



BITS Pilani
Pilani Campus

Object Oriented Programming CS F213

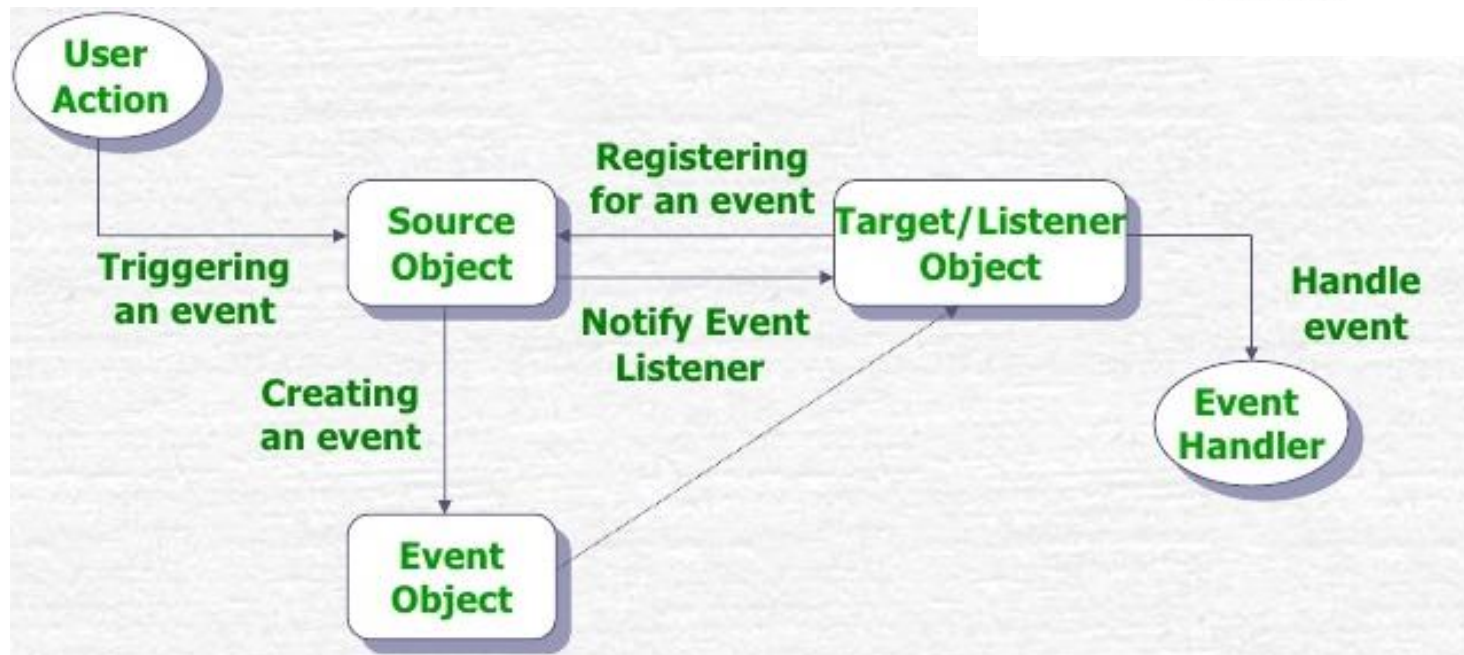
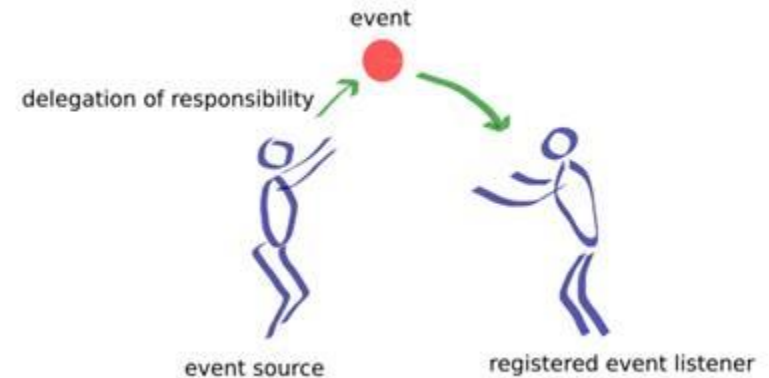
J. Jennifer Ranjani

email: jennifer.ranjani@pilani.bits-pilani.ac.in

Chamber: 6121 P, NAB

Consultation: Appointment by e-mail

Event Handling



AWT Event Handling - Example



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

class test extends Frame implements ActionListener{
    TextField tf;
    test(){
        setTitle("Core Banking");
        tf = new TextField();
        tf.setBounds(100,50,170,30);
        Button b=new Button("Submit");
        b.setBounds(100,100,100,30);
        add(b);
        add(tf);

        b.addActionListener(this);
    }
}
```

AWT Event Handling - Example



```
setSize(1000,1000);
setBackground(Color.cyan);
setLayout(null);
setVisible(true);
}
public static void main(String[] args)
{
    test t= new test();
}

public void actionPerformed(ActionEvent E)
{
    tf.setText("Welcome to Core Banking");
}
}
```

Event Handling by an External Class-Example



