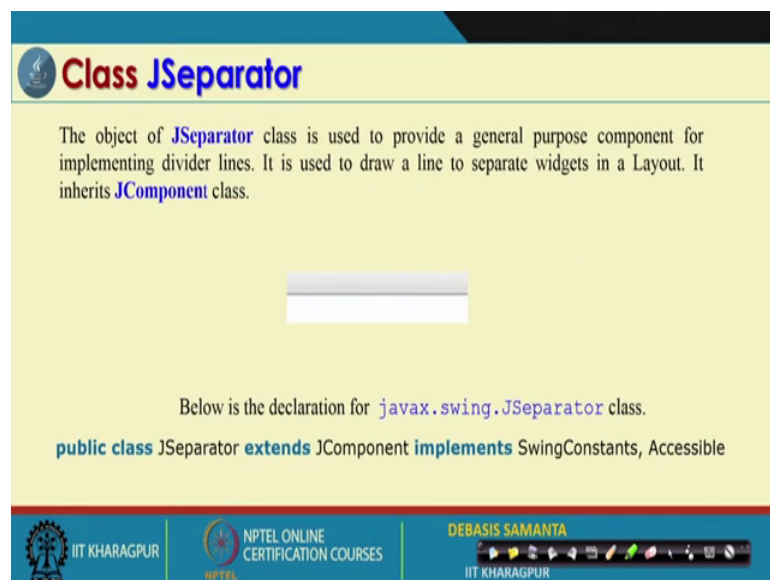


Programming in Java
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 44
Swing Programming – II


We are discussing about java swing and in the last module we have discussed some components in it, and there are few more components which we are going to cover in this module.

(Refer Slide Time: 00:33)



Class JSeparator

The object of **JSeparator** class is used to provide a general purpose component for implementing divider lines. It is used to draw a line to separate widgets in a Layout. It inherits **JComponent** class.



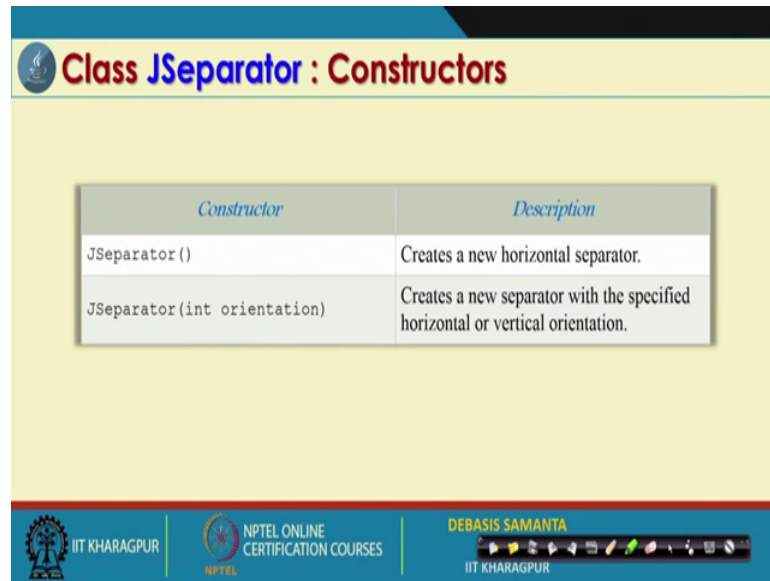
Below is the declaration for `javax.swing.JSeparator` class.

```
public class JSeparator extends JComponent implements SwingConstants, Accessible
```

The slide footer contains logos for IIT Kharagpur, NPTEL Online Certification Courses, and the name Debasis Samanta.

So, first component that here JSeparator; JSeparator is a new one which was not there in AWT and then JSeparator basically as it is shown here it looks like a separator in the sense that it will basically divide the display area into two parts with the different setting look like.

(Refer Slide Time: 00:55)

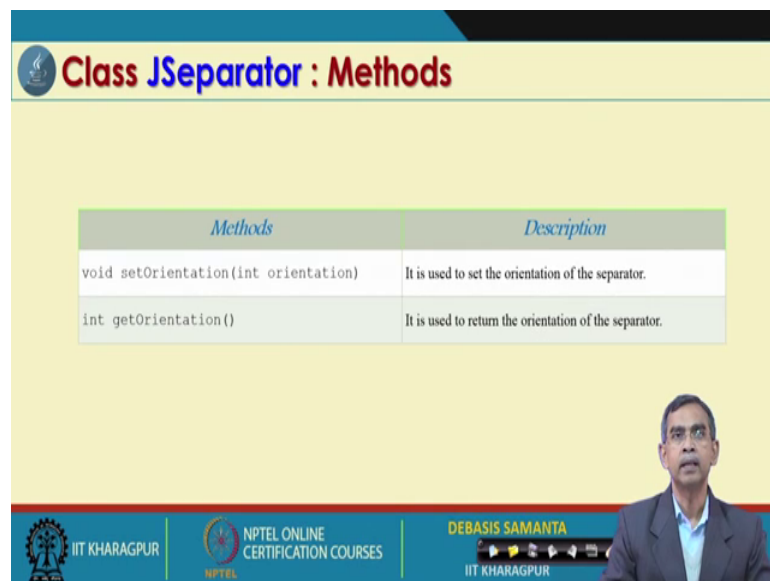


Class JSeparator : Constructors

<i>Constructor</i>	<i>Description</i>
JSeparator()	Creates a new horizontal separator.
JSeparator(int orientation)	Creates a new separator with the specified horizontal or vertical orientation.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

(Refer Slide Time: 00:57)



Class JSeparator : Methods

<i>Methods</i>	<i>Description</i>
void setOrientation(int orientation)	It is used to set the orientation of the separator.
int getOrientation()	It is used to return the orientation of the separator.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And it has two constructors and the two methods.

(Refer Slide Time: 00:59)

```
import javax.swing.*;
class SeparatorExample{
    JMenu menu, submenu;
    JMenuItem i1, i2, i3, i4, i5;
    SeparatorExample() {
        JFrame f = new JFrame("Separator Example");
        JMenuBar mb = new JMenuBar();
        menu = new JMenu("Menu");
        i1 = new JMenuItem("Item 1");
        i2 = new JMenuItem("Item 2");
        menu.add(i1);
        menu.addSeparator();
        menu.add(i2);
        mb.add(menu);
        f.setMenuBar(mb);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String args[]){
        new SeparatorExample();
    }
}
```


And there is a simple example which will show the usage of JSeparator in any window program as we see here.

So, this is basically the JSeparator. This JSeparator includes the menu and whenever you click menu some items will appear here. So, in the JSeparator some other things also can be added, we will discuss about the tab and other menus also can be added like and if you see then menu item will be displayed it is there. So, this is basically the usage of the JSeparator in your program and this program you can understand yourself and is very simple that you will be able to follow it how it is worked there.

(Refer Slide Time: 01:43)

Class JProgressBar

The **JProgressBar** class is used to display the progress of the task. It inherits **JComponent** class.



Below is the declaration for `javax.swing.JProgressBar` class.

```
public class JProgressBar extends JComponent implements SwingConstants, Accessible
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And they progress bar usually we are familiar to whenever you are some downloading or saving some file or copying some from one source to another, usually a progress bar will show that how much percentage of things that we have done.

So, progress bar is a dynamic component and this is a look of a clear progress bar. Now in your java program also you can have progress bar.

(Refer Slide Time: 02:03)

Class JProgressBar : Constructors

Constructor	Description
<code>JProgressBar()</code>	It is used to create a horizontal progress bar but no string text.
<code>JProgressBar(int min, int max)</code>	It is used to create a horizontal progress bar with the specified minimum and maximum value.
<code>JProgressBar(int orient)</code>	It is used to create a progress bar with the specified orientation, it can be either Vertical or Horizontal by using <code>SwingConstants.VERTICAL</code> and <code>SwingConstants.HORIZONTAL</code> constants.
<code>JProgressBar(int orient, int min, int max)</code>	It is used to create a progress bar with the specified orientation, minimum and maximum value.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And then you can show it.

(Refer Slide Time: 02:07)

Class JProgressBar : Methods

Method	Description
void setStringPainted(boolean b)	It is used to determine whether string should be displayed.
void setString(String s)	It is used to set value to the progress string.
void setOrientation(int orientation)	It is used to set the orientation, it may be either vertical or horizontal by using SwingConstants.VERTICAL and SwingConstants.HORIZONTAL constants.
void setValue(int value)	It is used to set the current value on the progress bar.

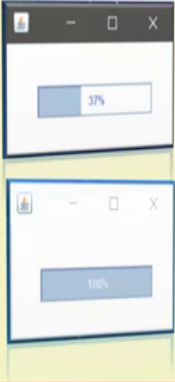
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, there are 4 constructor to, do it and then it has some methods by which different information about the progress bar can be extracted.

(Refer Slide Time: 02:11)

Java JProgressBar : An example

```
import javax.swing.*;
public class ProgressBarExample extends JFrame{
    JProgressBar jb; int i=0,num=0;
    ProgressBarExample(){
        jb=new JProgressBar(0,2000);
        jb.setBounds(40,40,160,30);
        jb.setValue(0);
        jb.setStringPainted(true);
        add(jb);
        setSize(250,150); setLayout(null);
    }
    public void iterate(){
        while(i<=2000){
            jb.setValue(i);
            i=i+20;
            try{Thread.sleep(150);}catch(Exception e){}
        }
    }
    public static void main(String[] args) {
        ProgressBarExample m=new ProgressBarExample();
        m.setVisible(true);
        m.iterate();
    }
}
```

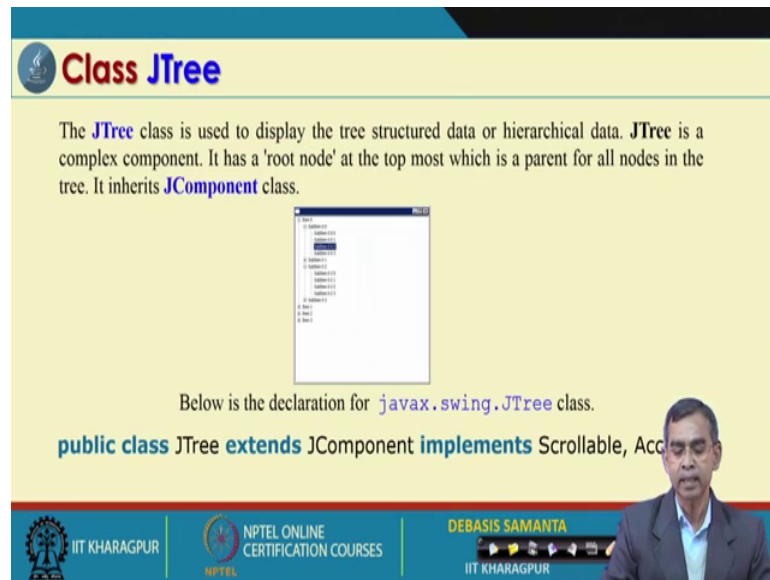


IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And this is a simple example by which you can show we have given the two outputs like in the middle way or in the progress bar is working, and when the finish job is there. So, this is a simple example that you can try and you can see how it will work in your program.


And then JTree this is very interesting one a swing components. A JTree actually we are also familiar to these things whenever we browse a directory and then that. So, it basically shows what are the subdirectories under a directory what are the files under the subdirectories like.

(Refer Slide Time: 02:51)



Class JTree

The **JTree** class is used to display the tree structured data or hierarchical data. **JTree** is a complex component. It has a 'root node' at the top most which is a parent for all nodes in the tree. It inherits **JComponent** class.



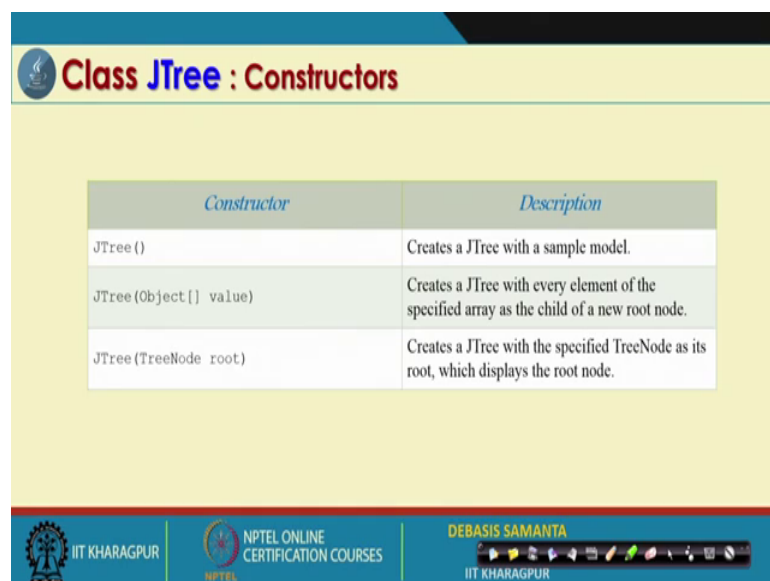
Below is the declaration for `javax.swing.JTree` class.

```
public class JTree extends JComponent implements Scrollable, Acc...
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, the entire tree will look like this, it can be maintained using a component class only from which the information can be obtained from the drive.

(Refer Slide Time: 03:01)



Class JTree : Constructors

Constructor	Description
<code>JTree ()</code>	Creates a JTree with a sample model.
<code>JTree (Object[] value)</code>	Creates a JTree with every element of the specified array as the child of a new root node.
<code>JTree (TreeNode root)</code>	Creates a JTree with the specified <code>TreeNode</code> as its root, which displays the root node.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, it has the 3 constructors.

(Refer Slide Time: 03:05)

And it is simple example which basically shows how JTree can be shown can be used as we see here, we declare is this is basically the one we can say the style is the one root under this root there is a two nodes it is nodes under this again nodes there are subnodes actually.

Now, you see how we can create it, is basically style it is basically the default mutable mu t node this is the file it is there in class which is declared there in swing package; we have to use it and then we can create style and then we can create other what is called the menu t node like say color and font.

Now, here you see style dot add color, that mean we add color as a part of the style and then add font as a part of the style like style under style color and under style font. Then we can add some other mu note under this color. So, it is basically we define it and under this color dot at rate, this means we go into the color rate. So, this way tree of any branches after the another branches can be created. So, this is the simple example which basically tell you how the JJavaTree can be crated whenever it is required in your program.

So, this is the idea about JTree and the next is JColorChooser this is also very interesting probably you know whenever you use any word application like MS word and whenever you have to decide the font of a color of a font, then you have to choose it and then color can be chosen from a color pen.

(Refer Slide Time: 04:49)

Class JColorChooser

The **JColorChooser** class is used to create a color chooser dialog box so that user can select any color. It inherits **JComponent** class.



Below is the declaration for `javax.swing.JColorChooser` class

```
public class JColorChooser extends JToggleButton implements ActionListener
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, here is a color pen look like and you can choose any one color and then that color can be settle there. Instead of writing color dot blue color dot green like any other color can be added here

(Refer Slide Time: 05:09)

Class JColorChooser : Constructors

Constructor	Description
<code>JColorChooser()</code>	It is used to create a color chooser panel with white color initially.
<code>JColorChooser(color initialcolor)</code>	It is used to create a color chooser panel with the specified color initially.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, by default it will have this is the color chosen which say 200 colors like you can add more colors into it also there is a pro facilities those things can be done by this constructor.

(Refer Slide Time: 05:13)

Class JColorChooser : Methods

Method	Description
<code>void addChooserPanel(AbstractColorChooserPanel panel)</code>	It is used to add a color chooser panel to the color chooser.
<code>static Color showDialog(Component c, String title, Color initialColor)</code>	It is used to show the color chooser dialog box.



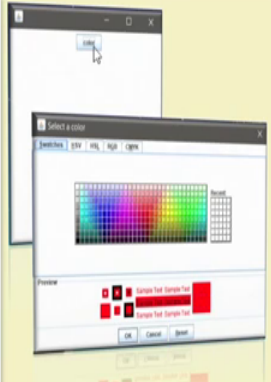
IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

And there is a method by which the color chooser items can which color it has been chosen the value of that color can be used there.

(Refer Slide Time: 05:19)

Java JColorChooser : An example

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
public class ColorChooserExample extends JFrame implements ActionListener {
    JButton b;Container c;
    ColorChooserExample(){
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        c.setContentPane(new JPanel());
        c.setLayout(new BorderLayout());
        b=new JButton("Color");
        b.addActionListener(this);
        c.add(b);
    }
    public void actionPerformed(ActionEvent e) {
        Color initialColor=Color.RED;
        Color color=JColorChooser.showDialog(this,"Select a color",initialColor);
        c.setBackground(color);
    }
    public static void main(String[] args) {
        ColorChooserExample ch=new ColorChooserExample();
        ch.setSize(400,400);
        ch.setVisible(true);
        ch.setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

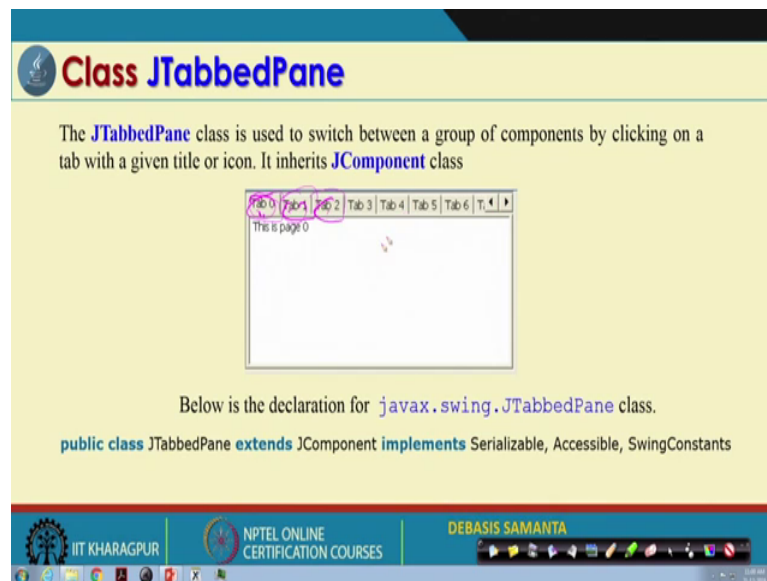


IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

So, again this is a simple program for practice you can practice this program you can see how the color can be chosen, here is basically one button is added and using this button the action handler is there. If you click it the color chooser option will popup and then from there you can select some color and the color will be selected and using that color if you write something if you draw something, that will be with that color and then

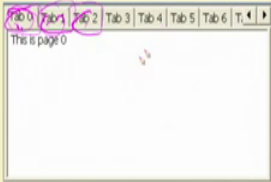
color can be reset. And here is also see whatever the recent color that you have used that also will be displayed to you. It is a very similar to it is in the context of say micro (Refer Time: 05:57) or this kind of things it is there the same concept it can. So, using this color chooser you can develop our very sophisticated and smart program in this program.

(Refer Slide Time: 06:09)



Class JTabbedPane

The **JTabbedPane** class is used to switch between a group of components by clicking on a tab with a given title or icon. It inherits **JComponent** class



Below is the declaration for `javax.swing.JTabbedPane` class.

```
public class JTabbedPane extends JComponent implements Serializable, Accessible, SwingConstants
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA

And then tabbed pane is basically essential one component, which is also new not there in the AWT and within these things it basically if you create a frame, in this frame lot of other tabs is there. If you know excel and you know probably at the bottom, there are lot of tabs will be there that tabs corresponding the page 1 page 2 page 2 or more probably it is called the seat 1 seat 2 seat 3 like the one. So, these are the tab and you can rename this tab. So, this name is seat 1 this is seat 2 or say photo album 1 album 2 album 3 or whatever it is there.

So, it basically helps to develop a program develop a windows program which to organize the different menu items on the frame itself and it can be placed again the constructor is there, the tab can be placed either in the top or usually bottom. So, usually the by default the top tabbed pane is there and then otherwise you can customize into according to your.

(Refer Slide Time: 07:05)

Class JTabbedPane : Constructors

Constructor	Description
JTabbedPane ()	Creates an empty TabbedPane with a default tab placement of JTabbedPane.Top.
JTabbedPane (int tabPlacement)	Creates an empty TabbedPane with a specified tab placement.
JTabbedPane (int tabPlacement, int tabLayoutPolicy)	Creates an empty TabbedPane with a specified tab placement and tab layout policy.


IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, here is basically JTabbedPane and then JTabbedPane in tab placement in it is basically is the placement the 0 on these are the 2 things are there, otherwise in the left side also items also 0 1 2 3 these are the different value that can be mentioned and this basically table layout policy; that means, the tabbed pane if it is there if you want to add more items into it, and then what is the layout manage that you can follow.

(Refer Slide Time: 07:29)

Java JTabbedPane : An example

```
import javax.swing.*;
public class TabbedPaneExample {
    JFrame f;
    TabbedPaneExample() {
        f=new JFrame();
        JTextArea ta=new JTextArea(200,200);
        JPanel p1=new JPanel();
        p1.add(ta);
        JPanel p2=new JPanel();
        JPanel p3=new JPanel();
        JTabbedPane tp=new JTabbedPane();
        tp.setBounds(50,50,700,200);
        tp.add("main",p1);
        tp.add("visit",p2);
        tp.add("help",p3);
        f.add(tp);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String[] args) {
        new TabbedPaneExample();
    }
}
```



IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And it has the simple example for your practice again, you can see how the tabbed pane will appear into a frame which is included in a in a container.

(Refer Slide Time: 07:43)

Class JSlider

The Java **JSlider** class is used to create the slider. By using **JSlider**, a user can select a value from a specific range.

Setting:

0 10 20 30 40 50

OK

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And then JSlider is another one example which is also knew which is not there is AWT, and as you see this is a basically is usual look of a slider using the mouse you can just select a slider and then slider can be dragged to show that this is a position that you want to have. So, this way the different value can be entered into the system and if you press at the position the slider has released, it will basically get the same value and it return to the system.

(Refer Slide Time: 08:13)

Class JSlider : Constructors

Constructor	Description
<code>JSlider()</code>	creates a slider with the initial value of 50 and range of 0 to 100.
<code>JSlider(int orientation)</code>	creates a slider with the specified orientation set by either <code>JSlider.HORIZONTAL</code> or <code>JSlider.VERTICAL</code> with the range 0 to 100 and initial value 50.
<code>JSlider(int min, int max)</code>	creates a horizontal slider using the given min and max.
<code>JSlider(int min, int max, int value)</code>	creates a horizontal slider using the given min, max and value.
<code>JSlider(int orientation, int min, int max, int value)</code>	creates a slider using the given orientation, min, max and value.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

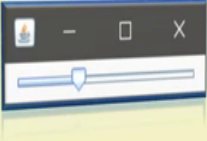
It has the few constructor and then methods.

(Refer Slide Time: 08:17)

Java JSlider : An example

```
import javax.swing.*;
public class SliderExample1 extends JFrame{
public SliderExample1() {
JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 50, 25);
JPanel panel=new JPanel();
panel.add(slider);
add(panel);
}

public static void main(String s[]) {
SliderExample1 frame=new SliderExample1();
frame.pack();
frame.setVisible(true);
}
}
```



DEBASIS SAMANTA
IIT KHARAGPUR


And this is the one simple program for your practice, you can run it and you can see this how it will work and then you can click using mouse you can drag and move the slider into any positions and release it like that.

And then JSpinner is also similar to the slider, but here actually the numerical value is a continuous range or some discrete and both can be defined can be utilized here.

(Refer Slide Time: 08:39)

Class JSpinner

The object of **JSpinner** class is a single line input field that allows the user to select a number or an object value from an ordered sequence.



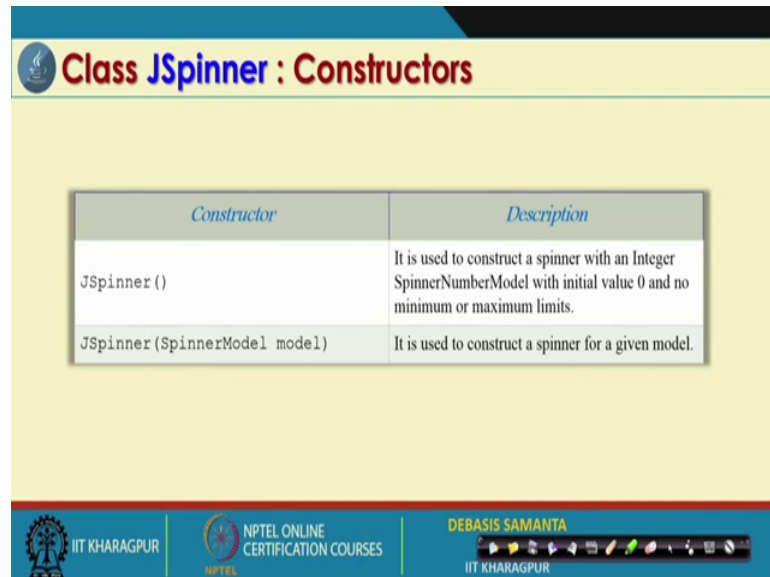
Below is the declaration for `javax.swing.JSpinner` class.

```
public class JSpinner extends JComponent implements Accessible
```

DEBASIS SAMANTA
IIT KHARAGPUR

Here is basically it is like look like whenever you have to select font size usually you know in the Microsoft word or other word documents processor, there is basically this kind of spinner is there. If you click it then it will change it, if you click if you keep on clicking there is a value automatically will increase and then it will release it then that value will be selected and then if you press it then that value is selected like this.

(Refer Slide Time: 09:05)



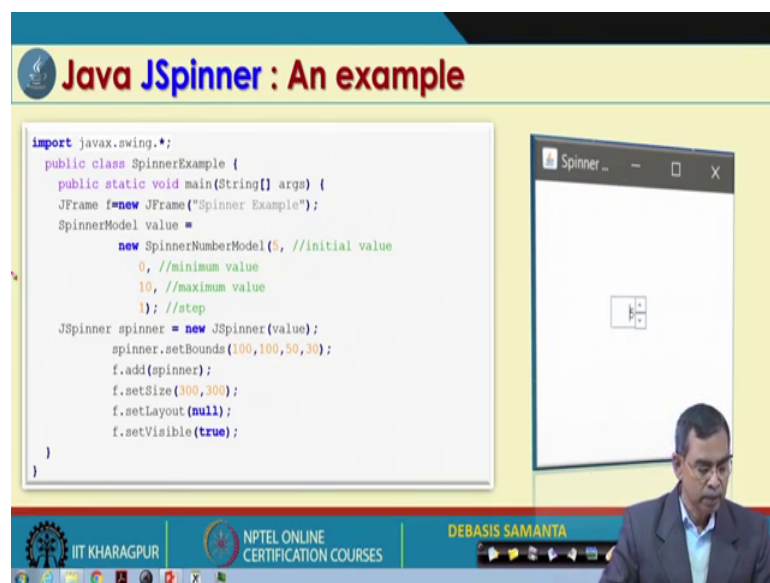
Class JSpinner : Constructors

Constructor	Description
JSpinner ()	It is used to construct a spinner with an Integer SpinnerNumberModel with initial value 0 and no minimum or maximum limits.
JSpinner (SpinnerModel model)	It is used to construct a spinner for a given model.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, this is a leisure look of a JSpinner and it has the two constructors and two methods.

(Refer Slide Time: 09:09)



Java JSpinner : An example

```
import javax.swing.*;

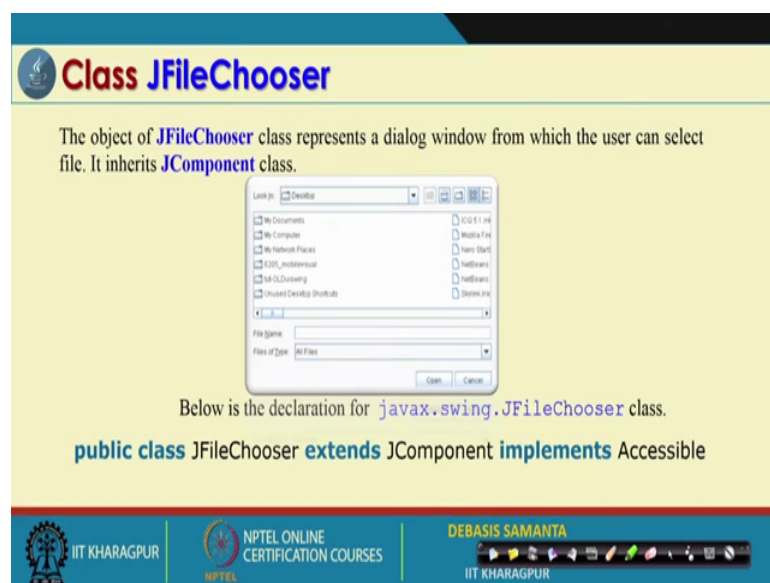
public class SpinnerExample {
    public static void main(String[] args) {
        JFrame f=new JFrame("Spinner Example");
        SpinnerModel value =
            new SpinnerNumberModel(5, //initial value
                0, //minimum value
                10, //maximum value
                1); //step
        JSpinner spinner = new JSpinner(value);
        spinner.setBounds(100,100,50,30);
        f.add(spinner);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

Spinner ...

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

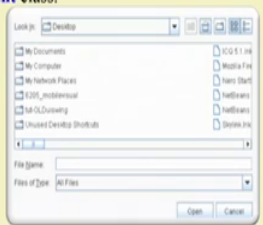
And this is the one simple example by which a JSpinner with some default values can be planned and you can change the different values, there we can see the JSpinner whenever it is display you with initial value phi and it has range 0 to 10. So, we can decide what is the range. So, for example, 0 to 100 also we can decide it and this is basically the step; that means, how much incremention 0 1 like this 1 if we say 0.5; that means, the in rate of increment that can be specified here. So, this is the different way the JSpinner can be controlled you can you can run this program with different settings and then you can get how the JSpinner will look like this here.

(Refer Slide Time: 09:51)



Class JFileChooser

The object of **JFileChooser** class represents a dialog window from which the user can select file. It inherits **JComponent** class.



Below is the declaration for `javax.swing.JFileChooser` class.

```
public class JFileChooser extends JComponent implements Accessible
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And JFileChooser is basically whenever you have to develop a system program like and then JFileChooser is really more preferable. Here basically whatever the files are there in the current directory it will select it and then it will display and then it will allow to generate many events by selecting a particular file and then saving it or copying it or opening it whatever it is there. So, this basically look like the component according to JFileChooser in java swing.

(Refer Slide Time: 10:21)

Class JFileChooser : Constructors

Constructor	Description
JFileChooser ()	Constructs a JFileChooser pointing to the user's default directory.
JFileChooser (File currentDirectory)	Constructs a JFileChooser using the given File as the path.
JFileChooser (String currentDirectoryPath)	Constructs a JFileChooser using the given path.

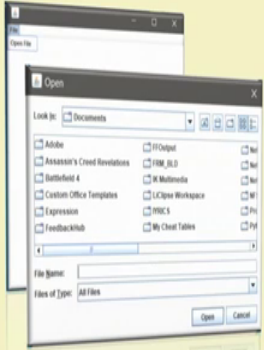
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

It has four constructors.

(Refer Slide Time: 10:25)

Java JFileChooser : An example

```
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
public class FileChooserExample extends JFrame implements ActionListener{
    JMenuBar mb;
    JMenuItem open;
    JMenuItem open2;
    JTextArea ta;
    FileChooserExample(){
        open=new JMenuItem("Open File");
        open.addActionListener(this);
        file=new JMenuItem("File");
        file.add(open);
        mb=new JMenuBar();
        mb.setBounds(0,0,800,20);
        mb.add(file);
        ta=new JTextArea(800,800);
        ta.setBounds(0,20,800,800);
        add(mb);
        add(ta);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==open) {
```



IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And this is a simple program that you can check of your own, these are the different folder that you have included here in this folder whatever the files it is there it basically will be displayed there. So, you can try it and then you can have fun regarding running and you see how it is there. If you specify the different directory according to the directory whatever the files it is there, that files will be displayed automatically.

(Refer Slide Time: 10:49)

Class JToggleButton

JToggleButton is used to create toggle button, it is two-states button to switch on or off.

Nested Classes

Modifier and Type	Class	Description
protected class	JToggleButton.AccessibleJToggleButton	This class implements accessibility support for the JToggleButton class.
static class	JToggleButton.ToggleButtonModel	The ToggleButton model

DEBASIS SAMANTA
IIT KHARAGPUR

And then JToggleButton it is also button, but the difference that it has only toggling either on off or open closed or male female this kind of things are there. So, binary pattern it is there.

(Refer Slide Time: 11:03)

Class JToggleButton : Constructor

Constructor	Description
JToggleButton ()	It creates an initially unselected toggle button without setting the text or image.
JToggleButton (Action a)	It creates a toggle button where properties are taken from the Action supplied.
JToggleButton (Icon icon)	It creates an initially unselected toggle button with the specified image but no text.
JToggleButton (Icon icon, boolean selected)	It creates a toggle button with the specified image and selection state, but no text.
JToggleButton (String text)	It creates an unselected toggle button with the specified text.
JToggleButton (String text, boolean selected)	It creates a toggle button with the specified text and selection state.
JToggleButton (String text, Icon icon)	It creates a toggle button that has the specified text and image, and that is initially unselected.
JToggleButton (String text, Icon icon, boolean selected)	It creates a toggle button with the specified text, image, and selection state.

DEBASIS SAMANTA
IIT KHARAGPUR

(Refer Slide Time: 11:05)

Class JToggleButton : Methods

Modifier and Type	Method	Description
AccessibleContext	getAccessibleContext ()	It gets the AccessibleContext associated with this JToggleButton.
String	getUIClassID ()	It returns a string that specifies the name of the l&f class that renders this component.
protected String	paramString ()	It returns a string representation of this JToggleButton.
void	updateUI ()	It resets the UI property to a value from the current look and feel.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

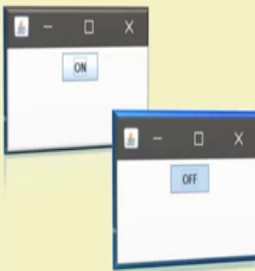
It has two constructors so, many constructor and then methods.

(Refer Slide Time: 11:07)

Java JToggleButton : An example

```
import java.awt.FlowLayout;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import javax.swing.JFrame;
import javax.swing.JToggleButton;

public class ToggleButtonExample extends JFrame implements
ItemListener {
    public static void main(String[] args) {
        new ToggleButtonExample();
    }
    private JToggleButton button;
    ToggleButtonExample() {
        setTitle("JToggleButton with ItemListener Example");
        setLayout(new FlowLayout());
        setJToggleButton();
        setAction();
        setSize(200, 200);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And it is an example as we see we have created one button and then it has two fields on and off and how this if you click it here, automatically on will be toggled to off and whenever it is off it is click it or toggle to on. So, toggling is occurs whenever there is an event occurs and then it can bis that button value will be automatically say it accordingly and then that value will be used for some other requirement that is there in your program.

(Refer Slide Time: 11:39)

Class JToolBar

JToolBar container allows us to group other components, usually buttons with icons in a row or column. **JToolBar** provides a component which is useful for displaying commonly used actions or controls.

Nested Classes

Modifier and Type	Class	Description
protected class	JToolBar.AccessibleJToolBar	This class implements accessibility support for the JToolBar class.
static class	JToolBar.Separator	A toolbar-specific separator.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And then JToolBar it is very similar to the menu item itself it looks it has the two constructors.

(Refer Slide Time: 11:45)

Class JToolBar : Methods

Modifier and Type	Method	Description
JButton	add(Action a)	It adds a new JButton which dispatches the action.
protected void	addImpl(Component comp, Object constraints, int index)	If a JButton is being added, it is initially set to be disabled.
void	addSeparator()	It appends a separator of default size to the end of the tool bar.
protected PropertyChangeListener	createActionChangeListener(JButton b)	It returns a properly configured PropertyChangeListener which updates the control as changes to the Action occur, or null if the default property change listener for the control is desired.
protected JButton	createActionComponent(Action a)	Factory method which creates the JButton for Actions added to the JToolBar.
ToolBarUI	getUI()	It returns the tool bar's current UI.
void	setUI(ToolBarUI ui)	It sets the L&F object that renders this component.
void	setOrientation(int o)	It sets the orientation of the tool bar.



IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And then so, many methods are there.

(Refer Slide Time: 11:47)

Class JToolBar : Methods

Modifier and Type	Method	Description
JButton	add(Action a)	It adds a new JButton which dispatches the action.
protected void	addImpl(Component comp, Object constraints, int index)	If a JButton is being added, it is initially set to be disabled.
void	addSeparator()	It appends a separator of default size to the end of the tool bar.
protected PropertyChangeListener	createActionChangeListener(JButton b)	It returns a properly configured PropertyChangeListener which updates the control as changes to the Action occur, or null if the default property change listener for the control is desired.
protected JButton	createActionComponent(Action a)	Factory method which creates the JButton for Actions added to the JToolBar.
ToolBarUI	getUI()	It returns the tool bar's current UI.
void	setUI(ToolBarUI ui)	It sets the L&F object that renders this component.
void	setOrientation(int o)	It sets the orientation of the tool bar.

DEBASIS SAMANTA
IIT KHARAGPUR

(Refer Slide Time: 11:49)



Java JToolBar : An example

```

import java.awt.BorderLayout;
import java.awt.Container;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JToolBar;

public class ToolBarExample {
    public static void main(final String args[]) {
        JFrame myframe = new JFrame("JToolBar Example");
        myframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JToolBar toolbar = new JToolBar();
        toolbar.setRollover(true);
    }
}

```

DEBASIS SAMANTA
IIT KHARAGPUR

This is an example as we see the file edit option 1 like in under this option 1 so, many other things are there. So, it is called the tool bar like it is just similar to the menu bar, but it is special in MS toolbar and we can use it. So, as it is used to develop some tools that is the it is the name is toolbar.

(Refer Slide Time: 12:07)

Class JViewport

The **JViewport** class is used to implement scrolling. **JViewport** is designed to support both logical scrolling and pixel-based scrolling. The viewport's child, called the view, is scrolled by calling the **JViewport.setViewPosition()** method.

Nested Classes

<i>Modifier and Type</i>	<i>Class</i>	<i>Description</i>
protected class	JViewport.AccessibleJViewport	This class implements accessibility support for the Jviewport class.
protected class	JViewport.ViewListener	A listener for the view.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And JViewport it is just like a text area look like, but it is java or it is also very similar to the panel also has a special area that can be viewed in a separate way like.

(Refer Slide Time: 12:23)

Class JViewport : Fields

<i>Modifier and Type</i>	<i>Field</i>	<i>Description</i>
static int	BACKINGSTORE_SCROLL_MODE	It draws viewport contents into an offscreen image.
protected Image	backingStoreImage	The view image used for a backing store.
static int	BLIT_SCROLL_MODE	It uses graphics.copyArea to implement scrolling.
protected boolean	isViewSizeSet	True when the viewport dimensions have been determined.
protected Point	lastPaintPosition	The last viewPosition that we've painted, so we know how much of the backing store image is valid.
protected boolean	scrollUnderway	The scrollUnderway flag is used for components like JList.
static int	SIMPLE_SCROLL_MODE	This mode uses the very simple method of redrawing the entire contents of the scrollpane each time it is scrolled.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And it has the two classes two nested class in it and then it has the two different values and then constructor.

(Refer Slide Time: 12:29)

Class JViewport : Constructors

Constructor	Description
JViewport ()	Creates a JViewport.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

(Refer Slide Time: 12:31)

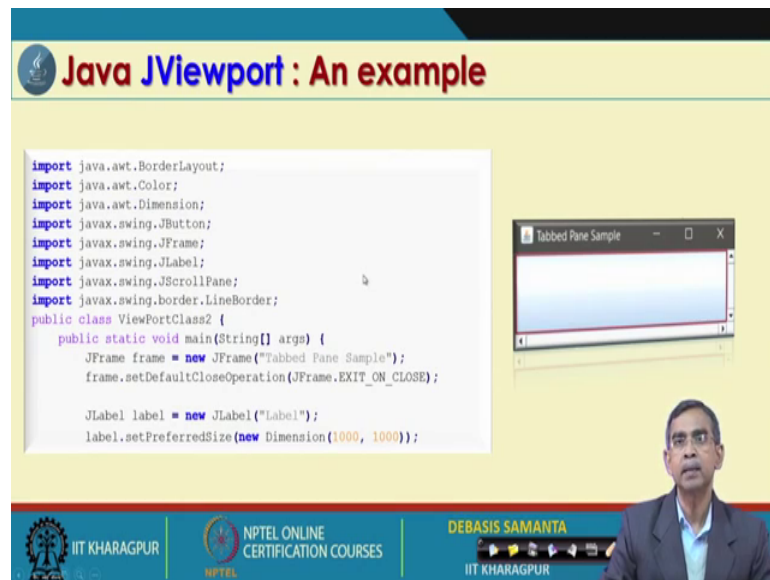
Class JViewport : Methods

Modifier and Type	Method	Description
void	addChangeListener (ChangeListener l)	It adds a ChangeListener to the list that is notified each time the view's size, position, or the viewport's extent size has changed.
protected LayoutManager	createLayoutManager ()	Subclassers can override this to install a different layout manager (or null) in the constructor.
protected Jviewport.ViewListener	createViewListener ()	It creates a listener for the view.
int	getScrollMode ()	It returns the current scrolling mode.
Component	getView ()	It returns the Jviewport's one child or null.
Point	getViewPosition ()	It returns the view coordinates that appear in the upper left hand corner of the viewport, or 0,0 if there's no view.
Dimension	getViewSize ()	If the view's size hasn't been explicitly set, return the preferred size, otherwise return the view's current size.
void	setExtentSize (Dimension newExtent)	It sets the size of the visible part of the view using view coordinates.
void	setScrollMode (int mode)	It used to control the method of scrolling the viewport contents.
void	setViewSize (Dimension newSize)	It sets the size of the view.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

Only one constructor and there are so, many methods.

(Refer Slide Time: 12:33)



The slide is titled "Java JViewport : An example". It features a code editor on the left with the following Java code:

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.border.LineBorder;

public class ViewPortClass2 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Tabbed Pane Sample");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel label = new JLabel("Label");
        label.setPreferredSize(new Dimension(1000, 1000));
    }
}
```

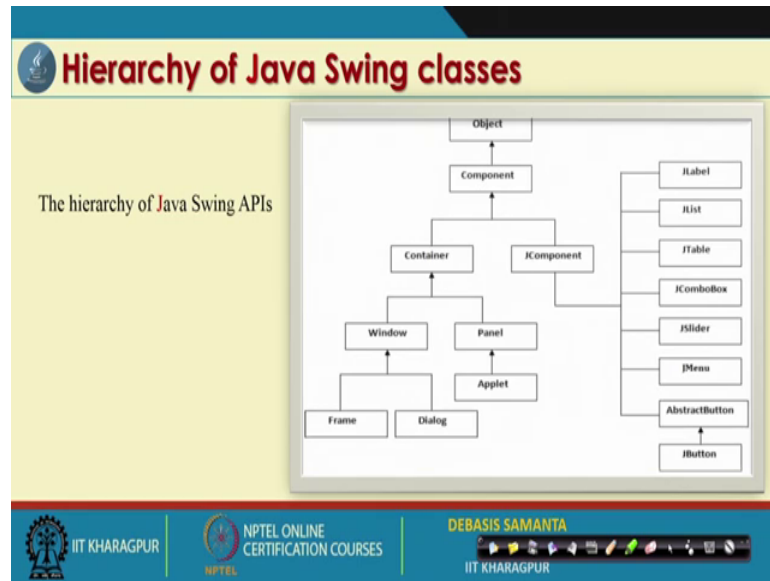
On the right side of the slide, there is a screenshot of a Java Swing window titled "Tabbed Pane Sample". The window contains a single tab with a light blue background and a white border. In the bottom right corner of the slide, there is a small video inset of the presenter, Debasis Samanta, with the text "DEBASIS SAMANTA" and "IIT KHARAGPUR" below it.

And it is an example which basically display the view port and then the background everything can be chosen there and the different setting style the different theme also can be included and those things you can use the different constructor to do that.

So, you can try with same thing here I have used the default constructor, but my advice is that you can try with the different constructor that I have already mentioned and then on this program. It is applicable to all examples that I have instead there I have used only one constructor, but for your practice for your learning you should try all these things with the different constructors. And the different method also you can call once this object is created you can see what is the method returns and then you can try to print this method on the windows or you can try to print the method using (Refer Time: 13:23) into the same frame like this one.

So, you can understand that what they are doing for you actually, and then this is just starting point. You have to ultimately you have to use to develop your program that you can decide one project and then in for that project you can think that what you want to have and according try to realize it, and that is all basically this it the way of learning this kind of concept it is there.

(Refer Slide Time: 13:49)



So, this is the j view point and then I will come to the container class there as you see this is the component. There are so, many components we have covered in the this module as well in the last module and then. So, for the container is concerned there are 3 things are there frame panel dialog box and 1 container is also applet. So, applet also can be added here and although we have not used any example for that, but it is basically if you create one program which extends an applet, and whatever the component you want to add can be added into the applet also which we have already given the enough illustration discussion in the while we are discussing about AWT. So, applet is very same we just ignore it here in this discussion

Anyway. So, for the container is concerned as we see container includes window, it can input panel and also frame dialog and it can also include applet. So, these are the full what is called the I mean concept. So, for the container in the java swing is concerned there, and all those things are there in AWT itself. So, it is an addition with some better what is called the look and feeling.

(Refer Slide Time: 14:57)

Class JPanel

The **JPanel** is a simplest container class. It provides space in which an application can attach any other component. It inherits the **JComponent** class.

It doesn't have title bar.

Below is the declaration for `javax.swing.JPanel` class.

```
public class JPanel extends JComponent implements Accessible
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

Now, let us see what is the JPanel it is there and as we see a panel can includes within another panel and so, on so, on. So, it can be nested that is why it is called the nested facility which was not there in AWT of course.

(Refer Slide Time: 15:09)

Class JPanel : Constructors

Constructor	Description
<code>JPanel ()</code>	It is used to create a new JPanel with a double buffer and a flow layout.
<code>JPanel (boolean isDoubleBuffered)</code>	It is used to create a new JPanel with FlowLayout and the specified buffering strategy.
<code>JPanel (LayoutManager layout)</code>	It is used to create a new JPanel with the specified layout manager.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And it has the constructor like panel and then it has one pa argument w buffer; that means, all the contents can be stored and the layout manager means which layout manger if you want to add many more items into the panel itself, then which it the layout manger that you have to follow. Layout manager concept is same as the layout manager that we

have discussed in J java dot AWT; that means, flow layout water layout grid layout char layout all those things are applicable here also.

That is why they are they do not have any explicit or special bearing in java swing that is why it is not included here. And this is an example about how the panel can be there as we see the here entirely this is basically a frame this is the entire frame, in this frame the panel is there.

(Refer Slide Time: 15:47)

The slide is titled "Java JPanel : An example". On the left, there is a code editor showing the following Java code:

```
import java.awt.*;
import javax.swing.*;
public class PanelExample {
    PanelExample() {
        JFrame f= new JFrame("Panel Example");
        JPanel panel=new JPanel();
        panel.setBounds(50,50,200,100);
        panel.setBackground(Color.gray);
        JButton b1=new JButton("Button 1");
        b1.setBounds(50,100,80,30);
        b1.setBackground(Color.yellow);
        JButton b2=new JButton("Button 2");
        b2.setBounds(100,100,80,30);
        b2.setBackground(Color.green);
        panel.add(b1); panel.add(b2);
        f.add(panel);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String arg[]) {
        new PanelExample();
    }
}
```

On the right, a screenshot of a Java Swing window titled "Panel Example" is shown. The window contains a grey JPanel with two buttons: "Button 1" (yellow background) and "Button 2" (green background). A presenter is visible in the bottom right corner of the slide.

And in this panel two buttons are added here. So, this is the example and the; for this output the program will look like as it is shown here. So, you can run this program and you can see the output, and you can change many parameters in the a program that we have used there and then you can see it. For example, the background of the panel can be changed and then button instead of button you can add some other components into it whatever it is there.

You can do a lot of experiments, and starting this program you can start changing many things and they learn it. So, this is the idea actually that I want to mention.

(Refer Slide Time: 16:35)

Class JFrame

The `javax.swing.JFrame` class is a type of container which inherits the `java.awt.Frame` class. **JFrame** works like the main window where components like labels, buttons, textfields are added to create a GUI.

Unlike `Frame`, **JFrame** has the option to hide or close the window with the help of `setDefaultCloseOperation(int)` method.

Nested Class

<i>Modifier and Type</i>	<i>Class</i>	<i>Description</i>
protected class	<code>JFrame.AccessibleJFrame</code>	This class implements accessibility support for the <code>JFrame</code> class.

DEBASIS SAMANTA
IIT KHARAGPUR

And so, this is the panel and then like frame we have already familiar to frame concept there is an AWT is a same concept it is here also. It is also nested class means frame is defined we inside the component class that is why it is called the nested class.

(Refer Slide Time: 16:49)

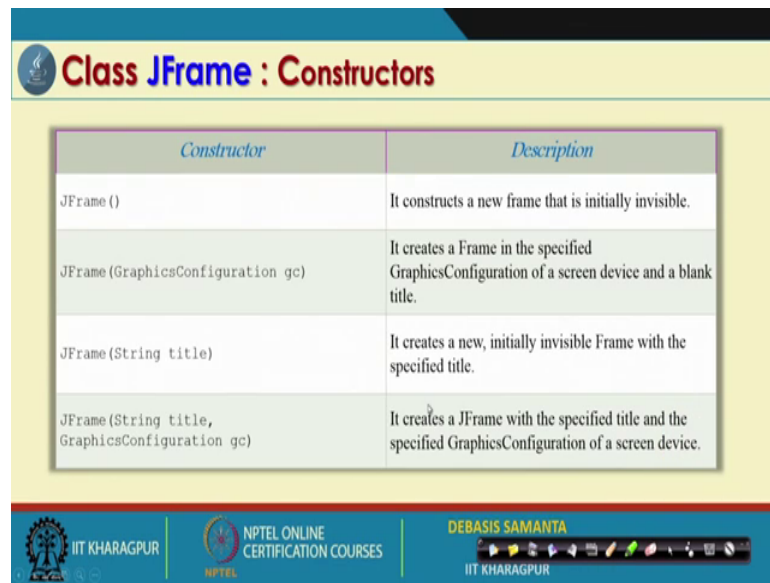
Class JFrame : Fields

<i>Modifier and Type</i>	<i>Field</i>	<i>Description</i>
protected <code>AccessibleContext</code>	<code>accessibleContext</code>	The accessible context property.
static int	<code>EXIT_ON_CLOSE</code>	The exit application default window close operation.
protected <code>JRootPane</code>	<code>rootPane</code>	The <code>JRootPane</code> instance that manages the <code>contentPane</code> and optional <code>menuBar</code> for this frame, as well as the <code>glassPane</code> .
protected boolean	<code>rootPaneCheckingEnabled</code>	If true then calls to <code>add</code> and <code>setLayout</code> will be forwarded to the <code>contentPane</code> .

DEBASIS SAMANTA
IIT KHARAGPUR

It has some fields like exit on close root pane root pane checking enable.

(Refer Slide Time: 16:55)



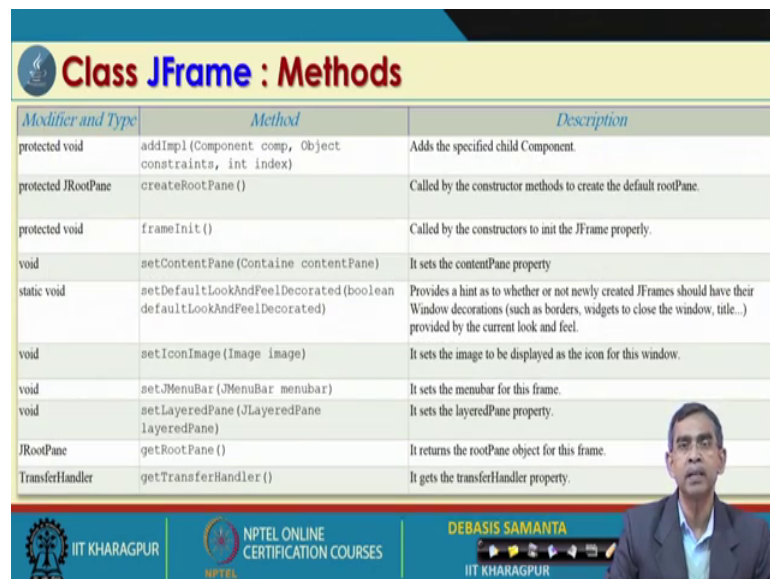
<i>Constructor</i>	<i>Description</i>
<code>JFrame ()</code>	It constructs a new frame that is initially invisible.
<code>JFrame(GraphicsConfiguration gc)</code>	It creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title.
<code>JFrame(String title)</code>	It creates a new, initially invisible Frame with the specified title.
<code>JFrame(String title, GraphicsConfiguration gc)</code>	It creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

These are the fields and it has the four constructor for the different way the JFrame can be initialized actually; all constructors as per initializing the object whenever you create it.

So, there are default constructor and then some graphics can be added it basically this graphics is for style, a frame can be with different style shading color background all these things are there. The frame can be added using some title also and both title and then configure and also it is there frame can be without any title like this one.

(Refer Slide Time: 17:23)



<i>Modifier and Type</i>	<i>Method</i>	<i>Description</i>
protected void	<code>addImpl(Component comp, Object constraints, int index)</code>	Adds the specified child Component.
protected JRootPane	<code>createRootPane ()</code>	Called by the constructor methods to create the default rootPane.
protected void	<code>frameInit()</code>	Called by the constructors to init the JFrame properly.
void	<code>setContentPane(Container contentPane)</code>	It sets the contentPane property
static void	<code>setDefaultLookAndFeelDecorated(boolean defaultLookAndFeelDecorated)</code>	Provides a hint as to whether or not newly created JFrames should have their Window decorations (such as borders, widgets to close the window, title...) provided by the current look and feel.
void	<code>setIconImage(Image image)</code>	It sets the image to be displayed as the icon for this window.
void	<code>setMenuBar(JMenuBar menubar)</code>	It sets the menubar for this frame.
void	<code>setLayeredPane(JLayeredPane layeredPane)</code>	It sets the layeredPane property.
JRootPane	<code>getRootPane ()</code>	It returns the rootPane object for this frame.
TransferHandler	<code>getTransferHandler ()</code>	It gets the transferHandler property.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

So, these are the constructor that can be used there while you are having the program and there are so, many methods using these methods the information about the panel can be extracted which can be utilized for controlling your program.

(Refer Slide Time: 17:33)

```
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
public class JFrameExample {
    public static void main(String s[]) {
        JFrame frame = new JFrame("JFrame Example");
        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());
        JLabel label = new JLabel("JFrame By Example");
        JButton button = new JButton();
        button.setText("Button");
        panel.add(label);
        panel.add(button);
        frame.add(panel);
        frame.setSize(200, 300);
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

And this is a simple example as we see it is a frame and this frame includes on simple button and then it will basically use it. And then this pa this program is as we see here a level is also used in this frame. So, level, button, button is level as button and then the frame is created as a. So, frame it is there title is JFrame by example.

So, this is the one thing that we have used in simple container, but the same thing again if you can repeat it, here just only changed it what change you should do it? For example, you want to include these things into an applet; that means, we want to add frame in their applet, only simple change you should do public class JFrame ex extends applet and then main method should not be there. Instead of main method you can write some unique method here you can and then code include all these codes that is all it will basically create a frame and then frame include some other elements like.

So, it is a simple that you have already experienced while we are discussing about AWT, how a component can be added into an applet how a component can be added into frame like.

(Refer Slide Time: 18:51)

Calculator with Swing : JCalculator

Methods used :

- add(Component c) : adds component to container.
- addActionListener(ActionListener d) : add ActionListener for specified component
- setBackground(Color c) : sets the background color of the specified container
- setSize(int a, int b) : sets the size of container to specified dimensions.
- setText(String s) : sets the text of the label to s.
- getText() : returns the text of the label.

The slide also features a screenshot of a Java Swing calculator application with a blue background and a grid of buttons. A small inset video of the presenter, Debasis Samanta, is visible in the bottom right corner. The footer includes logos for IIT Kharagpur and NPTEL Online Certification Courses.

So, this is the idea about frame and then calculator let us see one idea about. We have already given one illustration about how a calculator will look like if we use AWT. Now that time also we have explained we have mentioned that the calculator is not pretty look good, and lot of other styling that you have to do from the programmer end only so, that the calculator can look very good and more elegant is there.

However java swing has lot of simplicity; so that a calculator can be easily drawn according to the nice view of each without much headache about the programmer here. And as you see here this is a very good looking calculator and this is a calculator that we can develop using swing components easily. So, I will give you the step by step whatever the method that we can follow to develop such a calculator.

So, what are the method that is required and then designing a calculator is basically is a first time practice actually. If you can design a calculator successfully, then you can earn lot of confident about how the swing can be utilized to develop any application for your requirement that we can suit to your requirement.

Now, first the method that we should mention is that. So, we have to add with the component which are the component that needs to be used that say suppose we have to create this as a frame or is a panel, then you have to add thus component and then add other components those will be included in this one. So, here actually as we see wa; so all buttons are there. So, we can see 16 buttons as well as one equals button total

altogether 17 buttons are there and in addition to 17 buttons as we see 1 text field area is there. So, as we see these are the 17 buttons right and this is basically the frame area and in this frame area there is another text field also there.

So, the enter is the frame and in this frame there is one text field area and these are the button and this 1 there. And buttons are now included in this component; obviously, we can follow certain layout manager may be say grid layout manger. So, we can create a grid layout like this one then accordingly we can decide it. And if you use the flow layout manger also we can do that whatever the way you can do no issues. So, you can adjust it.

So, these are the 17 buttons you can customize to place it say suppose equals button you can say this one. So, right aligned or it is there, we have used the center aligned for this 1 like. So, the anyways so, these are the buttons and then text field and then the background these are the few things that you have to consider. So, for the this calculator design is concerned. You can change the color as a instead of blue may be say silent audio low whatever it is there anyway.

So, first we have to add the components and then all the components that will include into the container, we have to define then and then adjust. And then finally, this will give a look like which is basically just a structure, but we have to add the event now how the event can be added? We have already discussed about we have to register the action listener which action listener that you want to use here.

Here user is the mouse being the action listener can be added there, add action listener or specify event to be controlled. And then background color whatever it is there you can choose it size of the that calculator, also you can decide using set size function and then setText; setText is basically set the text to the level to s; that means, whatever the levels are there you can set the text like and then getText is basically returned to the text that we have leveled it here. So, here basically 1 2 3 all these things are text that we have set to each ba each button like and then it will basically return the values. So, whether value will be here in the text field or the value will be here all those things can be here.

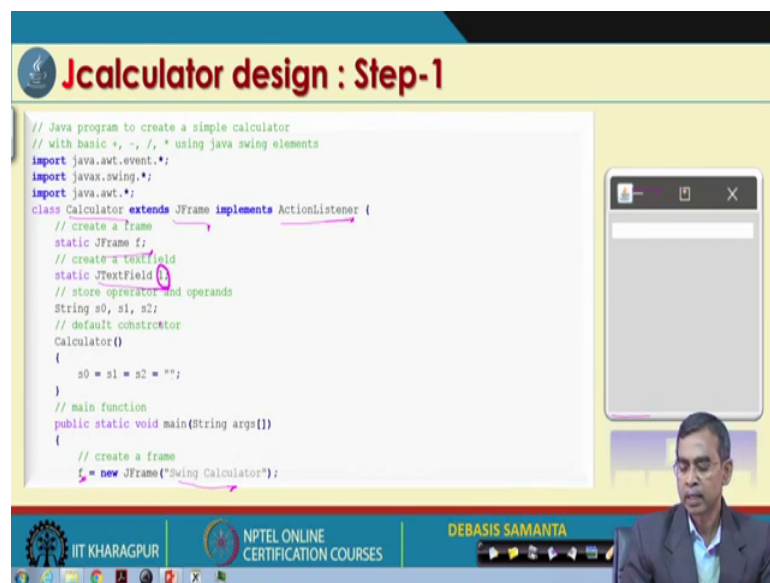
Now, in addition to this as we see these are the few other it is not the text actually, they are basically for some operation and those operation means if we click it basically ca action is that whatever the results it is there and then what are the subsequent things will

be entered into this it will appear and then the addition of the two apparent will takes place. So, this is the idea actually it is there.

Now, So, the idea is that we have to first design the layout and then place all the components in it and then finally, define the event handling mechanism. So, these are the steps major steps are involved here.

Now, let us see the step wise everything one by one for each discussion here.

(Refer Slide Time: 23:45)



```
// Java program to create a simple calculator
// with basic +, -, /, * using java swing elements
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;

class Calculator extends JFrame implements ActionListener {
    // create a frame
    static JFrame f;
    // create a textfield
    static JTextField t;
    // store operator and operands
    String s0, s1, s2;
    // default constructor
    Calculator()
    {
        s0 = s1 = s2 = "";
    }
    // main function
    public static void main(String args[])
    {
        // create a frame
        f = new JFrame("Swing Calculator");
```

Now, this is the first step in this step what we do is that, as we see here let us go slowly so, that you can understand about it. So, we are creating a calculator extends JFrame and implement ActionListener because we want to include event there.

Now, So, first we create a JFrame and this is a filter frame and then text field that is given as a 1 and a 0 s 1 and s 2 are the 3 strings that we have considered for the 2 apparent and one result actually. So, calculator is the ba constructor initially they are basically null stream and so, initially calculator is basically has the null value like and then we have to go to the main frame main method. The main method create a frame f using swing calculator. So, it will be basically title here or it will be displayed here. So, in calculator look like

So, with these things it will basically create like this one.

(Refer Slide Time: 24:47)

The slide displays the following Java code:

```
try {
    // set look and feel
    UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
}
catch (Exception e) {
    System.err.println(e.getMessage());
}
// create a object of class
Calculator c = new Calculator();
// create a textfield
JTextField t = new JTextField(10);
// set the textfield to non editable
t.setEditable(false);
// create numeric buttons and some operators
JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, ba, bs, bd, be, bf, bq, beq;
// create numeric buttons
b0 = new JButton("0");
b1 = new JButton("1");
b2 = new JButton("2");
b3 = new JButton("3");
b4 = new JButton("4");
b5 = new JButton("5");
```

The slide also features a small window icon on the right and a video feed of the presenter, Debasis Samanta, at the bottom right. The footer includes logos for IIT Kharagpur and NPTEL Online Certification Courses.

Now if we go put the next step; our step 2 and then in the step 2 as we see we just use one UIManager it is defined in the a java swing itself SetLookAndFeel. UIManager dot getSystemLookAndFeelClassName. So, is basically automatically defined whichever the look that you can have. So, it is basically that this is a default look that we have used it here and if suppose it is not a able to decide this UIManager, then it will just simply throw an error like this one.

Anyway we can this one and get system look and feel class name is basically give you what are the UIManager that you are including. It is just same as the layout manger concept it is a here in the java swing expression it is called the user interface manger UIManager like.

Anyway then we create a calculator object c is a type we have already designed the constructor there with initial value s 0 equals to s 1 equals to s 2 equals to non-string like and then we create we create a text field they then text field is there and set editable; that means, text field can be editable we can try something here also this is why it is make editable and this is a false means no editable is possible. So, we cannot type anything here

And then we create so, many buttons b 0 to b nine for the numeric button and b 8 to b m for addition subtraction and division multiplication for equal and b equal b equal lay b equal other 2 also we have created it anyway. And then we add all these buttons or we

create a button v 0 with some level as it is shown here. So, we are just we are continuing the step 2 again, because there are few more items to be added there and so.

(Refer Slide Time: 26:39)

```
b6 = new JButton("6");
b7 = new JButton("7");
b8 = new JButton("8");
b9 = new JButton("9");
// equals button
beq1 = new JButton("=");
// create operator buttons
ba = new JButton("+");
bs = new JButton("-");
bd = new JButton("/");
bm = new JButton("*");
beq = new JButton("%");
// create . button
be = new JButton(".");
// create a panel
JPanel p = new JPanel();
// add action listeners
bm.addActionListener(c);
bd.addActionListener(c);
bs.addActionListener(c);
ba.addActionListener(c);
b9.addActionListener(c);
```

So, we are adding all these aa items all buttons are there na 0 to 9 buttons all these basically some addition subtraction multiplication all these buttons are there with label all these things are there and 10 b another button is created with dot. So, point is there and b equal another button also will be added.

Now, JPanel p equals to new panel we create 1 panel that is a panel is basically we want to include this is basically rather we can say instead of frame it is a panel actually. So, in this panel we want to add all these things there. Now we are defining the action listener method here. So, bn bd bs ba for this class component actually here in this case calculate the component. So, we are adding some methods that that if we press bn then what of action that it will produce. So, these are the things it is declared here and then. So, these are the action listener for all components those we have included here we have added it there.

(Refer Slide Time: 27:41)

```
b8.addActionListener(c);
b7.addActionListener(c);
b6.addActionListener(c);
b5.addActionListener(c);
b4.addActionListener(c);
b3.addActionListener(c);
b2.addActionListener(c);
b1.addActionListener(c);
b0.addActionListener(c);
be.addActionListener(c);
beq.addActionListener(c);
beql.addActionListener(c);
// add elements to panel
p.add(l);
p.add(ba);
p.add(b1);
p.add(b2);
p.add(b3);
p.add(b4);
p.add(b5);
p.add(b6);
```

And then finally, we add all the buttons and labels and text field into the panel. So, this is the statement for that. So, this completes the addition of all the elements into the button side by side and then this one; what is my suggestion is that, you just do 1 step and then run it check it that how it is going on. So, usually if you do everything together and then you may not find that how it is working. So, first create the button and then you complete it and then run it and see the output then add buttons then run it see how it is coming add text field run it see and these things. So, one by one if you do it you will be able to have the good feeling that how it is working actually for you.

(Refer Slide Time: 28:33)

```
p.add(tm);
p.add(b7);
p.add(b8);
p.add(b9);
p.add(bd);
p.add(bo);
p.add(b0);
p.add(beg);
p.add(beq);
// set Background of panel
p.setBackground(Color.blue);
// add panel to fram
f.add(p);
f.setSize(200, 220);
f.show();
}
public void actionPerformed(ActionEvent e)
{
String s = e.getActionCommand();
// if the value is a number
if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {
// if operand is present then add to second no
```

And then next step is basically once the all components are added into the button and we have to define the action performed for each what is called the component. So, here is a routine for action performed that you have to write it. Now here if s dot character 0; that means, the first character is 0 or it is less than 9 or it is this one. So, this basically go on appending here. So, if you type something here in this one. So, it will go on appending here. So, 45.6. So, it will 45.6 will be display here like that.

So, this is basically the first thing that we are doing and then. So, this is basically the in continuation.

(Refer Slide Time: 29:19)

The slide is titled "JCalculator design : Step-6". It features a code editor on the left with Java code for a calculator's actionPerformed method. The code includes logic for adding, subtracting, multiplying, dividing, and clearing the display. A small calculator UI is shown on the right, with pink arrows pointing from the code to the corresponding buttons. The speaker, Debasis Samanta, is visible in the bottom right corner.

```
if (s1.equals("")) {
    s2 = s2 + s;
} else {
    s0 = s0 + s;
    // set the value of text
    l.setText(s0 + s1 + s2);
}
else if (s.charAt(0) == 'C') {
    // clear the one letter
    s0 = s1 = s2 = "";
    // set the value of text
    l.setText(s0 + s1 + s2);
}
else if (s.charAt(0) == '+') {
    double te;
    // store the value in lst
    if (s1.equals("")) {
        te = (Double.parseDouble(s0) + Double.parseDouble(s2));
    } else if (s1.equals("-")) {
        te = (Double.parseDouble(s0) - Double.parseDouble(s2));
    } else if (s1.equals("/")) {
        te = (Double.parseDouble(s0) / Double.parseDouble(s2));
    }
}
```

Next is here we are clearing if the different things are app for example, if this equals right this thing is there then is basically sum and then if this is c the clear. So, c button is there if we then it is basically what will happen and then. So, these are the different actions if something is happened for plus or minus or division then is basically what it will do all these things. So, whenever you face it with basically read one string another read string. So, that those string and attach there and then result will be obtained. So, this is the procedure by which the different actions can be listens and for this actions the consequence can be added there and operation will be there.

(Refer Slide Time: 30:11)

The slide is titled "JCalculator design : Step-7". It features a code editor on the left with the following Java code:

```
else
    te = (Double.parseDouble(n0) * Double.parseDouble(n2));
    // convert it to string
    s0 = Double.toString(te);
    // place the operator
    s1 = s;
    // make the operand blank
    s2 = "";
}
// set the value of text
l.setText(s0 + s1 + s2);
}
```

On the right, there is a screenshot of a Java Swing calculator window. The display shows "12*3=15.0". The calculator has a numeric keypad (0-9), a decimal point, a clear button (C), and an equals button (=). Below the code and calculator, a small video inset shows the presenter, Debasis Samanta, speaking.

The footer of the slide includes the IIT Khargapur logo, the NPTEL Online Certification Courses logo, and the name "DEBASIS SAMANTA" with "IIT KHARGAPUR" below it.

And finally, as we see here finally, the result will be stored there and the result will be like this.

(Refer Slide Time: 30:17)

The slide is titled "Questions to think...". It contains two bullet points:

- Can we mix both Swing and AWT elements in the same program?
- What is the class counts in AWT and Swing?

The footer of the slide includes the IIT Khargapur logo, the NPTEL Online Certification Courses logo, and the name "DEBASIS SAMANTA" with "IIT KHARGAPUR" below it.

So, set text is a level that means text field area it can shown it here. Now so, idea how a calculator can be designed I have given an idea about it. So, two things are there first you have to plan the entire layout and then according to the layout we have to decide that what are the different components are to be included and then you include those

components and then you have to add the action handling mechanism that is all. So, this way you can we will be able to do it.

So, this program as I have given for your practice I should advise you to run this program, and then you will see how it is working and then part by part program execution is better suggested so, that you can understand this concept clearly.

Anyway so, we have learned about AWT and Swing and these are the things very much essential for window programming, other than the simple pro programming that we have learnt regarding mutli threading input output handling and then in capsulation all those things are there, but this will add more color to your programming arena actually. And here obviously, you can have the answer that whether both sw string and AWT can be punched together or not; obviously, it is possible and then as you see if you compare AWT and string then they are comparable in terms of many other parameters, but 1 parameters is that swing gives you much more elements than the AWT.

It is much much more actually. Now I just slip it as an exercise that how many classes that you can use. So, for the AWT is concerned and how many swing it is there. Now you can count it and you can just learn it that swing is mo much more elaborative more voluminous. However, easier than the AWT to use it and it gives you more what is called the smart looking program development.

So, thank you for your attending this module.