Experiment No._____ *Date___/___/2020*

TITLE OF EXPERIMENT: - A Program to create a frame using AWT. Implement mouseClicked, mouseEntered() and mouseExited() events. Frame should become visible when the mouse enters it.

DIVISION:_____ **BRANCH:** _____

BATCH:_____ **ROLL NO.:** _____

PERFORMED ON DATE: _____

SIGNATURE OF TEACHING STAFF:

EXPERIMENT NO. 2

Aim: Write a program to create a frame using AWT. Implement mouseClicked, mouseEntered() and mouseExited() events. Frame should become visible when the mouse enters it.

Software:

1. Command prompt
2. JDK 8
3. Applet viewer

Theory:

Java MouseListener Interface

The Java MouseListener is notified whenever you change the state of mouse. It is notified against MouseEvent. The MouseListener interface is found in java.awt.event package. It has five methods.

Methods of MouseListener interface

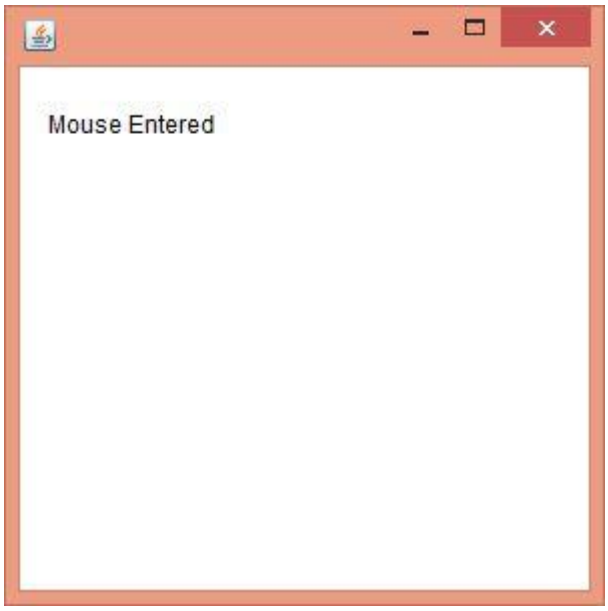
The signature of 5 methods found in MouseListener interface are given below:

1. **public abstract void** mouseClicked(MouseEvent e);
2. **public abstract void** mouseEntered(MouseEvent e);
3. **public abstract void** mouseExited(MouseEvent e);
4. **public abstract void** mousePressed(MouseEvent e);
5. **public abstract void** mouseReleased(MouseEvent e);

Java MouseListener Example

```
import java.awt.*;
import java.awt.event.*;
public class MouseListenerExample extends Frame implements MouseListener{
    Label l;
    MouseListenerExample(){
        addMouseListener(this);

        l=new Label();
        l.setBounds(20,50,100,20);
        add(l);
        setSize(300,300);
        setLayout(null);
        setVisible(true);
    }
    public void mouseClicked(MouseEvent e) {
        l.setText("Mouse Clicked");
    }
    public void mouseEntered(MouseEvent e) {
        l.setText("Mouse Entered");
    }
    public void mouseExited(MouseEvent e) {
        l.setText("Mouse Exited");
    }
    public void mousePressed(MouseEvent e) {
        l.setText("Mouse Pressed");
    }
    public void mouseReleased(MouseEvent e) {
        l.setText("Mouse Released");
    }
public static void main(String[] args) {
    new MouseListenerExample();
}
}
```

Output:



MouseListener and MouseMotionListener in Java

MouseListener and MouseMotionListener is an interface in java.awt.event package. Mouse events are of two types. MouseListener handles the events when the mouse is not in motion. While MouseMotionListener handles the events when mouse is in motion. There are five types of events that MouseListener can generate. There are five abstract functions that represent these five events. **The abstract functions are:**

1. **void mouseReleased(MouseEvent e)** : Mouse key is released
2. **void mouseClicked(MouseEvent e)** : Mouse key is pressed/released
3. **void mouseExited(MouseEvent e)** : Mouse exited the component
4. **void mouseEntered(MouseEvent e)** : Mouse entered the component
5. **void mousepressed(MouseEvent e)** : Mouse key is pressed

There are two types of events that MouseMotionListener can generate. There are two abstract functions that represent these five events. **The abstract functions are:**

1. **void mouseDragged(MouseEvent e)** : Invoked when a mouse button is pressed in the component and dragged. Events are passed until the user releases the mouse button.
2. **void mouseMoved(MouseEvent e)** : invoked when the mouse cursor is moved from one point to another within the component, without pressing any mouse buttons.

