



Government Engineering College, Thrissur  
CS206 - OBJECT ORIENTED DESIGN AND  
PROGRAMMING  
Assignment -  
AWT- Frames and Events

Date of Submission  
17 June 2020

Submitted By  
Kowsik Nandagopan D  
Roll No 31  
TCR18CS031  
GECT CSE S4

// Java Source Code

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

// Setting UP Frame class
class MyFrame extends Frame implements MouseListner, MouseMotionListner{

    // Variable for paint message
    String msg="";

    // Constructor
    MyFrame(String title){
        super(title);
        // Setting Up Window Adapter to to do window events
        MyWindowAdapter adapter = new MyWindowAdapter(this);
        addWindowListner(adapter);
        // Adding Motion Listner
        addMouseListners(this);
        addMouseMotionListner(this);
    }

    // MouseListner and MouseMotionListner implementation methods for Frame
    public void mouseClicked(MouseEvent me){
        msg = "Mouse Clicked in Frame";
        repaint();
    }

    public void mousePressed(MouseEvent me){
        msg = "Mouse Pressed in Frame";
        repaint();
    }

    public void mouseReleased(MouseEvent me){
        msg = "Mouse Released in Frame";
        repaint();
    }

    public void mouseEntered(MouseEvent me){
        msg = "Mouse Entered in Frame";
        repaint();
    }
}
```

```

    public void mouseExited(MouseEvent me){
        msg = "Mouse Exited in Frame";
        repaint();
    }

    public void mouseMoved(MouseEvent me){
        msg = "Mouse Moved in Frame X:" + me.getX() + " Y:" + me.getY();
        repaint();
    }

    public void mouseDragged(MouseEvent me){
        msg = "Mouse Dragged in Frame";
        repaint();
    }

    public void paint(Graphics g){
        g.drawString(msg, 10, 40);
    }
}

```

// Window Adapter class to do closing activity. Its better than implemenation interface so that we can avoid unnecessary methods

```

class MyWindowAdapter extends WindowAdapter {
    MyFrame myframe;
    // Constructor
    MyWindowAdapter(MyFrame mframe){
        this.myframe = mframe;
    }

    // Closing Function
    public void windowClosing(WindowEvent we){
        myframe.setVisible(false);
    }
}

```

```

public class SampleFrame extends Applet implements MouseListner, MouseMotionListner {

    // Variable
    MyFrame mframe;
    String msg="";

    public void init(){

```

```

        // Initializing frame in applet window
        mframe = new MyFrame("My Frame Title");
        mframe.setSize(250, 250)
            .setVisible(true);
    }

    public void start(){
        mframe.setVisible(true);
    }

    public void stop(){
        mframe.setVisible(false);
    }

    public void paint(Graphics g){
        g.drawString(msg, 10, 40);
    }

    // MouseListner and MouseMotionListner implementation methods for Main Window
    public void mouseClicked(MouseEvent me){
        msg = "Mouse Clicked in Main Window";
        repaint();
    }

    public void mousePressed(MouseEvent me){
        msg = "Mouse Pressed in Main Window";
        repaint();
    }

    public void mouseReleased(MouseEvent me){
        msg = "Mouse Released in Main Window";
        repaint();
    }

    public void mouseEntered(MouseEvent me){
        msg = "Mouse Entered in Main Window";
        repaint();
    }

    public void mouseExited(MouseEvent me){
        msg = "Mouse Exited in Main Window";
        repaint();
    }

```

```
public void mouseMoved(MouseEvent me){
    msg = "Mouse Moved in Main Window X:" + me.getX() + " Y:" + me.getY();
    repaint();
}

public void mouseDragged(MouseEvent me){
    msg = "Mouse Dragged in Main Window";
    repaint();
}
}
```