```
import java.awt.*;
import java.awt.event.*;
class test extends Frame{
    TextField tf;
    Label l;
    test() {
        setTitle("Core Banking");
        tf = new TextField();
        tf.setBounds(100,100,170,30);

        Button b=new Button("Submit");
        b.setBounds(100,150,100,30);

        l = new Label();
        l.setBounds(100,50,170,30);
        l.setBackground(Color.green);
```

Event Handling by an External Class-Example



```
add(b);  
add(tf);  
add(l);
```

```
AHandler a = new AHandler(this);  
b.addActionListener(a);  
tf.addTextListener(a);
```

```
setSize(1000,1000);  
setBackground(Color.cyan);  
setLayout(null);  
setVisible(true); }
```

```
public static void main(String[] args) {  
    test t= new test(); }  
}
```

Event Handling by an External Class-Example



```
class AHandler implements ActionListener,TextListener{
test obj;
AHandler(test t){
this.obj = t;
}
public void textValueChanged(TextEvent e) {
    obj.l.setText("Entered text: " + obj.tf.getText());
}

public void actionPerformed(ActionEvent E) {
    obj.l.setText("Welcome to Core Banking");}
}
```

Screen Shot



Core Banking

Welcome to Core Banking

Submit

Core Banking

Entered text: Java Program

Submit

Event Handling by Anonymous Class- Example



```
b.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent E) {  
        l.setText("Welcome to Core Banking");  
    }  
});
```

```
tf.addTextListener(new TextListener() {  
    public void textValueChanged(TextEvent E) {  
        l.setText("Entered text: " + tf.getText());  
    }  
});
```

Key AWT GUI Concepts



- Java supports three GUI frameworks
 - AWT, Swing, JavaFX
- Four Key features of AWT programs
 - Frame: Top level window is created by extending the Frame class
 - paint(): Override the paint method to display output in the window. This method is called by the run time system.
 - repaint(): Invoke the repaint method if the program needs output to be displayed. Awt program cannot call the paint method directly.
 - System.exit(): When the top level window is closed, it does not cause the program to terminate. It is necessary to handle the window-close event through a System.exit() call.

Handling Mouse Events - Example



```
import java.awt.*;
import java.awt.event.*;
public class test extends Frame implements MouseListener{
    String msg = "Welcome";
    Color c = Color.red;

    test(){
        addMouseListener(this);
        setSize(300,300);
        setLayout(null);
        setVisible(true);    }

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

Handling Mouse Events - Example



```
public void mouseExited(MouseEvent e) {  
    msg = "Mouse Exited";  
    repaint();  
}
```

```
public void mouseClicked(MouseEvent e) {  
    Graphics g=getGraphics();  
    g.setColor(Color.BLUE);  
    g.fillOval(e.getX(),e.getY(),30,30);  
}
```

```
public void mousePressed(MouseEvent e) {  
}
```

```
public void mouseReleased(MouseEvent e) {  
}
```

```
public void paint(Graphics g) {  
    g.setColor(c);  
    Font font = new Font("TimesNewRoman", Font.PLAIN, 24);  
    g.setFont(font);  
    g.drawString(msg, 50, 150);  
}
```

```
public static void main(String[] args) {  
    new test();  
}
```