The following programs are a illustration of MouseListener and MouseMotionListener.

1. Program to handle MouseListener events

```
// Java program to handle MouseListener events

import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

class Mouse extends Frame implements MouseListener {

    // Jlabels to display the actions of events of mouseListener

     static JLabel label1, label2, label3;

    // default constructor

    Mouse()

    {

    }

    // main class

    public static void main(String[] args)

    {

        // create a frame

        JFrame f = new JFrame("MouseListener");

        // set the size of the frame

        f.setSize(600, 100);

        // close the frame when close button is pressed
```

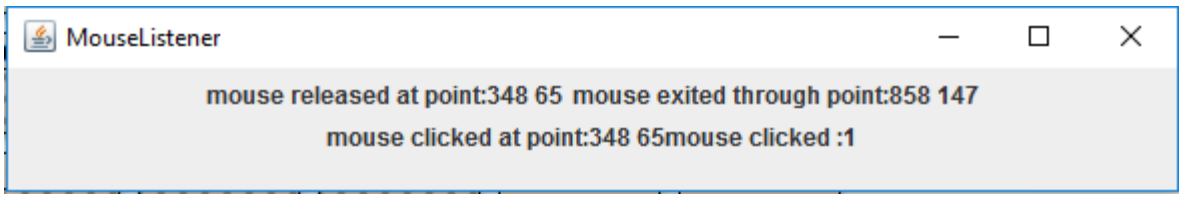
```
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// create a new panel

JPanel p = new JPanel();

// set the layout of the panel

p.setLayout(new FlowLayout());

// initialize the labels

label1 = new JLabel("no event  ");

 label2 = new JLabel("no event  ");

label3 = new JLabel("no event  ");

// create an object of mouse class

Mouse m = new Mouse();

 // add mouseListener to the frame

f.addMouseListener(m);

// add labels to the panel

p.add(label1);

p.add(label2);

p.add(label3);

// add panel to the frame

f.add(p);

f.show();
```

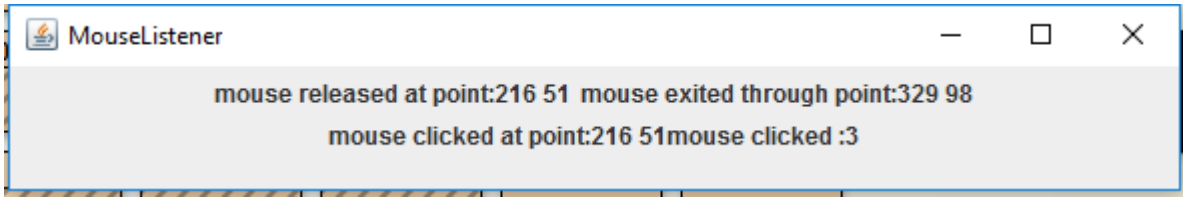
```
}

 // getX() and getY() functions return the

// x and y coordinates of the current

// mouse position

// getClickCount() returns the number of

// quick consecutive clicks made by the user

// this function is invoked when the mouse is pressed

public void mousePressed(MouseEvent e)

{

     // show the point where the user pressed the mouse

    label1.setText("mouse pressed at point:"

                + e.getX() + " " + e.getY());

}

// this function is invoked when the mouse is released

public void mouseReleased(MouseEvent e)

{

    // show the point where the user released the mouse click

    label1.setText("mouse released at point:"

                + e.getX() + " " + e.getY());

}
```

```
// this function is invoked when the mouse exits the component
public void mouseExited(MouseEvent e)
{
    // show the point through which the mouse exited the frame
    label2.setText("mouse exited through point:"
                + e.getX() + " " + e.getY());
}
 // this function is invoked when the mouse enters the component
public void mouseEntered(MouseEvent e)
{
     // show the point through which the mouse entered the frame
    label2.setText("mouse entered at point:"
                 + e.getX() + " " + e.getY());
}
// this function is invoked when the mouse is pressed or released
public void mouseClicked(MouseEvent e)
{
     // getClickCount gives the number of quick,
    // consecutive clicks made by the user
    // show the point where the mouse is i.e
```

```
        // the x and y coordinates

        label3.setText("mouse clicked at point:"

                        + e.getX() + " "

                        + e.getY() + "mouse clicked :" + e.getClickCount());

    }

}
```

