psp
{
public static void main(String gg[])
{
awt14 a=new awt14();
}
}
// When you will test the above example, you won't see the mouse released
// message, because the click event will get fired just after release and
// it will set the mouse clicked message. To see the mouse released message
// just comment the t1.setText in mouse clicked event
import java.awt.*;
import java.awt.event.*;
class awt15 extends Frame implements MouseMotionListener
{
```

---

```
awt15()
{
addMouseMotionListener(this);
setLayout(new BorderLayout());
setLocation(10,10);
setSize(400,400);
setVisible(true);
}
public void mouseMoved(MouseEvent ev)
{
// do some testing on your own
}
public void mouseDragged(MouseEvent ev)
{
// do some testing on your own
}
}
class example33psp
{
public static void main(String gg[])
{
awt15 a=new awt15();
}
}
import java.awt.*;
import java.awt.event.*;
class DrawingBoard extends Canvas
{
private int lastClickedXLocation,lastClickedYLocation;
DrawingBoard()
{
lastClickedXLocation=0;
lastClickedYLocation=0;
this.setBackground(Color.gray);
this.setForeground(Color.red);
}
public boolean mouseDown(Event e,int currentXLocation,int currentYLocation)
{
lastClickedXLocation = currentXLocation;
lastClickedYLocation = currentYLocation;
return true;
}
public boolean mouseDrag(Event e,int currentXLocation,int currentYLocation)
{
Graphics g = getGraphics();
g.drawLine(lastClickedXLocation,lastClickedYLocation,currentXLocation,currentYLocation);
```

```
lastClickedXLocation = currentXLocation;
lastClickedYLocation = currentYLocation;
return true;
}
}
class DrawingBoardFrame extends Frame
{
private DrawingBoard db;
DrawingBoardFrame()
{
db=new DrawingBoard();
setLayout(new BorderLayout());
add(db,BorderLayout.CENTER);
setLocation(10,10);
setSize(500,600);
setVisible(true);
}
}
class example34psp
{
public static void main(String gg[])
{
DrawingBoardFrame dbf=new DrawingBoardFrame();
}
}
// a gif file (java.gif) has been used in this example
// make sure that such kind of a image file is lying
// in the working directory. You can take any image file
// and adjust the code accordingly
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
interface CrossButtonListener
{
public void windowClosing(WindowEvent e);
}
class CrossButtonHandler extends WindowAdapter
{
private CrossButtonListener target;
CrossButtonHandler(CrossButtonListener t)
{
target=t;
}
public void windowClosing(WindowEvent ev)
{
target.windowClosing(ev);
}
```

---

```

}
}
class AdditionFrame extends Frame implements CrossButtonListener,ActionListener
{
private TextField firstNum,secondNum;
private Button addButton;
private Label firstLabel,secondLabel;
private Label resultLabel;
AdditionFrame()
{
setTitle("Addition Module");
firstLabel=new Label("First Number");
firstNum=new TextField(10);
secondLabel=new Label("Second Label");
secondNum=new TextField(10);
addButton=new Button("Add");
addButton.addActionListener(this);
resultLabel=new Label("      ");
Panel p1=new Panel();
p1.setLayout(new GridLayout(2,2));
p1.add(firstLabel);
p1.add(firstNum);
p1.add(secondLabel);
p1.add(secondNum);
Panel p2=new Panel();
p2.setLayout(new GridLayout(2,1));
p2.add(resultLabel);
p2.add(addButton);
setLayout(new BorderLayout());
add(p1,BorderLayout.CENTER);
add(p2,BorderLayout.SOUTH);
CrossButtonHandler ch;
ch=new CrossButtonHandler(this);
addWindowListener(ch);
setLocation(10,10);
setSize(400,150);
setVisible(true);
}
public void actionPerformed(ActionEvent ev)
{
// Integer.parseInt to conver a string to int
// String.valueOf to convert int to string
int num1,num2;
try
{
num1=Integer.parseInt(firstNum.getText());
num2=Integer.parseInt(secondNum.getText());

```



```

int total=num1+num2;
resultLabel.setText("Result : "+String.valueOf(total));
} catch(NumberFormatException nfe)
{
resultLabel.setText("Please provied 2 numbers");
}
}
}
public void windowClosing(WindowEvent ev)
{
setVisible(false);
}
}

```

```

class SubstractionFrame extends Frame implements CrossButtonListener,ActionListener
{
private TextField firstNum,secondNum;
private Button subtractButton;
private Label firstLabel,secondLabel;
private Label resultLabel;
SubstractionFrame()
{
setTitle("Substraction Module");
firstLabel=new Label("First Number");
firstNum=new TextField(10);
secondLabel=new Label("Second Label");
secondNum=new TextField(10);
subtractButton=new Button("Subtract");
subtractButton.addActionListener(this);
resultLabel=new Label("      ");
Panel p1=new Panel();
p1.setLayout(new GridLayout(2,2));
p1.add(firstLabel);
p1.add(firstNum);
p1.add(secondLabel);
p1.add(secondNum);
Panel p2=new Panel();
p2.setLayout(new GridLayout(2,1));
p2.add(resultLabel);
p2.add(subtractButton);
setLayout(new BorderLayout());
add(p1,BorderLayout.CENTER);
add(p2,BorderLayout.SOUTH);
CrossButtonHandler ch;
ch=new CrossButtonHandler(this);
addWindowListener(ch);
setLocation(10,10);
setSize(400,150);
}
}

```

```

setVisible(true);
}
public void actionPerformed(ActionEvent ev)
{
    int num1,num2;
    try
    {
        num1=Integer.parseInt(firstNum.getText());
        num2=Integer.parseInt(secondNum.getText());
        int difference=num1-num2;
        resultLabel.setText("Result : "+String.valueOf(difference));
    } catch(NumberFormatException nfe)
    {
        resultLabel.setText("Please provied 2 numbers");
    }
}
public void windowClosing(WindowEvent ev)
{
    setVisible(false);
}
}
class ImagePanel extends Panel
{
    private Image img;
    ImagePanel()
    {
        img=Toolkit.getDefaultToolkit().getImage("java.gif");
        MediaTracker mt=new MediaTracker(this);
        mt.addImage(img,0);
    }
    public void update(Graphics g)
    {
        paint(g);
    }
    public void paint(Graphics g)
    {
        if(img!=null)
        {
            g.drawImage(img,0,0,this);
        }
    }
}
class AboutBox extends Frame implements CrossButtonListener
{
    private ImagePanel ip;
    private Label topMessageLabel,bottomMessageLabel;
    AboutBox()

```

```
{
setTitle("About Us");
ip=new ImagePanel();
topMessageLabel=new Label("Thinking Machines");
bottomMessageLabel=new Label("Menu Example");
setLayout(new BorderLayout());
add(topMessageLabel,BorderLayout.NORTH);
add(ip,BorderLayout.CENTER);
add(bottomMessageLabel,BorderLayout.SOUTH);
addWindowListener(new CrossButtonHandler(this));
setLocation(10,10);
setSize(300,300);
setVisible(true);
}
public void windowClosing(WindowEvent ev)
{
setVisible(false);
}
}
class MainMenu extends Frame implements ActionListener,CrossButtonListener
{
private MenuBar mb;
private Menu m1,m2,m3;
private MenuItem addMenuItem,subtractMenuItem,exitMenuItem,aboutMenuItem;
MainMenu()
{
addMenuItem=new MenuItem("Add");
addMenuItem.addActionListener(this);
subtractMenuItem=new MenuItem("subtract");
subtractMenuItem.addActionListener(this);
exitMenuItem=new MenuItem("Exit");
exitMenuItem.addActionListener(this);
aboutMenuItem=new MenuItem("About Us");
aboutMenuItem.addActionListener(this);
m1=new Menu("Options");
m2=new Menu("Math");
m3=new Menu("Help");
m3.add(aboutMenuItem);
m2.add(addMenuItem);
m2.add(subtractMenuItem);
m1.add(m2);
m1.add(exitMenuItem);
mb=new MenuBar();
mb.add(m1);
mb.add(m3);
setMenuBar(mb);
addWindowListener(new CrossButtonHandler(this));
```

```
setLocation(10,10);
setSize(500,500);
setVisible(true);
}
public void windowClosing(WindowEvent ev)
{
System.exit(0);
}
public void actionPerformed(ActionEvent ev)
{
if(ev.getSource()==addMenuItem)
{
AdditionFrame af=new AdditionFrame();
}
if(ev.getSource()==subtractMenuItem)
{
SubstractionFrame sf=new SubstractionFrame();
}
if(ev.getSource()==exitMenuItem)
{
System.exit(0);
}
if(ev.getSource()==aboutMenuItem)
{
AboutBox ab=new AboutBox();
}
}
}
class example35psp
{
public static void main(String g[])
{
MainMenu m;
m=new MainMenu();
}
}
```

---