Screen Shot



Handling Key Events-Example



```
import java.awt.*;
import java.awt.event.*;
public class test extends Frame implements KeyListener{
    String msg = "";

    test(){
        addKeyListener(this);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
        setBackground(Color.cyan);    }

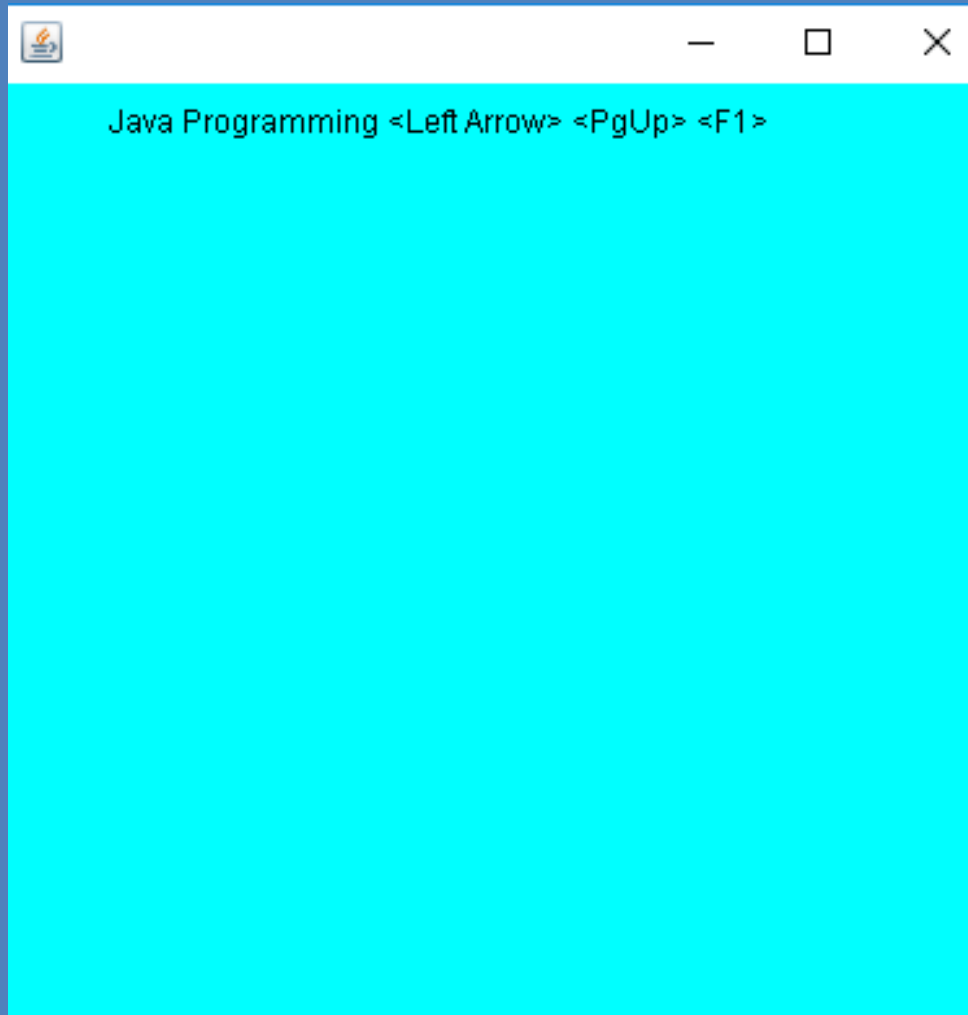
    public void keyPressed(KeyEvent e) {
        int key = e.getKeyCode();
        switch(key) {
            case KeyEvent.VK_F1: msg += "<F1>";break;
            case KeyEvent.VK_PAGE_UP: msg += "<PgUp>";break;
            case KeyEvent.VK_LEFT: msg += "<Left Arrow>";break;}
        repaint();    }
```

Handling Key Events - Example



```
public void keyReleased(KeyEvent e) {  
    repaint();  
}  
  
public void keyTyped(KeyEvent e) {  
    msg += e.getKeyChar();  
    repaint();  
}  
  
public void paint(Graphics g) {  
    g.drawString(msg, 50, 50);  
}  
  
public static void main(String[] args) {  
    new test();  
}
```

Screen shot



Review Questions



- What is a listener in context to event handling?
 - a. A listener is a variable that is notified when an event occurs
 - b. A listener is a object that is notified when an event occurs
 - c. A listener is a method that is notified when an event occurs
 - d. None of the mentioned

- Which of these events will be notified if scroll bar is manipulated?
 - a. ActionEvent
 - b. ComponentEvent
 - c. AdjustmentEvent**
 - d. WindowEvent

Review Questions



- Which of these constant value will change when the button at the end of scroll bar was clicked to increase its value?
 - a. BLOCK_DECREMENT
 - b. BLOCK_INCREMENT
 - c. UNIT_DECREMENT
 - d. UNIT_INCREMENT
- Create 3 labels, 3 text fields (Number 1, Number 2 and Result) and a Button (Enter) using AWT. Get the two numbers as input from the user and display the result when the 'Enter' Button is pressed. Handle the Key and mouse events.