Output:





MouseListener vs MouseMotionListener

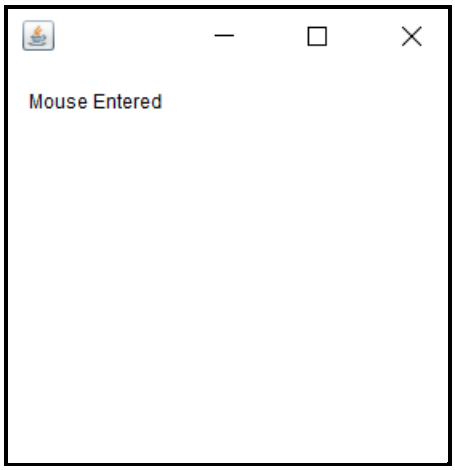
- **MouseListener:** MouseListener events are invoked when the mouse **is not in motion and is stable**. It generates events such as mousePressed, mouseReleased, mouseClicked, mouseExited and mouseEntered (i.e when the mouse buttons are pressed or the mouse enters or exits the component). The object of this class must be registered with the component and they are registered using addMouseListener() method.

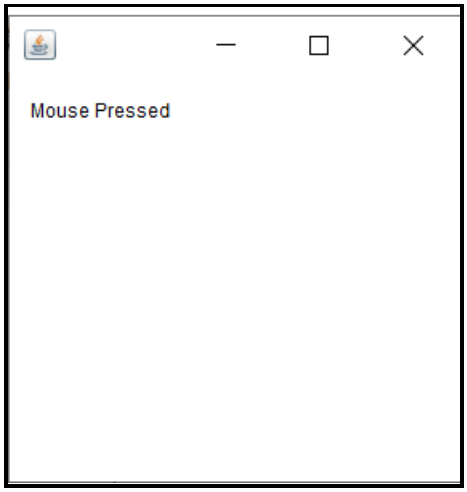
- **MouseMotionListener:** MouseMotionListener events are invoked **when the mouse is in motion**. It generates events such as mouseMoved and mouseDragged (i.e when the mouse is moved from one point to another within the component or the mouse button is pressed and dragged from one point to another). The object of this class must be registered with the component and they are registered using addMouseMotionListener() method.

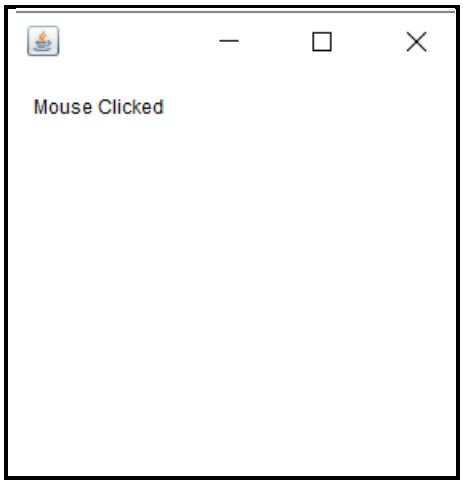
Program:

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


public class MouseListenerExample extends Frame implements MouseListener{
   Label l;

     MouseListenerExample(){
       addMouseListener(this);
       l=new Label();
       l.setBounds(20,50,100,20);
       add(l);
       setSize(300,300);
       setLayout(null);
       setVisible(true);
   }
   public void mouseClicked(MouseEvent e) {
       l.setText("Mouse Clicked");
   }
   public void mouseEntered(MouseEvent e) {
       l.setText("Mouse Entered");
   }
   public void mouseExited(MouseEvent e) {
       l.setText("Mouse Exited");
       setVisible(false);
   }
   public void mousePressed(MouseEvent e) {
       l.setText("Mouse Pressed");
   }
   public void mouseReleased(MouseEvent e) {
       l.setText("Mouse Released");
   }
public static void main(String[] args) {
   new MouseListenerExample();
}
}
```

Screenshot's of Output:







Conclusion: