Experiment No.10

Title: Implementing the concept of Mouse Event
Aim: To Study how to handle different mouse event with AWT controls.

Theory:

The MouseEvent Class

There are eight types of mouse events. The **MouseEvent** class defines the following integer constants that can be used to identify them:

MOUSE_CLICKED The user clicked the mouse.
MOUSE_DRAGGED The user dragged the mouse.
MOUSE_ENTERED The mouse entered a component.
MOUSE_EXITED The mouse exited from a component.
MOUSE_MOVED The mouse moved.
MOUSE_PRESSED The mouse was pressed.
MOUSE_RELEASED The mouse was released.
MOUSE_WHEEL The mouse wheel was moved (Java 2, v1.4).

MouseEvent is a subclass of **InputEvent**. Here is one of its constructors.

MouseEvent(Component *src*, int *type*, long *when*, int *modifiers*,int *x*, int *y*, int *clicks*, boolean *triggersPopup*)

Here, *src* is a reference to the component that generated the event. The type of the event is specified by *type*.

The most commonly used methods in this class are **getX( )** and **getY( )**. These return the X and Y coordinates of the mouse when the event occurred. Their forms are shown here:

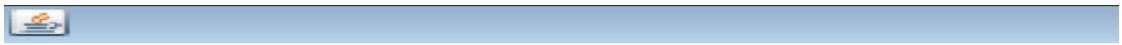
int getX( )
int getY( )

//program for the mouselistener

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


/*<applet code="ms" height=20 width=20>
</applet>*/


 public class ms extends Applet implements MouseMotionListener, MouseListener
{
        String    msg;
        int x,y;
        public void init()
        {

        addMouseMotionListener(this);
                addMouseListener(this);
        }
        public void mouseClicked(MouseEvent m)
        {       x=0;
                y=40;
                msg="Mouse Clicked";
                showStatus("mouse clicked at "+m.getX()+","+m.getY());
                repaint();
        }
        public void mouseEntered(MouseEvent m)
        {
                x=0;
                y=40;
                msg="Mouse  Entered";
                showStatus("mouse clicked at "+m.getX()+","+m.getY());
                //showStatus("");
                repaint();
        }

        public void mouseDragged(MouseEvent m)
        {
                x=m.getX();
                y=m.getY();
                msg="#";
                showStatus("Dragging mouse at "+x+","+y);
                        repaint();
        }
        public void mouseExited(MouseEvent m)
```

```
	{
		x=0;
		y=20;

		msg="mouse    Exited";
		showStatus("");



		repaint();
	}
	public void mouseReleased(MouseEvent m)
	{
		x=0;
		y=20;
		msg="mouse released ";
		showStatus("");
		repaint();
	}
	public void mousePressed(MouseEvent m)
	{


		x=0;
		y=40;
		msg="mouse pressed.";
		showStatus("Mouse    pressed    at    "+m.getX()+","+m.getY());
		repaint();
	}
	public void mouseMoved(MouseEvent m)
	{
		x=0;
		y=40;


		msg="Mouse  Moved";
		showStatus("");
		repaint();
	}
	public void paint(Graphics g)
	{
		g.drawString(msg,x,y);
	}
}
```

Output:



Conclusion: Thus we studied how to handle mouse event generated from AWT controls