Experiment no: 07

2) Develop a program which will implement various methods of MouseMotionListner

Code:

// Handle mouse events in both child and applet windows.

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

<applet code="WindowEvents" width=300 height=50>

</applet>

\*/

// Create a subclass of Frame.

class SampleFrame extends Frame

implements MouseListener, MouseMotionListener {

String msg = "";

int mouseX=10, mouseY=40;

int movX=0, movY=0;

SampleFrame(String title) {

super(title);

// register this object to receive its own mouse events

addMouseListener(this);

addMouseMotionListener(this);

// create an object to handle window events

MyWindowAdapter adapter = new MyWindowAdapter(this);

// register it to receive those events

addWindowListener(adapter);

}

// Handle mouse clicked.

public void mouseClicked(MouseEvent me) {

}

// Handle mouse entered.

public void mouseEntered(MouseEvent evtObj) {

// save coordinates

mouseX = 10;

mouseY = 54;

msg = "Mouse just entered child.";

repaint();

}

// Handle mouse exited.

public void mouseExited(MouseEvent evtObj) {

// save coordinates

mouseX = 10;

mouseY = 54;

msg = "Mouse just left child window.";

repaint();

}

// Handle mouse pressed.

public void mousePressed(MouseEvent me) {

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "Down";

repaint();

}

// Handle mouse 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();

movX = me.getX();

movY = me.getY();

msg = "\*";

repaint();

}

// Handle mouse moved.

public void mouseMoved(MouseEvent me) {

// save coordinates

movX = me.getX();

movY = me.getY();

repaint(0, 0, 100, 60);

}

public void paint(Graphics g) {

g.drawString(msg, mouseX, mouseY);

g.drawString("Mouse at " + movX + ", " + movY, 10, 40);

}

}

class MyWindowAdapter extends WindowAdapter {

SampleFrame sampleFrame;

public MyWindowAdapter(SampleFrame sampleFrame) {

this.sampleFrame = sampleFrame;

}

public void windowClosing(WindowEvent we) {

sampleFrame.setVisible(false);

}

}

// Applet window.

public class WindowEvents extends Applet

implements MouseListener, MouseMotionListener {

SampleFrame f;

String msg = "";

int mouseX=0, mouseY=10;

int movX=0, movY=0;

// Create a frame window.

public void init() {

f = new SampleFrame("Handle Mouse Events");

f.setSize(300, 200);

f.setVisible(true);

// register this object to receive its own mouse events

addMouseListener(this);

addMouseMotionListener(this);

}

// Remove frame window when stopping applet.

public void stop() {

f.setVisible(false);

}

// Show frame window when starting applet.

public void start() {

f.setVisible(true);

}

// Handle mouse clicked.

public void mouseClicked(MouseEvent me) {

}

// Handle mouse entered.

public void mouseEntered(MouseEvent me) {

// save coordinates

mouseX = 0;

mouseY = 24;

msg = "Mouse just entered applet window.";

repaint();

}

// Handle mouse exited.

public void mouseExited(MouseEvent me) {

// save coordinates

mouseX = 0;

mouseY = 24;

msg = "Mouse just left applet window.";

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();

movX = me.getX();

movY = me.getY();

msg = "\*";

repaint();

}

// Handle mouse moved.

public void mouseMoved(MouseEvent me) {

// save coordinates

movX = me.getX();

movY = me.getY();

repaint(0, 0, 100, 20);

}

// Display msg in applet window.

public void paint(Graphics g) {

g.drawString(msg, mouseX, mouseY);

g.drawString("Mouse at " + movX + ", " + movY, 0, 10);

}

}

Output:



3) Find Largest among three numbers.

Code:

**//find smallest and largest number among three numbers.**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

<applet code=Example width=400 height=400>

</applet>

\*/

public class Example extends Applet implements ActionListener

{

Label l1, l2, l3, l4;

TextField num1, num2, num3, result;

Button b1, b2;

public Example()

{

num1=new TextField(15);

num2=new TextField(15);

num3=new TextField(15);

result=new TextField(15);

l1=new Label("1st Number:");

l2=new Label("2nd Number:");

l3=new Label("3rd Number:");

l4=new Label("Result:");

b1=new Button("Find Largest");

b2=new Button("Find Smallest");

this.add(l1);

this.add(num1);

this.add(l2);

this.add(num2);

this.add(l3);

this.add(num3);

this.add(b1);

this.add(b2);

this.add(l4);

this.add(result);

result.setEnabled(false);

b1.addActionListener(this);

b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

int n1=0, n2=0, n3=0, re=0;

Button b=(Button) e.getSource();

n1=Integer.parseInt(num1.getText());

n2=Integer.parseInt(num2.getText());

n3=Integer.parseInt(num3.getText());

if(b==b1)

{

if((n1>n2)&&(n2>n3))

re=n1;

else if((n2>n1)&&(n2>n3))

re=n2;

else

re=n3;

}

else if(b==b2)

{

if((n1<n2)&&(n2<n3))

re=n1;

else if((n2<n1)&&(n2<n3))

re=n2;

else

re=n3;

}

Integer iob=new Integer(re);

String s=iob.toString();

result.setText(s);

}

}

Output:



