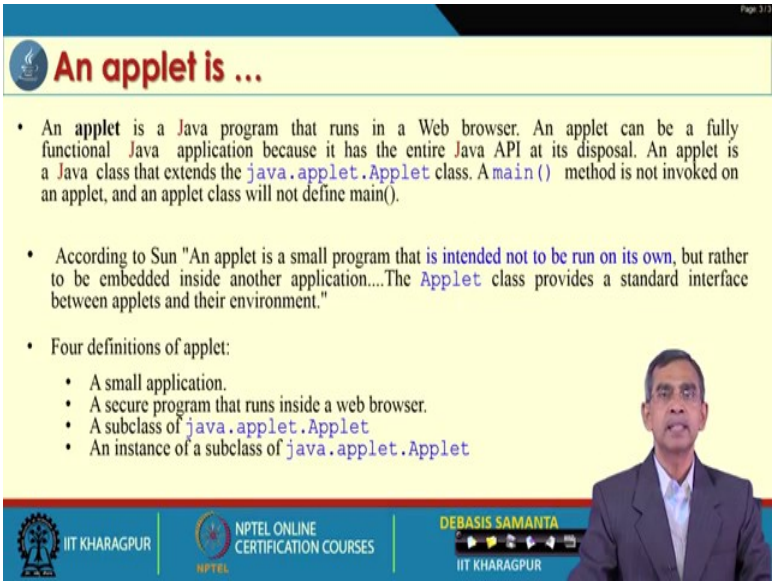
Programming in Java
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 39
A W T Programming – I

In Java, all programming's can be broadly classified into three categories, the core programming, applet programming and AWT programming. Now, AWT programming and applet programming they are alternatively called the graphics oriented programming or more precisely we can say window based programming. So, here basically we have to develop windows and through windows the programming aspects can be carried out.

(Refer Slide Time: 01:01)



Now, today we will discuss about AWT programming. AWT programming actually we have to see exactly both applet and AWT programming just now I have mentioned is a graphics oriented programming which is the main difference from the core programming.

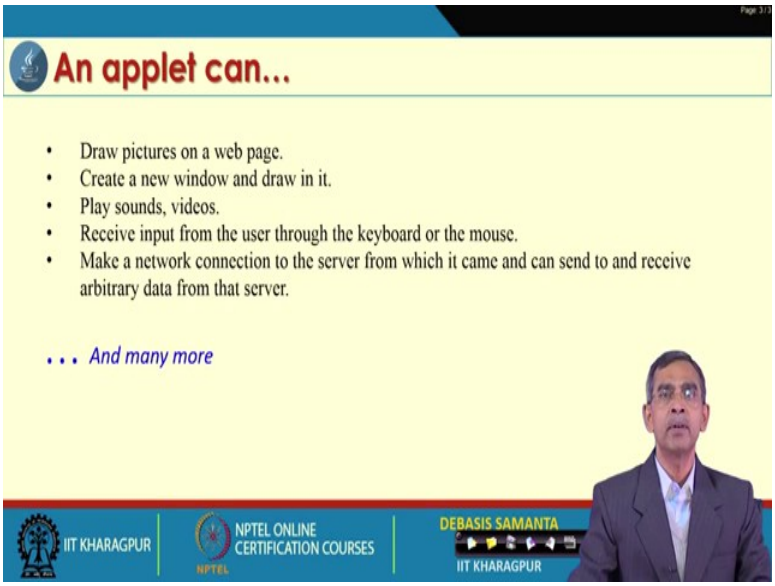
Now, what is the difference between an applet programming and then AWT programming? So, as we know an applet is basically a java program which can be executed through a browser. On the other hand, if we see the core programming we do not run it using a web browser, rather we can run independently from the console itself. Now, AWT in that sense similar to core programming that here we do not need any web

browser to run a program which is developed using AWT. Now, so this is the one difference, there are many more difference also.

Now, so, for the applet is concerned it is basically a small program as we have seen and usually this small program is mean for internet-oriented programming or more precisely is basically web programming we can say, where the browsers can browser the net and can access their web page and this web page will usually include applet. So, if you want to design web page then definitely you can think for applet. However, AWT and core is not in that sense web programming actually.

And as you know that applet, whenever we have to design an applet we usually follow the package applet where applet class is defined. So, that is why all applet programs are called subclass of java.applet.Applet. Applet class basically is an abstract class which you have to inherit for our applet program. It is also alternatively called an instance of a sub class of java.applet.Applet. So, this is basically the different way an applet can be defined; however, the core and AWT is not like the applet is as it is mentioned.

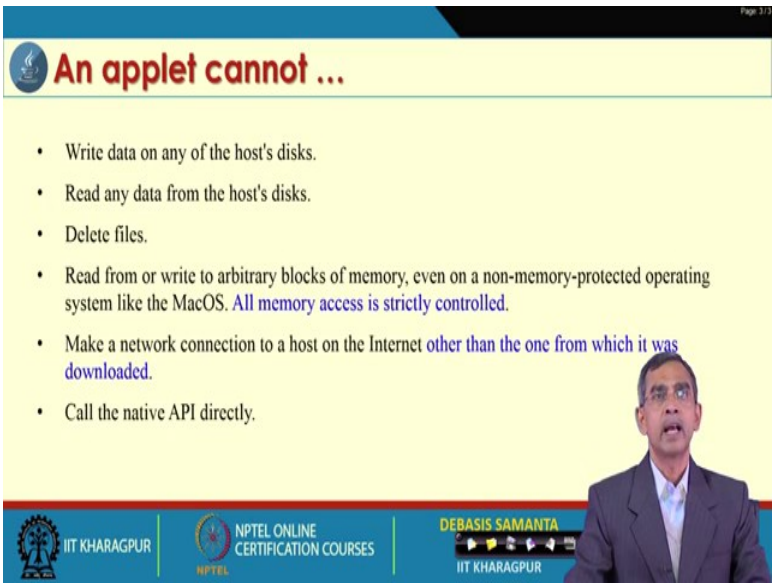
(Refer Slide Time: 03:31)



Now, what an applet can do for us? As you have checked it using applet it user can draw any pictures on the interface, I mean in graphic input screen, display screen rather we can say. It can create a new window, and this window can be enable to draw any graphics. It can play sounds audio video it also can receives input from the user through the keyboard or the mouse, but not the input the conventional way that a core program can

allow in reading an integer or float or string like this. It only the sense the keyboard or mouse whether there is a clicked or keyboard is pressed, mouse is dragged like this. And obviously, applet is for internet programming in the sense that in the distribute environment applet can communicate to another applet to serve the some communication need. There are many more applications of the applet is also there.

(Refer Slide Time: 04:42)



On the other hand, an applet which has many limitations as well as for example, an applet cannot write any data into any hard disk even in the local machine hard disk also. It cannot read any data from the host machine; that means, from which the applet is running. It cannot delete any files, it cannot read from or write into any arbitrary memory in any non-memory protected operating system like MacOS or like these.

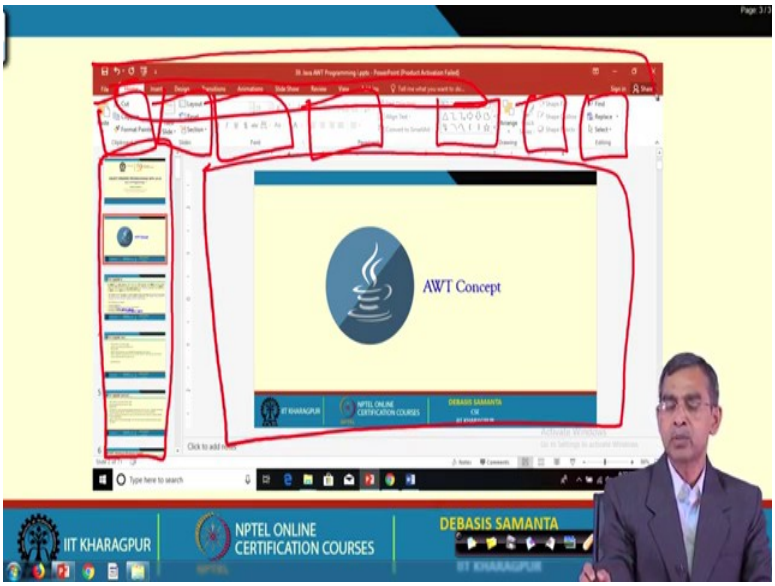
So, basically applet is heavily restricted, so far memory access is concerned. And it basically enables a network connection to I mean host the internet or any other from which it was download; that means, it cannot establish connection of its own. And it also cannot call the native API directly. If you develop your own API it cannot use this things, whatever the methods those are declared there in the applet class you can use it and if can extends then through extends only, it can access all those methods otherwise native API cannot be utilized in this applet whereas, core programming or AWT programming can do.

As we see core programming has many way resembles with AWT programming on the and the difference from the core programming and AWT programming is that, core is totally controlled console based; that means, giving input or output whatever it is there from the keyboard or whatever it is there is no graphical user interface component, there is no window component. Whereas, AWT is totally graphical user interface based or windows based.

Now, let us have the more elaborately the different concepts of the AWT that we are having. So, AWT as we are telling AWT, AWT it means that it is a full form of a big I mean 3 words actually Abstract Windowing Tool kit, as the name implies it is basically a tool kit. This tool kit a java programmer can utilize to develop their own applications software and is an Abstract Windowing Tool kit, it is called the abstract this is because it gives a line gives an way how the different method can be utilized. However, time to time user has the flexibility to over write that method and can used it is there.

So, in usually all AWT programming we can say little bit light weight programming process whereas, the core programming is heavy weight if you want to do something you can develop of your own other than those are the facilities available in the APIs. In general, so for the graphics oriented is concern core programming is heavily heavy weight whereas, using AWT it is light weight. Light weight means all methods almost are there you just play a plug and play that is all; that means, you can use and then solve your programming development.

(Refer Slide Time: 08:09)



So, this is the AWT. And as it is a tool kit this means that it has lot of packages and all this packages is defined in AWT package. So, java.awt., AWT is the package where all AWT tool kit classes are there. Now, using AWT what can be done, I have given an idea about it. This is basically a screen shot of a application which is running while I was preparing the PPT slide I took this screen shot actually. Now, I made this PPT slides using PowerPoint, PowerPoint is an application software which helps a programmer to develop their slides presentation slides.

Now, here if you see there are many components are there, many, it is basically a window also. Now, in this window as we see there are many, so this is basically a one view area where the content the user is preparing can be displayed here. Now, apart from seeing this is the another what is called the area where it will display some other elements also.

Now, here are many other components for example, here this is the there, right this is all these are basically is a small small; if the entire region window then all these are the small small window actually it is there. And every window has its own item some items are called the least of items, some items are called the menus, some items are called the choices, like this one.

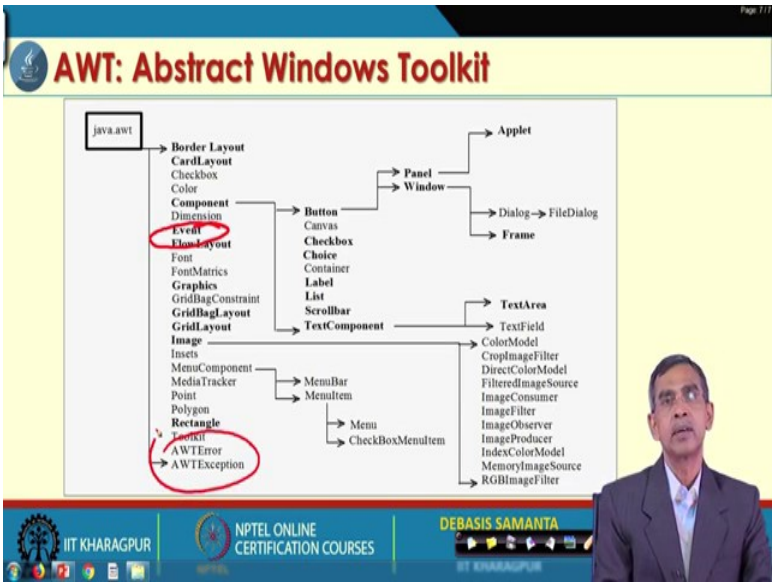
And there is also some other these are called the bottom like. So, if you click it using these things corresponding some pop menu will appear and it will give other options

there by which user can select and can do many things are there. And other things are very similar to the applet look like. So, these are the basically close, these are minimization, maximization and also and similarly the status bar is also there.

Now, this basically is a product of an AWT programming and this is the product that is used in Microsoft PowerPoint. Similarly, there are many other application softwares you usually used nowadays those are basically graphics oriented or we can say more precisely GUI oriented, Graphical User Interface oriented and or more precisely based on window programming. So, AWT is basically a window programming which is graphically user interface oriented or graphics oriented. So, graphically user interface popularly called GUI; so it is G U I short form is called as GUIs.

Now, if you want to have back GUI in your programs, in your software then AWT is the one best solution that I can recommend. So, this is the idea about AWT. And now the question is that if you want to develop a software we can that is application software like PowerPoint or Microsoft or Paint brush or something like then how we can do it.

(Refer Slide Time: 11:18)



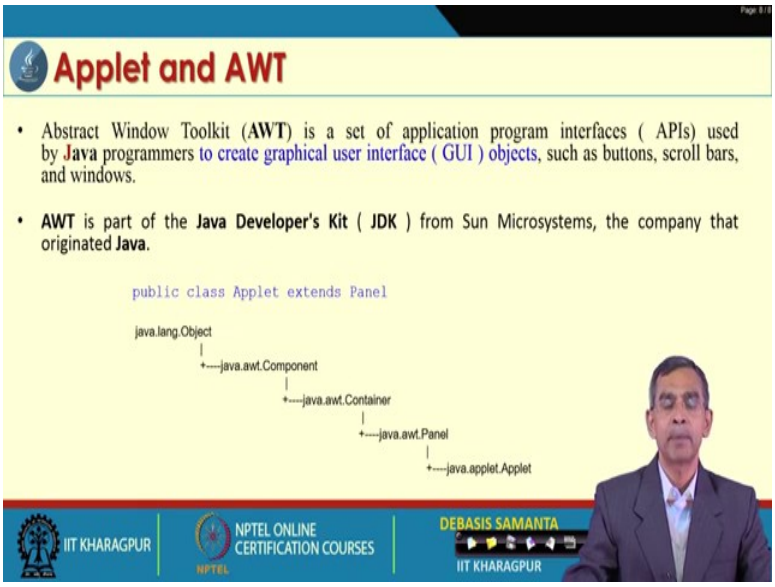
Obviously, we can do it using AWT package. Now, AWT package in fact, is a very vast package. It includes a lot of classes and the sub classes there. Now, all the entire packages can be broadly segmented into few categories the first thing is that there is a graphics oriented components are there. So, it is a graphics a related packages are there. So, is a for example, graphics and then this is the point polygon all these things are

basically the graphics-oriented things are there. And then there is another is called the component-oriented concept is there. So, component-oriented concept it is like these are the component-oriented concept are there in this, these are the basically component related facilities we can say classes, sub classes all these things are there. And then image related some facilities are there or classes are there. So, image related classes are like these are the image oriented things. So, if you want to process image and everything then AWT will help with so many classes and others.

And then and layout management, so this is called the layout manager in general; so like these are the different layout managers are there. So, this is the layout manager related packages are there. And in addition to this event is a very important in any windows programming, regarding event there is a tool kits are there, that is the event handling basically by which we can do it and other than all these things every packages needs to be very much provide the programmer to develop the robust software. So, error and exception handling somethings are also there in this package.

So, as you see there are so many things are there graphics-oriented, image-oriented, component-oriented, event-oriented and then others exception handling oriented facilities is provided by AWT. Now, in our discussion we will try to learn all these concepts, ok.

(Refer Slide Time: 13:43)

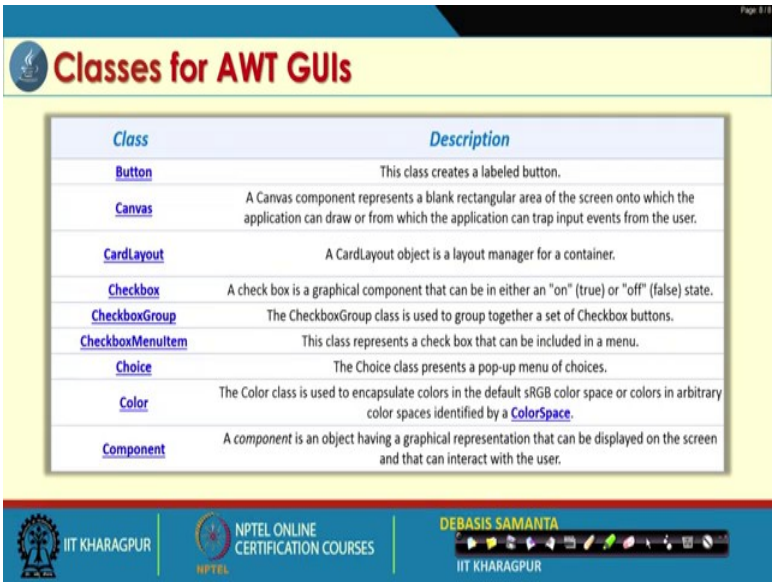


So, applet versus AWT as you have discussed it is, we have our understanding it is I hope it is clear. Now, AWT how this is whether it is a free or not, it is absolutely it is a

part of the APIs whenever you install JDK as a part of this JDK installation the AWT package will be automatically installed.

So, it is the part is there. And then you can use this AWT to inherit many other classes, some other there other classes to solve your problem or the access the different methods those are there in the package component or classes in that package. Now, AWT is therefore, developing graphical user interfaces are there. And mainly the component is responsible to build GUI or GUIs. Now, what are the component? Those are very important to learn at this stage or whether I can say at the beginning stage of your programming understand learning I just want to include it.

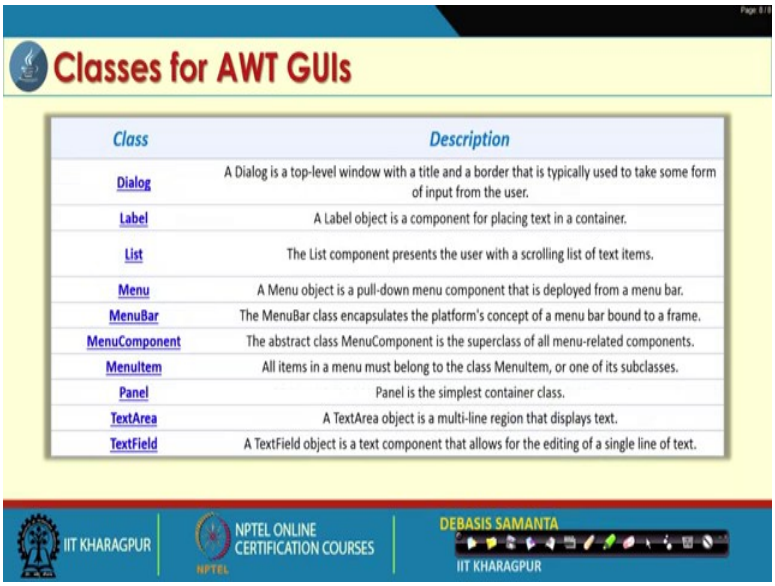
(Refer Slide Time: 14:41)



Class	Description
Button	This class creates a labeled button.
Canvas	A Canvas component represents a blank rectangular area of the screen onto which the application can draw or from which the application can trap input events from the user.
CardLayout	A CardLayout object is a layout manager for a container.
Checkbox	A check box is a graphical component that can be in either an "on" (true) or "off" (false) state.
CheckboxGroup	The CheckboxGroup class is used to group together a set of Checkbox buttons.
CheckboxMenuItem	This class represents a check box that can be included in a menu.
Choice	The Choice class presents a pop-up menu of choices.
Color	The Color class is used to encapsulate colors in the default sRGB color space or colors in arbitrary color spaces identified by a ColorSpace.
Component	A *component* is an object having a graphical representation that can be displayed on the screen and that can interact with the user.

There are many classes it is there or sub classes it is there, I have included those are related to the component here like say Button, Canvas, CardLayout, Checkbox, etcetera.

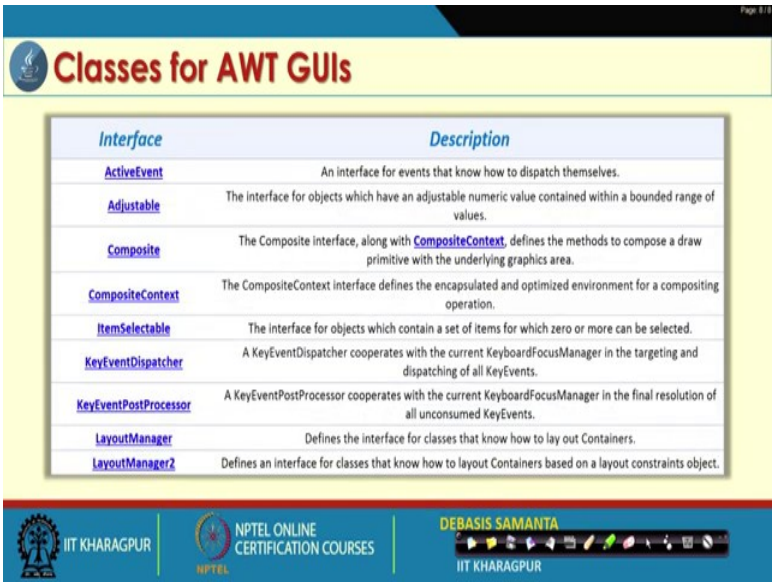
(Refer Slide Time: 14:52)



Class	Description
Dialog	A Dialog is a top-level window with a title and a border that is typically used to take some form of input from the user.
Label	A Label object is a component for placing text in a container.
List	The List component presents the user with a scrolling list of text items.
Menu	A Menu object is a pull-down menu component that is deployed from a menu bar.
MenuBar	The MenuBar class encapsulates the platform's concept of a menu bar bound to a frame.
MenuComponent	The abstract class MenuComponent is the superclass of all menu-related components.
MenuItem	All items in a menu must belong to the class MenuItem, or one of its subclasses.
Panel	Panel is the simplest container class.
TextArea	A TextArea object is a multi-line region that displays text.
TextField	A TextField object is a text component that allows for the editing of a single line of text.

And like there are many classes also, these are few more classes.

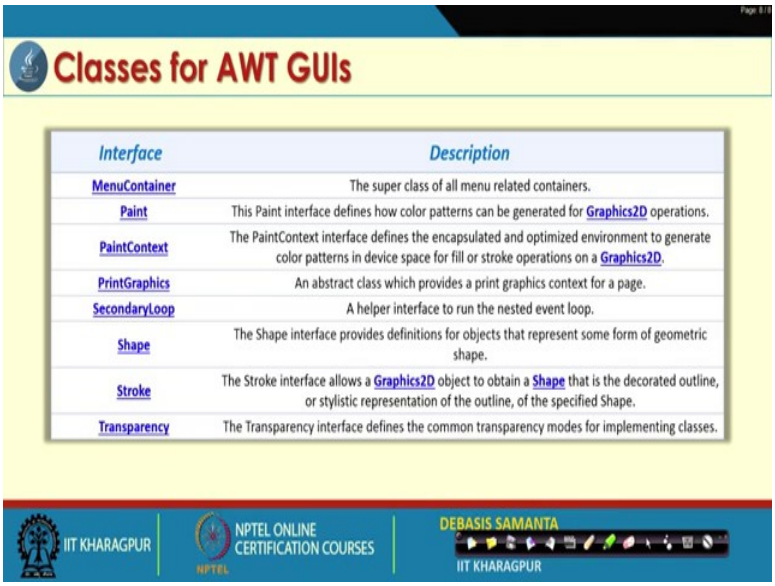
(Refer Slide Time: 14:56)



Interface	Description
ActiveEvent	An interface for events that know how to dispatch themselves.
Adjustable	The interface for objects which have an adjustable numeric value contained within a bounded range of values.
Composite	The Composite interface, along with CompositeContext, defines the methods to compose a draw primitive with the underlying graphics area.
CompositeContext	The CompositeContext interface defines the encapsulated and optimized environment for a compositing operation.
ItemSelectable	The interface for objects which contain a set of items for which zero or more can be selected.
KeyEventDispatcher	A KeyEventDispatcher cooperates with the current KeyboardFocusManager in the targeting and dispatching of all KeyEvents.
KeyEventPostProcessor	A KeyEventPostProcessor cooperates with the current KeyboardFocusManager in the final resolution of all unconsumed KeyEvents.
LayoutManager	Defines the interface for classes that know how to lay out Containers.
LayoutManager2	Defines an interface for classes that know how to layout Containers based on a layout constraints object.

And there are many interfaces also, those are things it is included here.

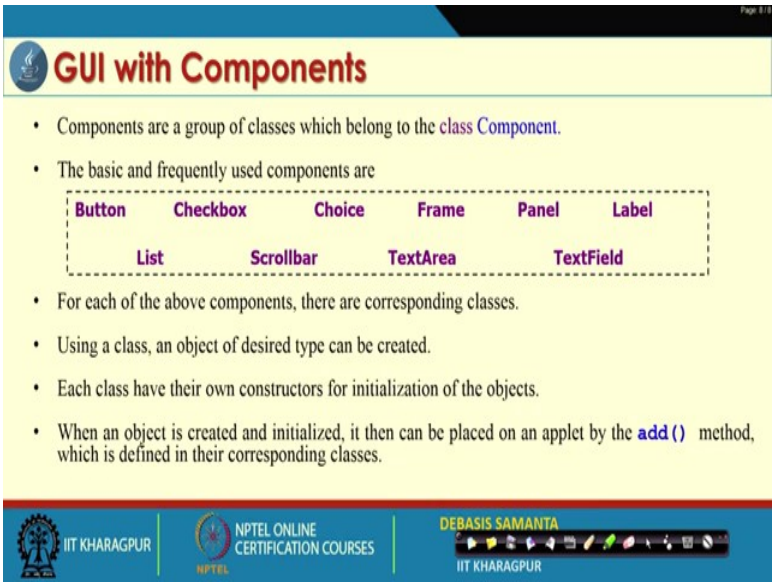
(Refer Slide Time: 15:59)



Now, so there are many class and interfaces are there. If you want to use AWT obviously, you should need some understanding about all these classes and interfaces and then the methods in all these classes and everything. As it is very exhaustive and elaborate, so it is also not possible feasible to discuss each and every classes one by one discussing their methods, but in my slides I will include all the classes, and the interfaces the methods in them, so that you can have the ready reference from here. However, if you want to learn it then definitely you will have to consult the many other sources which I have already given in the my first lectures of introduction.

Now, I will just discuss GUI development using the component that is there in AWT.

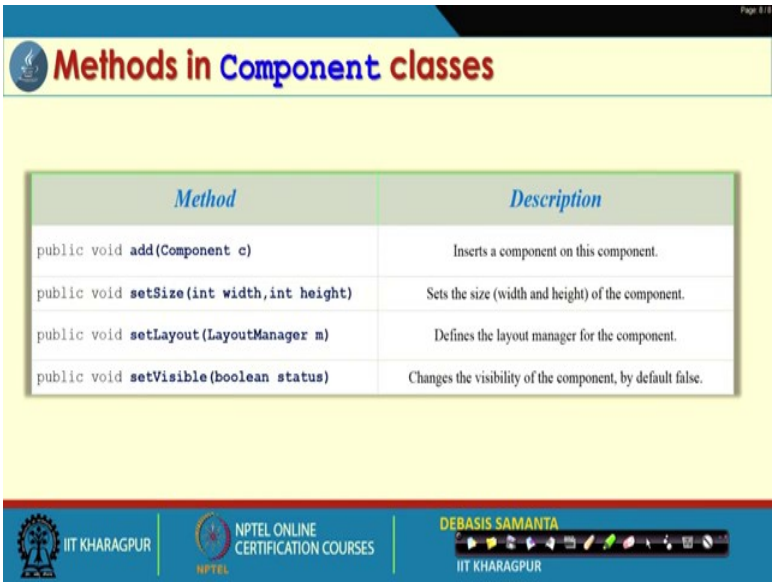
(Refer Slide Time: 15:51)



Component is basically itself is a class which is there in AWT and it has many other subclasses like button, checkbox, choice, frame, panel, label, list, scrollbar, TextArea, TextField which I have been included here. By name itself it means that what exactly this component elements is can do for us. Probably we have familiar with the button. So, if you want to create a button as a graphical user interface in your application then definitely you have to use this class.

Like, like button there are many other graphical user, so these are basically GUI elements we can say. They can be created for your pro software, and they can be added in your software, they can be used, they can be developed and they can be designed, they can be modeled whatever it is it basically gives all facilities for the programmer.

(Refer Slide Time: 16:43)



Now, I will discuss about the component classes. As we know, so first you have to create a component and to create a component, a component is basically a window I can say. So, if you want to add some other component into that window then there are many methods are there which is which are defined in component class itself. The methods like here add; as we see add means if you want to add button.

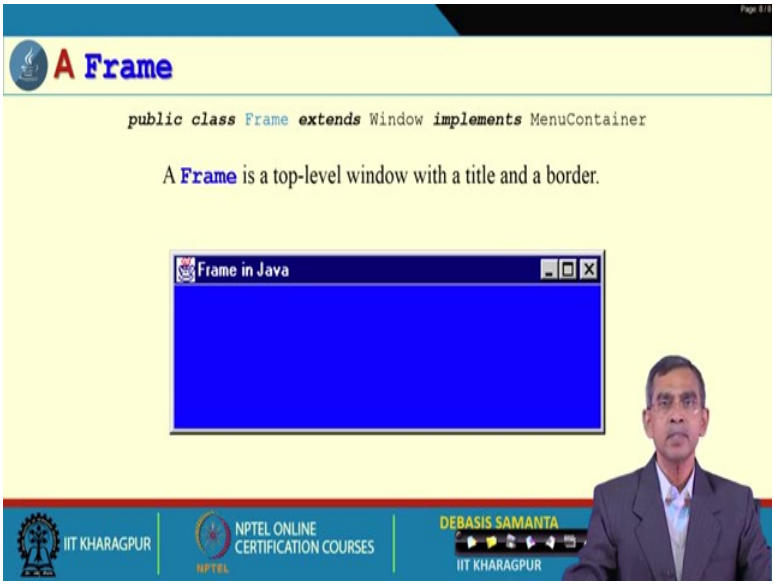
So, definitely add method should be there add, then there is a component we in which component you want to add it. And then set size, you can just resize the component and set layout a component can allow you to have many layouts. We will discuss about what are the different layouts that is possible for a window. And then, set visible it is basically you can sometimes a particular component or a window make it visible, make it invisible all these things are there, so methods.

These are the 4 methods which are very important methods. And all these methods are defined in component class, we have to just call this method in your program and then that method has their own facilities; that means, it is a light weight process actually.

Now, first let us discuss about how we can create a frame using or AWT tool kit and a frame is basically another container we can say. A container, basically an component basically same thing an applet is also container, container means it basically contains many graphical user elements GUIs like button, checkbox, scrollbox, whatever it is there; so an applet can include all those things. Now, we will see a frame is also similar

to this it can also include many components in it. So, frame is very important one items or important one component which is defined in AWT package.

(Refer Slide Time: 18:39)



Now, how actually a frame look like? So, here basically we have displayed one frame whose background color as you see with color blue and in the top is basically the status bar is a that that basically showing that what are the windows are there and as we know for any windows there is a minimization, maximization and cross symbol includes that if you want to close this window.

And there is a title also it is showing that is which application is PowerPoint or Microsoft or whatever the user wants to make it that this one. So, it basically is a container. And the frame also more precisely it is called. So, a frame has its own title and then some area, and in this area, we can include many other components graphical user elements in to it here. So, a frame is typically look like this.

(Refer Slide Time: 19:34)

Class Frame : Constructors

Constructor	Description
Frame ()	Constructs a new instance of Frame that is initially invisible.
Frame (GraphicsConfiguration gc)	Constructs a new, initially invisible Frame with the specified GraphicsConfiguration.
Frame (String title)	Constructs a new, initially invisible Frame object with the specified title.
Frame(String title, GraphicsConfiguration gc)	Constructs a new, initially invisible Frame object with the specified title and a GraphicsConfiguration.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

Now, let us see how this kind of frame can be designed. Now, a frame if you want to use it definitely you can call many constructor which is already defined in the class component class, class frame actually, a frame is define frame is a sub class of component class. Now, these are the 3 constructor as we have mentioned here. So, details you can just go through whatever the point it is written. I will not elaborate whatever the different construct it is there.

(Refer Slide Time: 20:02)

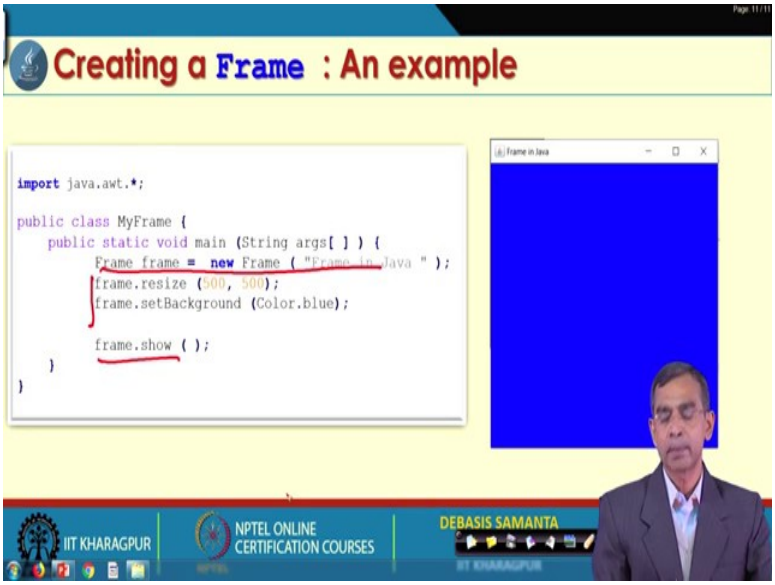
Class Frame : Methods

Modifier and Type	Method	Description
void	addNotify()	Makes this Frame displayable by connecting it to a native screen resource.
AccessibleContext	getAccessibleContext()	Gets the AccessibleContext associated with this Frame.
int	getExtendedState()	Gets the state of this frame.
static Frame[]	getFrames()	Returns an array of all Frames created by this application.
Image	getIconImage()	Returns the image to be displayed as the icon for this frame.
Rectangle	getMaximizedBounds()	Gets maximized bounds for this frame.
MenuBar	getMenuBar()	Gets the menu bar for this frame.
int	getState()	Gets the state of this frame (obsolete).
String	getTitle()	Gets the title of the frame.
boolean	isResizable()	
boolean	isUndecorated()	Indicates whether this frame is undecorated.
protected String	paramString()	Returns a string representing the state of this Frame.
void	remove(MenuComponent m)	Removes the specified menu bar from this frame.
void	removeNotify()	Makes this Frame undisplayable by removing its connection to its native screen resource.
void	setBackground(Color bgColor)	Sets the background color of this window.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

And apart from this constructor it also has many methods. Those are the methods which we have listed here.

(Refer Slide Time: 20:09)



And here is an example, we can check it how we can create a frame. Now, this program is interesting to watch it and I will just example this program actually.

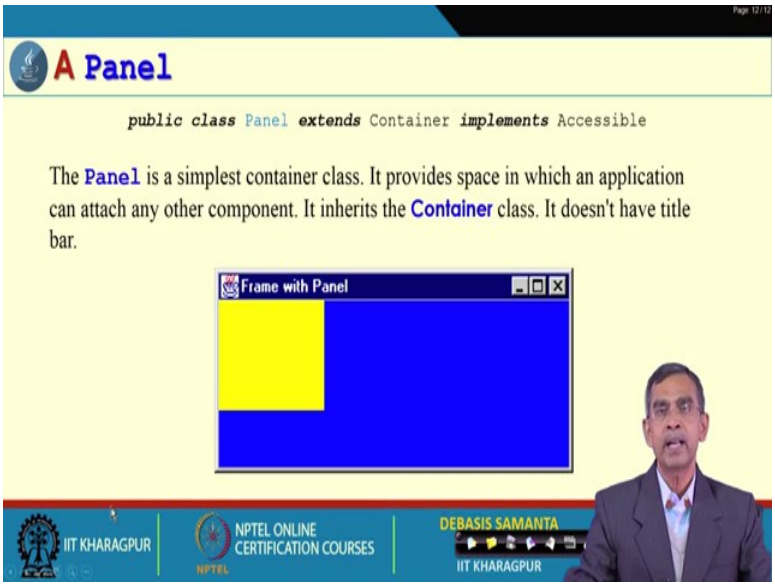
So, this program objective of this program is to how we can create a frame. Now, here we have created one core main program, basically we have defined one class called the MyFrame class and the main method we have defined is here and here the Frame is basically created of the class Frame. So, and then this is the title that will be displayed here like and frame dot resize( ) is basically what is the size of this frames. So, it is 500 by 500 size it is there and frame set back background color blue if we use it then it will make the background color like this one.

Now, here one thing is that this methods resize, set background, and these are constructor of course, the constructor is already defined in the frame class itself, but this resize set background and here is for example, show, show means the frame will be displayed is basically defined in the component class itself. As we have already used these class frame, so by virtue of this all these methods are accessible to this program itself. So, these are the, this is the way by which we can just create a frame. So, creating a frame is very simple. We have to use this and then define their size, their background if it is there,

if you do not do it the default background is a white will appear and then finally, we have to show the frame; that means, frame is we have to make the frame visible.

So, this is the simple way the program can be used to create a frame. Now, so frame is one container as we have discussed, there is another also panel.

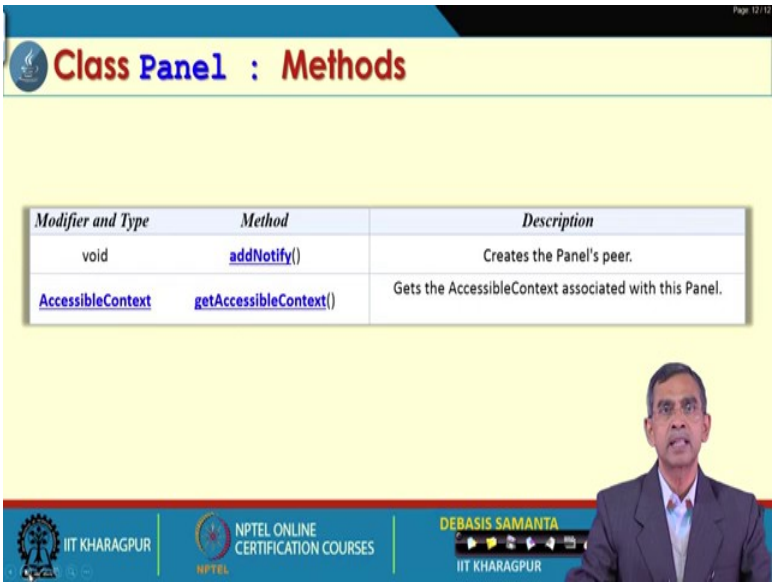
(Refer Slide Time: 22:03)



But the difference between frame and panel, basically frame and panel same the difference is that a frame as title whereas, a panel it does not have.

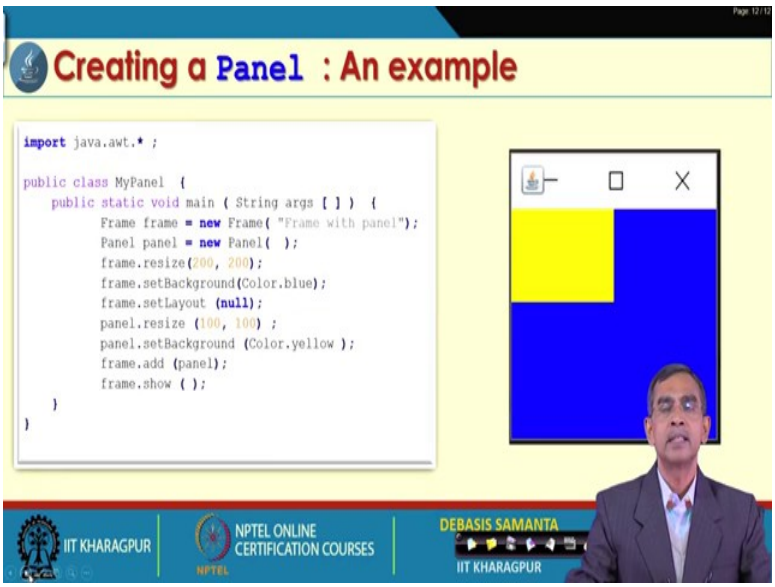
(Refer Slide Time: 22:14)

(Refer Slide Time: 22:16)



So, here is an example the constructor, these are the constructor, these are the methods that is defined.

(Refer Slide Time: 22:19)



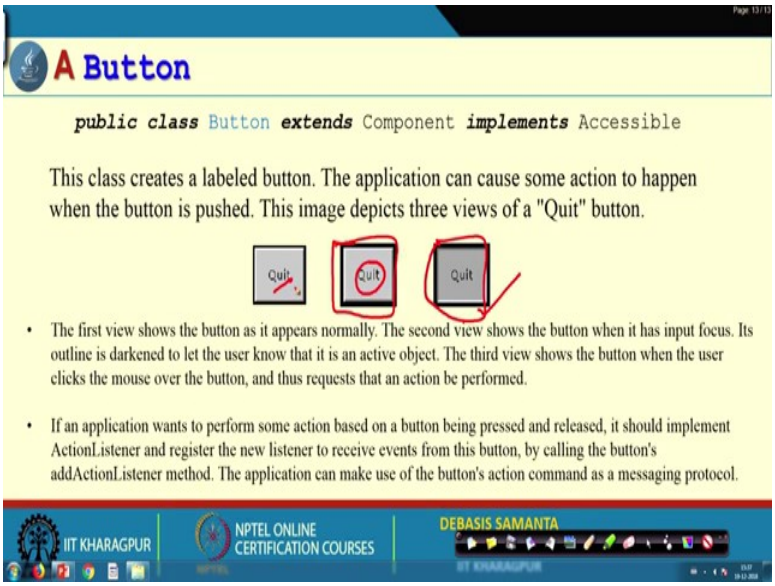
And this is one example by which we can create a panel and this panel can be included in a frame. Now, obviously, a panel can include a frame, a frame can include a panel and all these things can be plotted in a contained which is not a frame, not a panel whatever it is there that is a simply a container like.

Now, here this is the idea again how we can create a panel which basically a content in a frame. So, you can just watch out this code here, again the same this is the class program that is to create the panel actually and here you see we have created a frame and we have created then a panel. So, the default constructor it is there and this is the constructor that is there for the frame. And then this is for the frame resize setBackground layout and this is basically for the panel resize and setBackground is there. So, as you see all these things can be managed or controlled using this method.

And finally, frame dot add (panel) this is important; that means, we add the panel into this frame. So, this is the panel and this is the frame, we add this panel into this frame. So, by default it will be there, but again another option will be there this panel can be placed here or anywhere by deciding other parameters So, there is a overriding method of course, so overloading method other parameter can be used. So, that this panel can be placed anywhere you used and finally, once the frame is ready which includes panel it can be shown.

So, this is the idea about how a frame include a panel or more clearly how a panel can be created it can be. And likewise, previous example of creating frame independently a panel also can be created, but this does not have many used rather it should be is a part of frame or container or an applet like that, ok. So, we have learned about how the frame the panel can be there and all these container say may be applet or frame or a panel they can include some other components say for example, button.

(Refer Slide Time: 24:29)

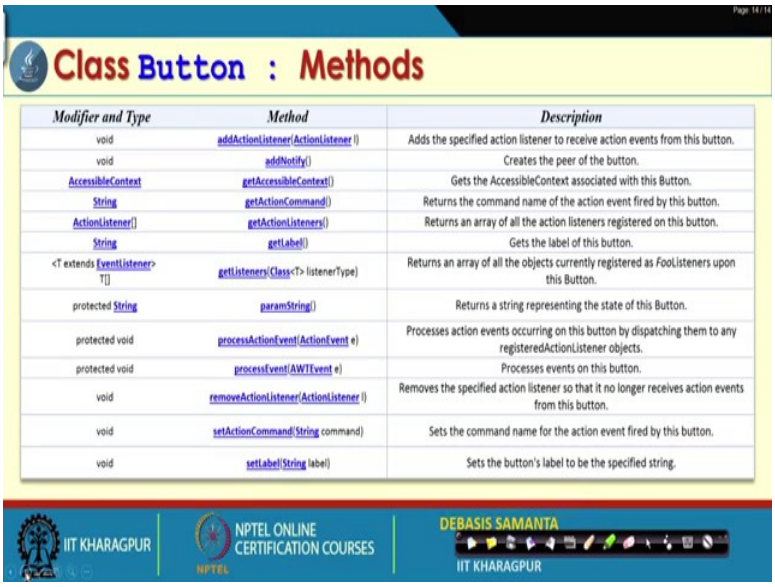


Now, we will discuss about how a button can be created and how this button can be added into frame or container or in an applet like. So, a button as you know it typically look like now whatever it is shown here, a button actually it is look like a button has some label like this one. It has some design style all these things can be done. So, all these things basically you can do it, if you want to have what exactly you want to do you do not have to do much coding only just invoke the different methods which are defined in the component class.

(Refer Slide Time: 25:10)

So, this is the button, button let us see for the button these are the constructors.

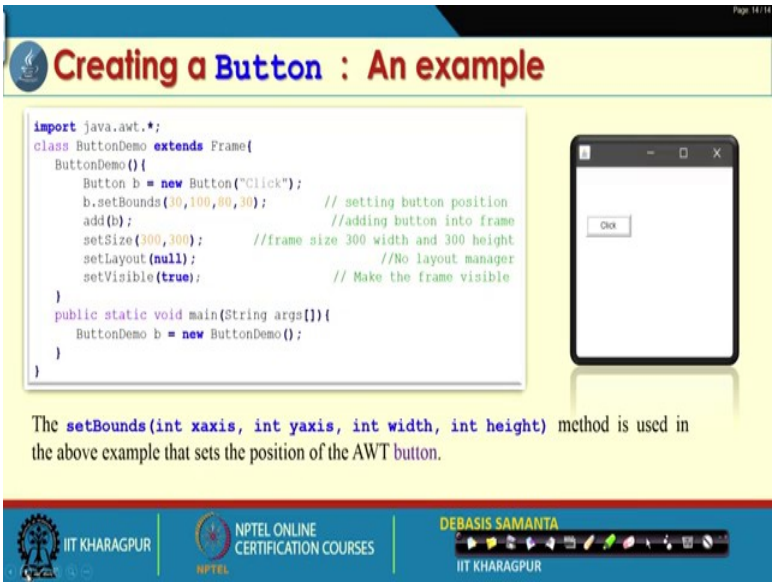
(Refer Slide Time: 25:14)



Modifier and Type	Method	Description
void	addActionListener(ActionListener l)	Adds the specified action listener to receive action events from this button.
void	addNotify()	Creates the peer of the button.
AccessibleContext	getAccessibleContext()	Gets the AccessibleContext associated with this Button.
String	getActionCommand()	Returns the command name of the action event fired by this button.
ActionListener[]	getActionListeners()	Returns an array of all the action listeners registered on this button.
String	getLabel()	Gets the label of this button.
<T extends EventListener> T[]	getListeners(Class<T> listenerType)	Returns an array of all the objects currently registered as *Foo*Listeners upon this Button.
protected String	paramString()	Returns a string representing the state of this Button.
protected void	processActionEvent(ActionEvent e)	Processes action events occurring on this button by dispatching them to any registeredActionListener objects.
protected void	processEvent(AWTEvent e)	Processes events on this button.
void	removeActionListener(ActionListener l)	Removes the specified action listener so that it no longer receives action events from this button.
void	setActionCommand(String command)	Sets the command name for the action event fired by this button.
void	setLabel(String label)	Sets the button's label to be the specified string.

And the methods are also there are lot of methods included here. You can go through this slides and then have the full understanding about the methods and that is, ok.

(Refer Slide Time: 25:21)



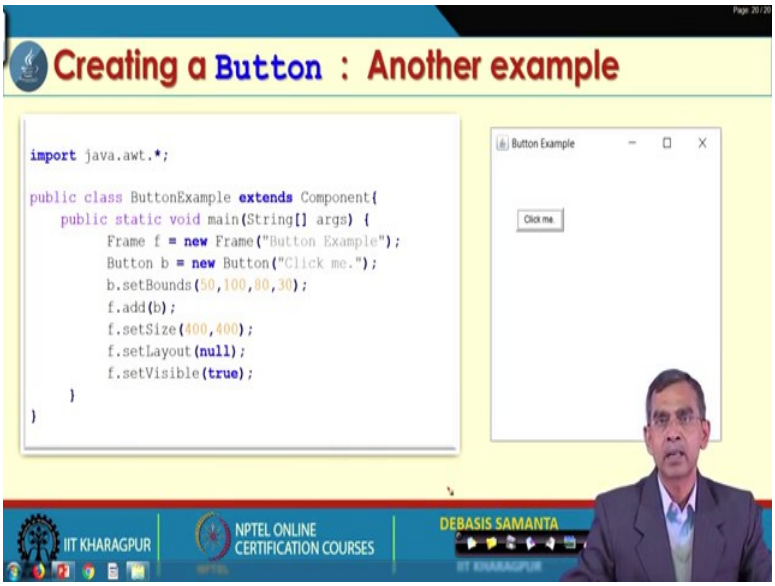
Now, let us see how we can create we can create a button which can be plotted on a frame. So, this example is like this. So, button if you want to create a button. So, this is the standard common that you can follow, and this is the label that button will be. So, it is a click like. And setBounds this methods is basically says that what is the location of

the button this is basically 30, 100 is the this location and 80, 30 is basically this is 80 and this is 30, this is the width and height like this one.

So, setBounds methods is declared in a component class. So, it can use it and finally, add (b); that means, is basically these button class extend frame; that means, it will overwrite on the frame itself. So, basically include in the frame or contain in the frame. So, add b means these button can be added into the frame. And then setSize, setLayout and setVisible are basically for your frame. So, this is related to the frame because it extends Frame all this methods are automatically for this method implies; so this basically the size deciding this one. And then finally, this ButtonDemo b, if we created then this class will be instantiated and the frame will appear look like this.

So, this is the way here the button can be created. Now, in the same way many buttons, with different what is called the labels with different background foreground and then different sizes can different location can be added into this frame. I have included on one example, so that for the illustration only.

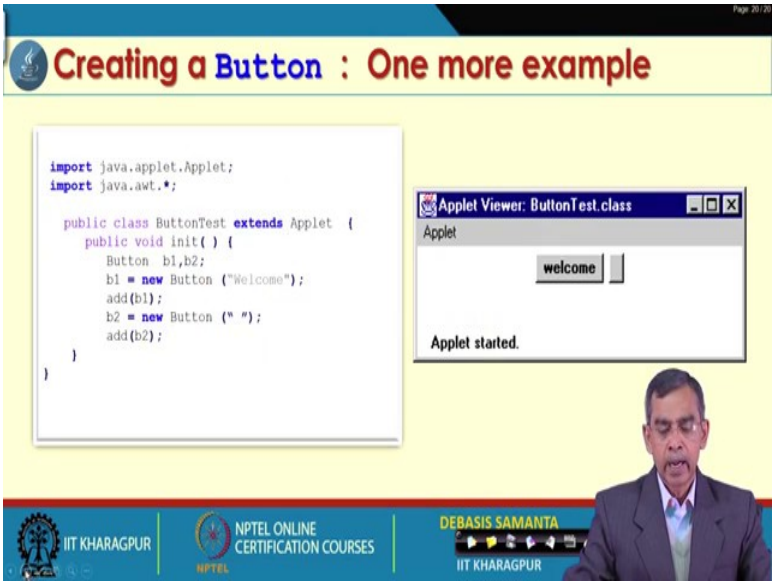
(Refer Slide Time: 26:57)



So, this is the button, and then button can be also included in a container that is a component without any frame or panel it is like this is the same example. Like we create a Frame f and this is a button example and then Button b is created and then all these methods are there. Now, here see button example extends component as is the frame is

also sub class of this component; so all those things will be there. So, this is the one way, another way of creating button or adding component into your content container.

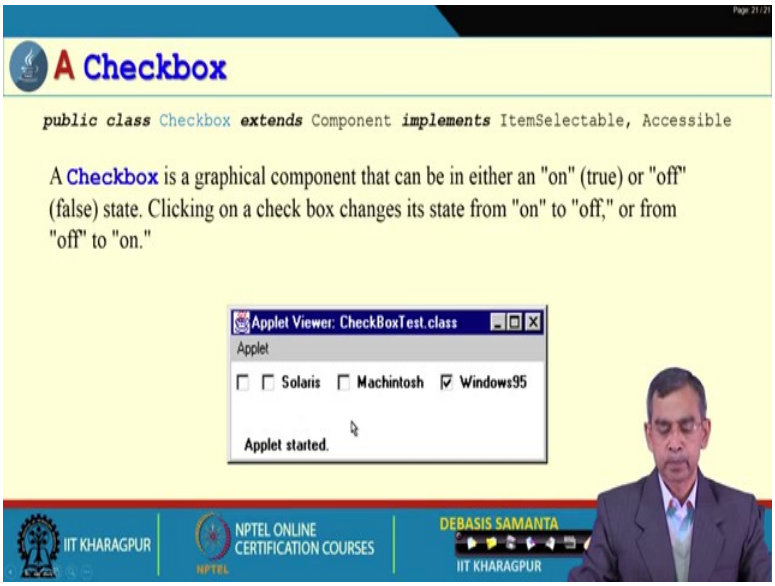
(Refer Slide Time: 27:32)



Now, so and another also how we can add the GUI into applet this example explain it. So, in that case you have to develop the class which inherit the applet class then is basically applet will be there, so applet will be created. And all these method here (Refer Time: 27:47) b1 and b2, two buttons are created, one button having the label another button without label and they can be added into his applet and it will look like this. So, button can be created, the button can be added into the frame, button can be added into the container which basically is a frame or panel and button can be added into the applet also.

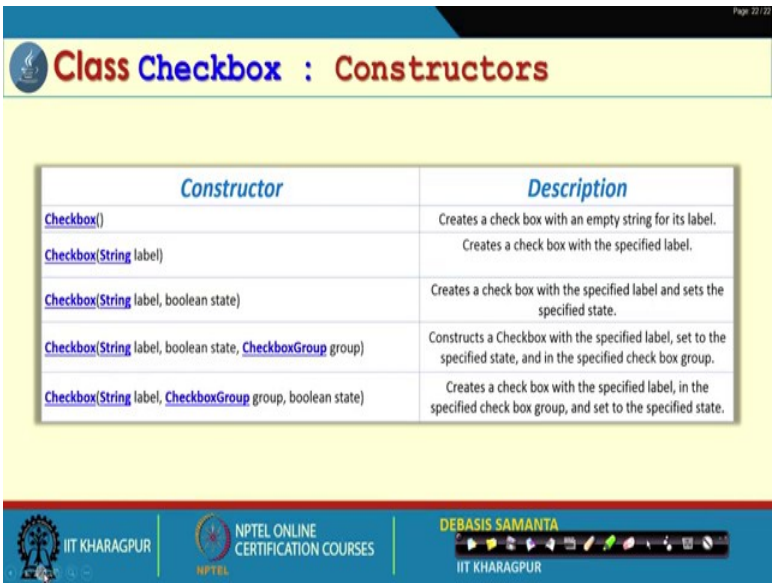
So, there are the different way; this an example that button can be like this button there are many other component like checkbox and all these things that we are going to discuss quickly. So, all these things are also can be included in your windows.

(Refer Slide Time: 28:21)



Now, if the checkbox how a checkbox look like. So, typically a checkbox is called like this one a checkbox has this kind of things and then there is a label and then this can be clicked or keyboard can be pressed there and then whatever there. So, this checkbox can be selected. For example, this checkbox is selected if selected it will keeps there. So, here we see 4 checkboxes are there which basically contain in an applet like.

(Refer Slide Time: 28:55)



Now, so, so now, let us see how the checkbox can be created. Definitely there is a class for the checkbox and for this class there is a constructor, and for there are some method also there.

(Refer Slide Time: 29:00)

Class Checkbox : Methods

Modifier and Type	Method	Description
void	add(String item)	Adds an item to this Choice menu.
void	addItem(String item)	Obsolete as of Java 2 platform v1.1.
void	addItemListener(ItemListener l)	Adds the specified item listener to receive item events from this Choice menu.
void	addNotify()	Creates the Choice's peer.
int	countItems()	**Deprecated.** *As of JDK version 1.1, replaced by getItemCount().*
AccessibleContext	getAccessibleContext()	Gets the AccessibleContext associated with this Choice.
String	getItem(int index)	Gets the string at the specified index in this Choice menu.
int	getItemCount()	Returns the number of items in this Choice menu.
ItemListener[]	getItemListeners()	Returns an array of all the item listeners registered on this choice.
<T extends EventListener> T[]	getListeners(Class<T> listenerType)	Returns an array of all the objects currently registered as *FooListeners* upon this Choice.
int	getSelectedIndex()	Returns the index of the currently selected item.
String	getSelectedItem()	Gets a representation of the current choice as a string.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

These are the checkbox method.

(Refer Slide Time: 29:02)

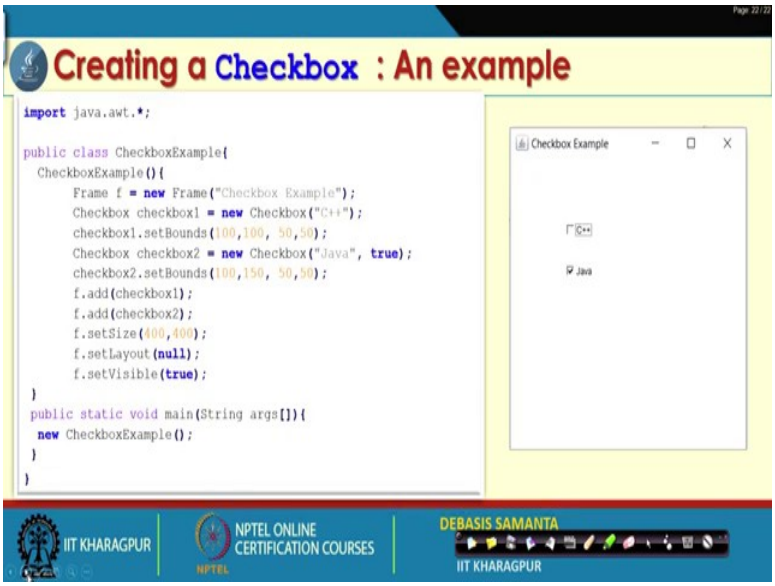
Class Checkbox : Methods

Modifier and Type	Method	Description
Object[]	getSelectedObjects()	Returns an array (length 1) containing the currently selected item.
void	insert(String item, int index)	Inserts the item into this choice at the specified position.
protected String	paramString()	Returns a string representing the state of this Choice menu.
protected void	processEvent(AWTEvent e)	Processes events on this choice.
protected void	processItemEvent(ItemEvent e)	Processes item events occurring on this Choice menu by dispatching them to any registered ItemListener objects.
void	remove(int position)	Removes an item from the choice menu at the specified position.
void	remove(String item)	Removes the first occurrence of item from the Choice menu.
void	removeAll()	Removes all items from the choice menu.
void	removeItemListener(ItemListener l)	Removes the specified item listener so that it no longer receives item events from this Choice menu.
void	select(int pos)	Sets the selected item in this Choice menu to be the item at the specified position.
void	select(String str)	Sets the selected item in this Choice menu to be the item whose name is equal to the specified string.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

And this is an example how we can add checkbox into a frame.

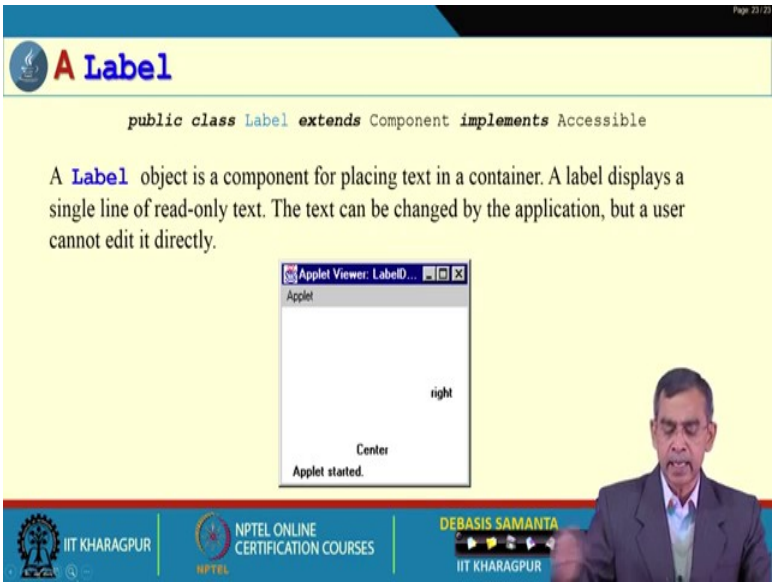
(Refer Slide Time: 29:04)



It is basically same thing the button look like. So, this basically two checkboxes are created the C++ and this checkbox is Java. So, this is the way the checkbox can be created. And as you have created a frame, so this is basically frame. So, this is our frame and these are the two checkboxes as we have C++ and Java, and whenever you make true so that means, a default selection is there and if you do not make it is basically there that will be uncheck like; and then we can add this one.

So, f **dot** add **(**checkbox1**)**, f dot means these are the two checkboxes that we have defined can be added into the frame and finally, frame can be resized. So, this is the concept that, checkbox can be designed. And you can see all these ideas basically same only the different class corresponding the different class the different component can be realized.

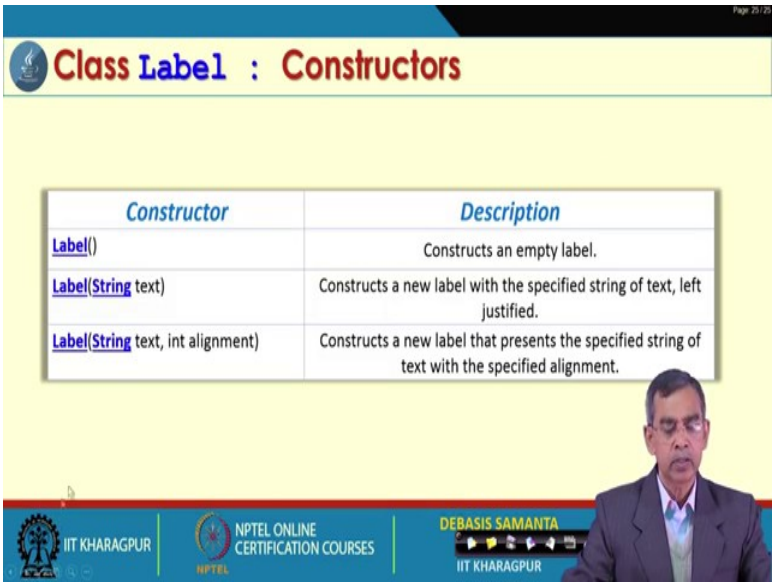
(Refer Slide Time: 30:03)



Now, Label, it is basically just like label is basically a component is like here you can see this is an applet and this is a label this is the one label. Label means anything that we can write you know in the applet. Like say g dot drawstring if we using g dot drawstring also some label can be putted here, but label is the one class which is there which is explicitly used for making different label.

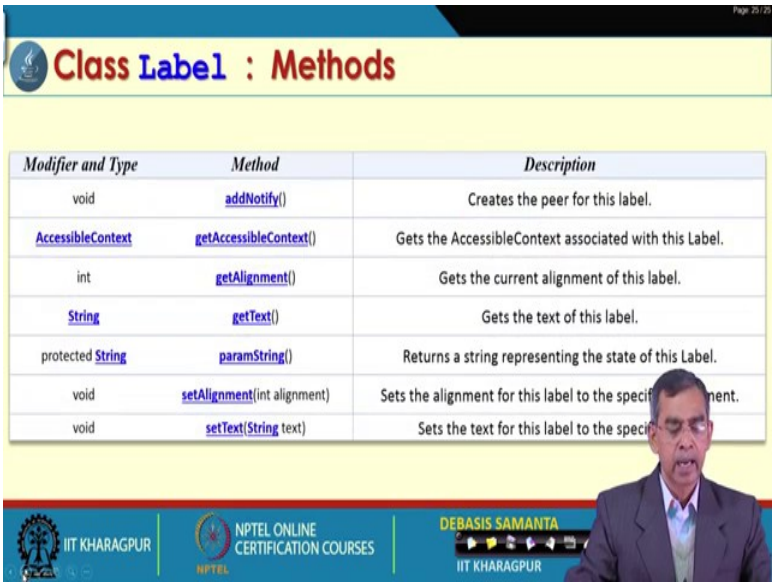
For example, I can write say first name, last name, roll number and these are the label. So, first name, last name, roll number we can write and then some other things can be included there like. So, label is just basically string which can be floated on a on a screen like. So, now, let us see how the label can be created.

(Refer Slide Time: 30:43)



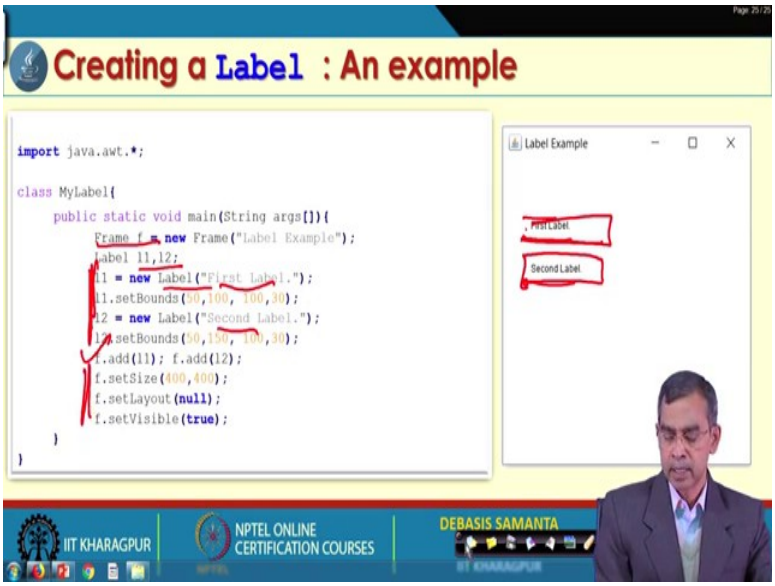
So, there are constructors here, you can see 3 constructors are there in the label class.

(Refer Slide Time: 30:49)



And then few methods are also there.

(Refer Slide Time: 30:50)



And this is an example by which the label can be created a see this is basically frame it is all obvious two Lables, l1 and l2 are created and they are redefined here using this one; that means, first level second level, as you see this is the first level, second level. And this is the setBounds basically size; that means, where it will be placed and where how long it will be placed and then how long it will be. So, basically total with and this everything is been mentioned here; adding into this frame and finally, frame is size. So, this is the idea about how the frame can be label can be added into frame.

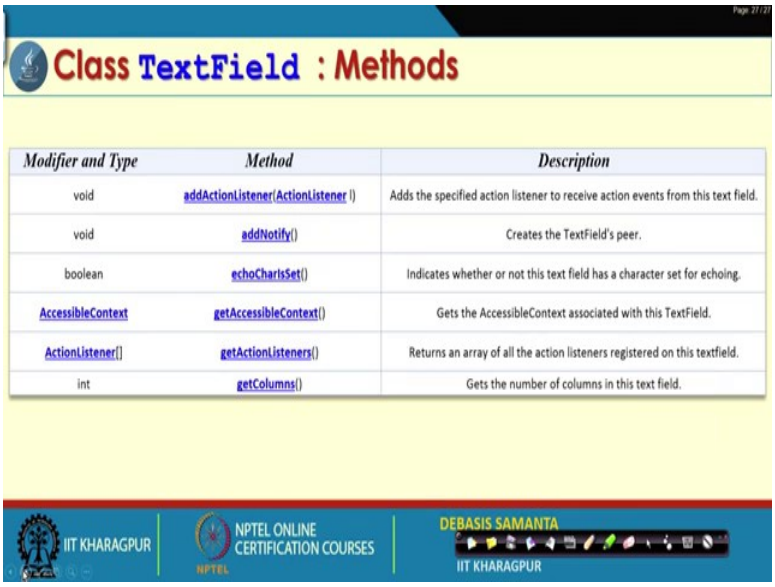
(Refer Slide Time: 31:27)

Textfiled, this is the idea about the TextField. As you see texfield is looks like this. So, this is the one TextField area, this is the another TextField area, this is the another TextField area; so this is an applet which contains field TextFields in each.

(Refer Slide Time: 31:46)

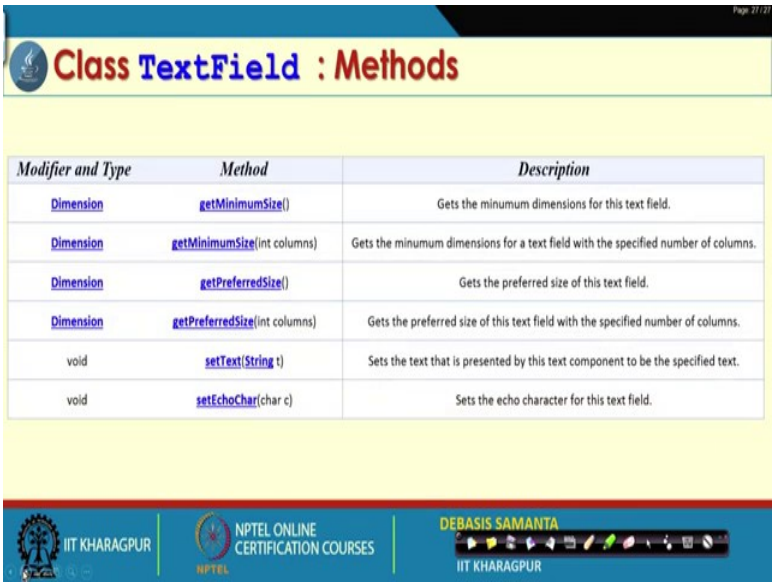
Now, let us see how this TextField can be created. Again, there is a class called TextField that is defined in the components as a subclass of the component class. It has 3 constructors and then few methods are there.

(Refer Slide Time: 31:52)



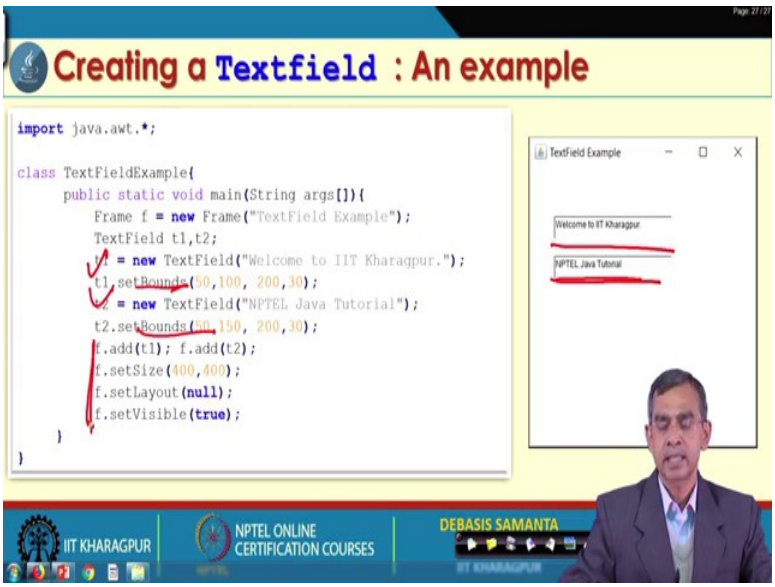
And this is the one ask few more methods are also there.

(Refer Slide Time: 31:55)



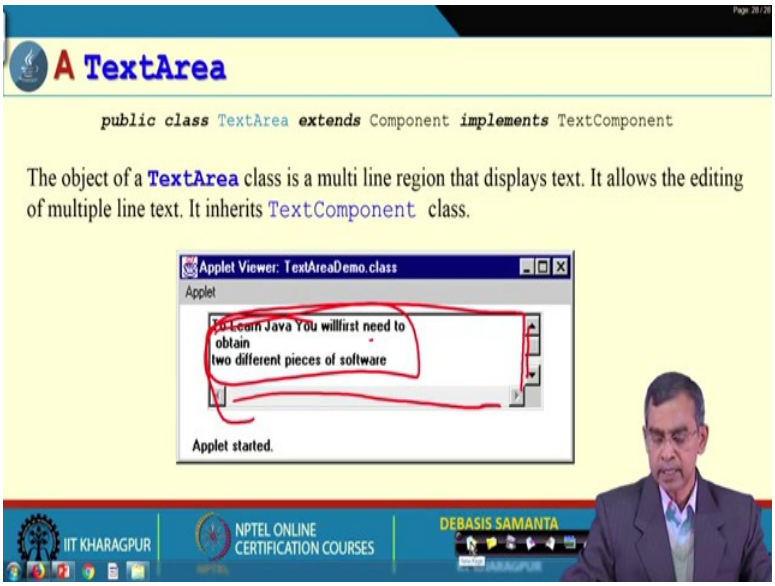
And this an example you can test it how that two text field like two TextField like this one and this one is can be used.

(Refer Slide Time: 31:58)



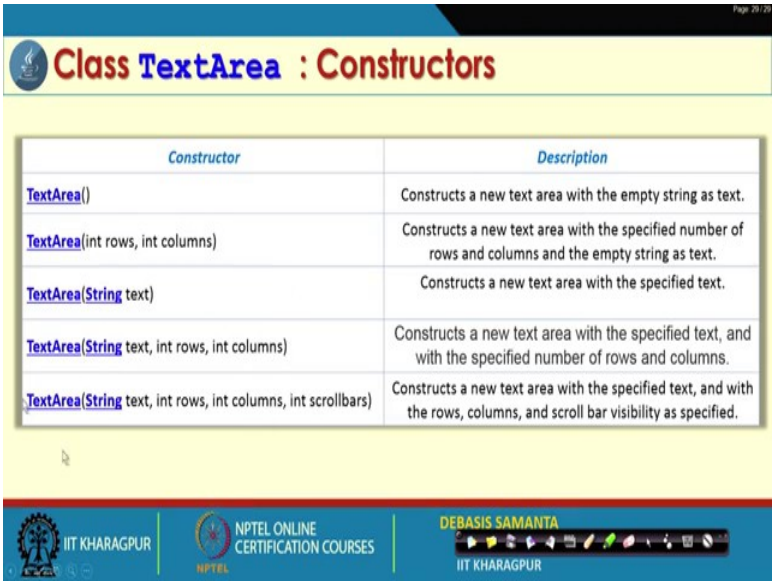
So, welcome to IIT, Kharagpur, NPTEL, Java. So, we just created t1 and t2 are TextField and for every component you see setBounds to decide the location of the placement of this elements into the container as well as the size and everything. So, setBounds is always there and finally, this is the frame should be casted, and then frame can be visible, can be displayed.

(Refer Slide Time: 32:40)



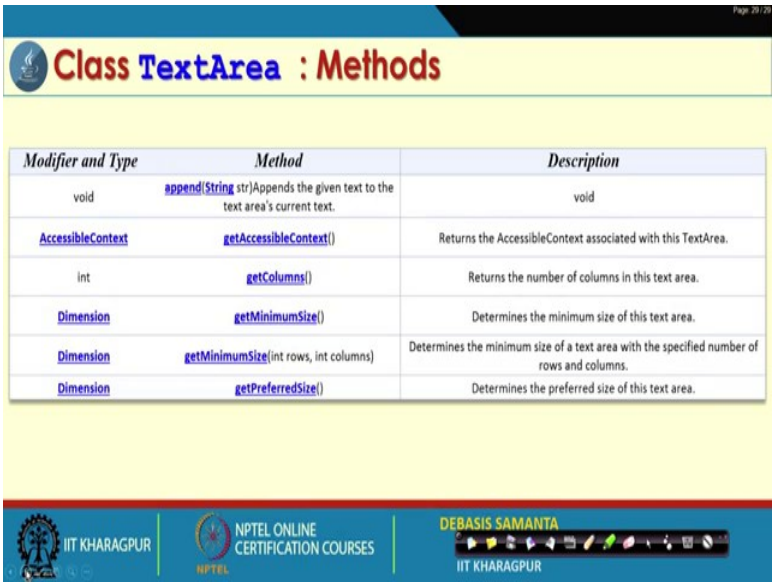
Now, so this is the way that TextField can be created. And TextArea is just like a TextField, but multiple line very long text can be used there and you see whenever you text area is there it has the scrollbar automatically will be there. So, we can define everything and within this thing any text can be out and then displayed.

(Refer Slide Time: 33:03)



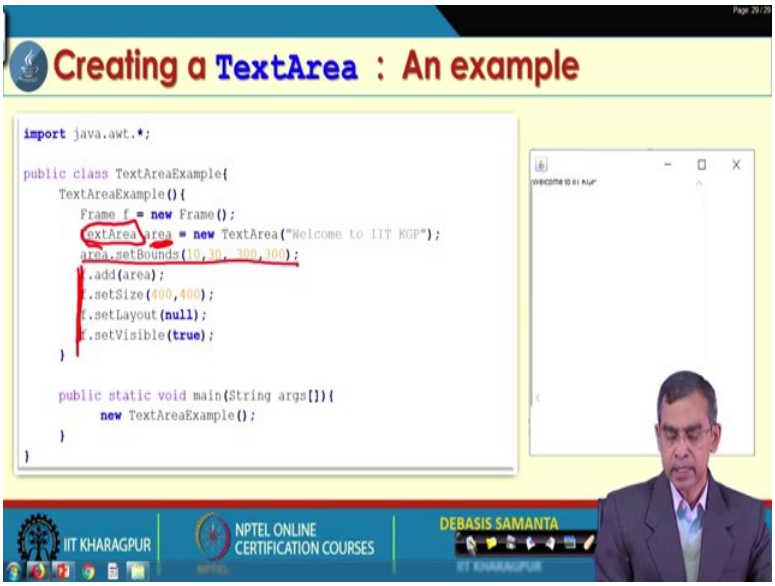
So, idea of this TextArea is like this and for this TextArea if you want to create the TextArea, there is a class defined in called the class name is TextArea. It has the constructor as we have listed here.

(Refer Slide Time: 33:09)



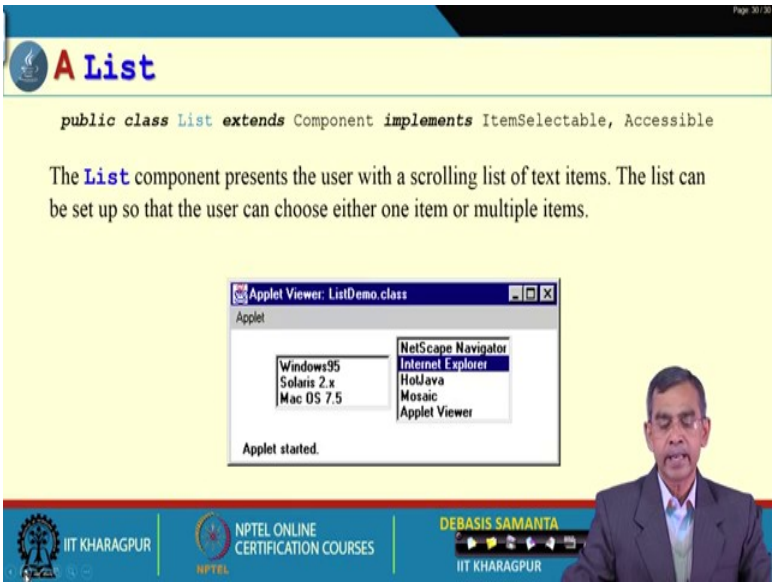
And few methods are also here.

(Refer Slide Time: 33:10)



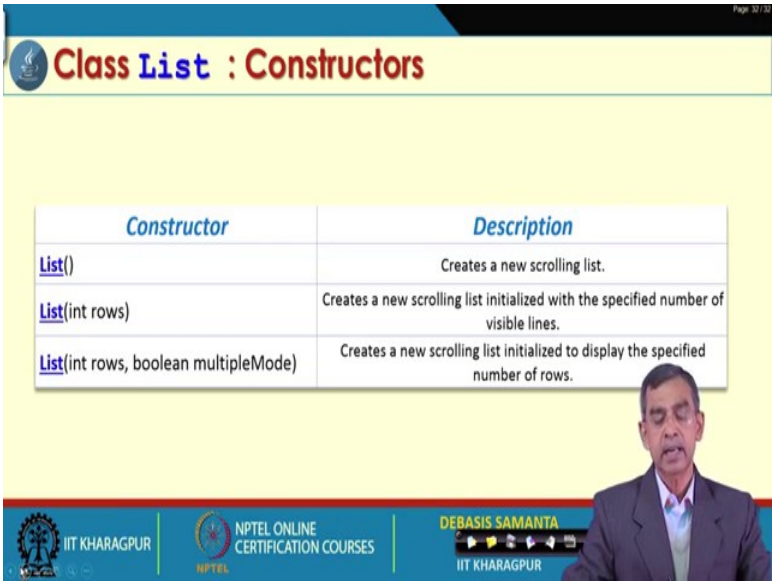
And this is an example by which a text area can be displayed on this. The method is there, the TextArea, this is the class that we have, this is the name of the TextArea and then setBounds and then added into this frame and then frame can be resized. So, this way the TextArea can be displayed on the screen.

(Refer Slide Time: 33:39)



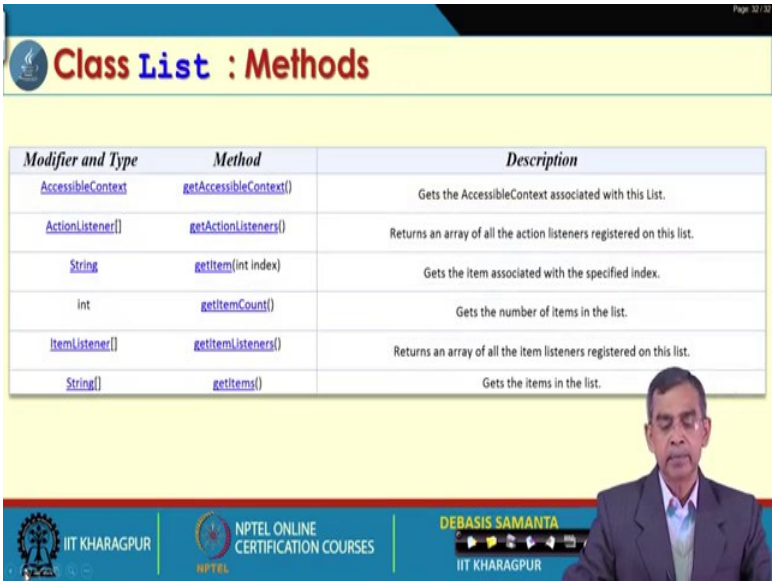
Now, List, it is another item another graphical user interface item we can say. A List typically look like this. So, here two list we have displayed here, this is the one list contents some elements it is just like TextArea, but it is called the list this is another list. And the idea is that we can scroll we can just, over our mouse to select any one list. For example, here the mouse whenever come here it will highlights and this basically list is selected we can say; so these are the example of a list.

(Refer Slide Time: 34:13)



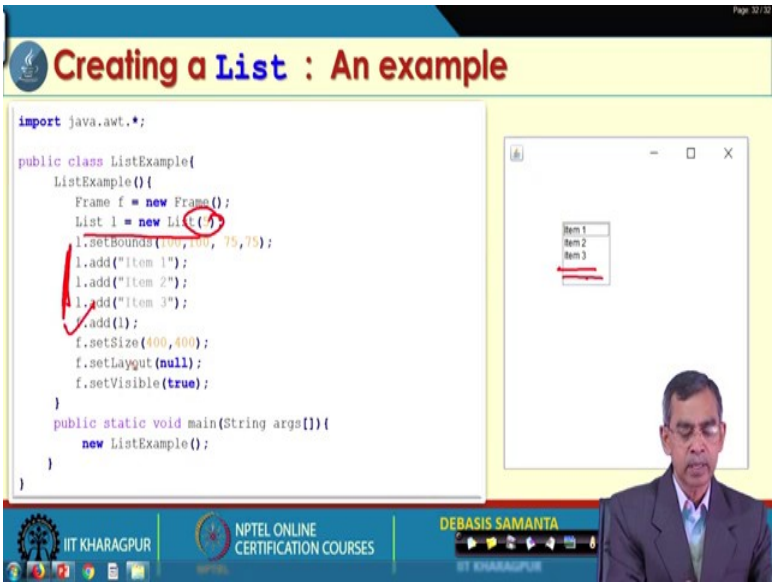
And the class definition is also there the name of the class is list. This, class has a constructor as it is mentioned here.

(Refer Slide Time: 34:17)



And then method also, these are the methods are there.

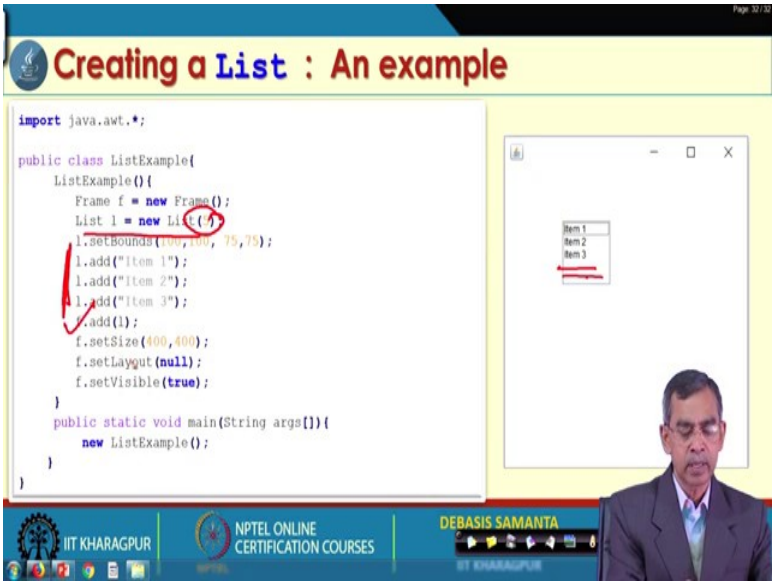
(Refer Slide Time: 34:20)



And then, this is an example which basically shows how the list with 3 items can be created. So, it is the list basically the 5 items so; that means, it has 5 items, right. And, but although we have created 5, but we have loaded with 3 only, so 2 more are blank

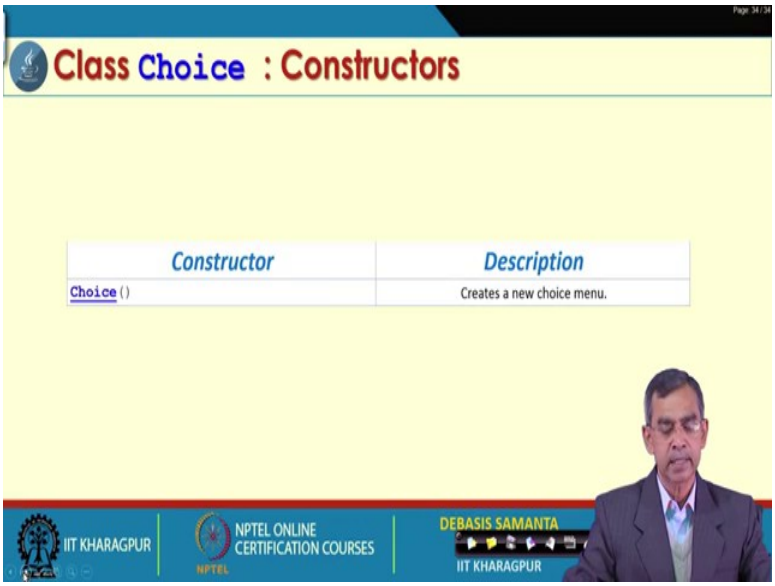
actually anyway. So, we have added 3 items in the list and they are added into the frame and then frame is displayed on the screen. So, this way a list can be created.

(Refer Slide Time: 34:52)



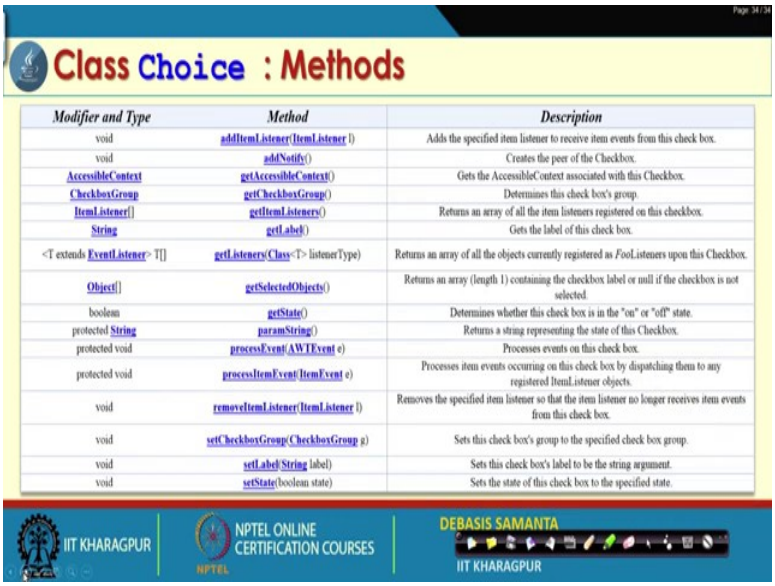
And then Choice is also similar to list. So, idea is that initially a choice will look like this only and whenever you click here all the elements which is there in this choice will be displayed here and then you can hover your mouse to go to particular choices and then we can select that one, whichever the choice is under focus can be displayed here. So, this is the idea about this one and for this kind of graphical user interface the class which is there in the AWT is called the choice class.

(Refer Slide Time: 35:23)



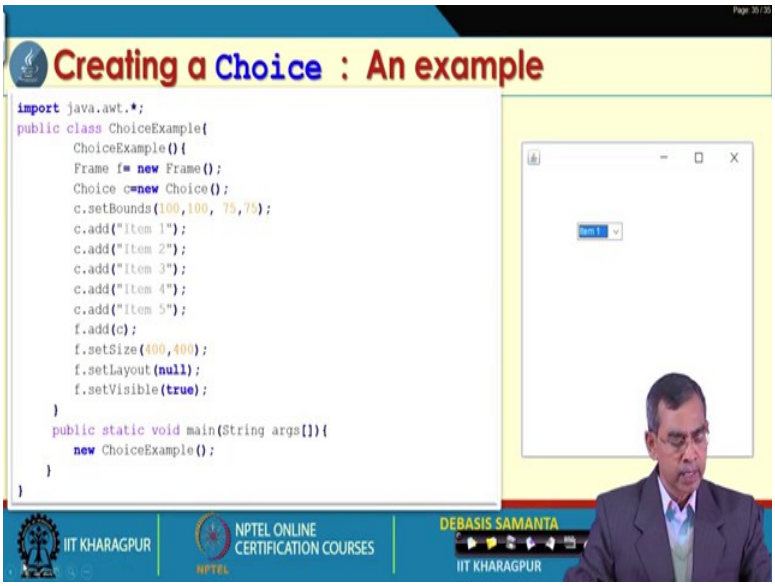
It has only one constructor.

(Refer Slide Time: 35:24)



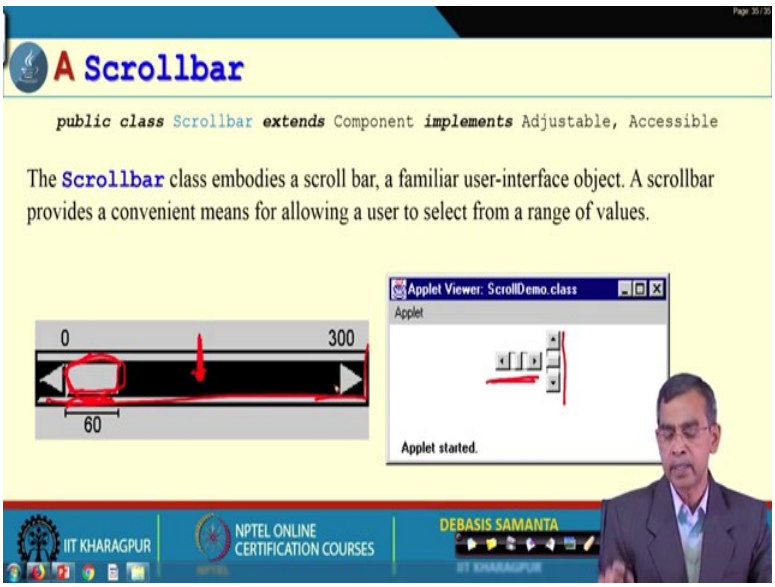
These are the so many methods are there.

(Refer Slide Time: 35:26)



And here is an example about how we can create a choice with 5 elements in it. So, this example is similar to the list example that we can see. We have created a choice, add this choice into the frame, and then frame is ready to use it. So, this way a choice items can be created and then it can be included in the either is a frame or it can be included in applet or it can be included in any other container itself.

(Refer Slide Time: 35:54)

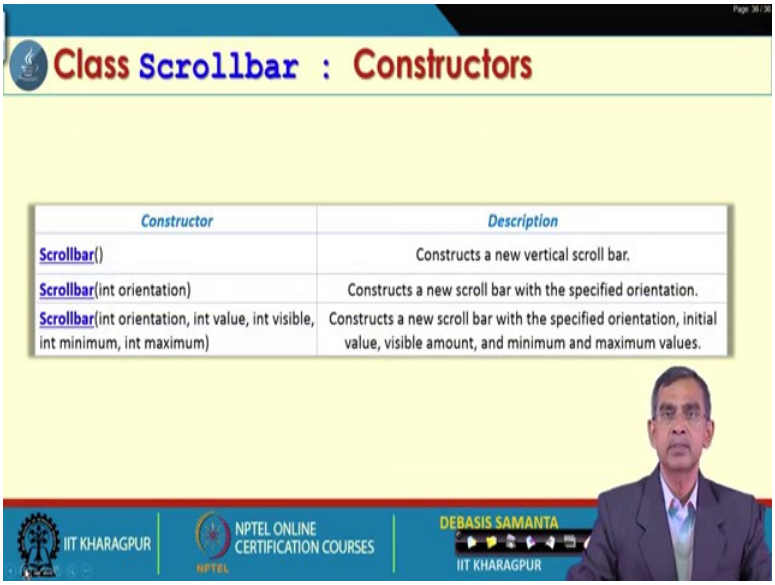


Now, here is a Scrollbar, we are already get familiar about the scrollbar, but independently we can use some our scrollbar. So, this is the vertical bar and this is the

horizontal scrollbar. And then scrollbar has few things are there this is basically called the visibility portion, this is the range; that means, scrollbar will range from 0 to 100. So, if we click here, so whatever the value say may be 150 will be the value of this point actually and this is basically the visible space is called the 60 pixels and here also scrollbar has width and height and everything.

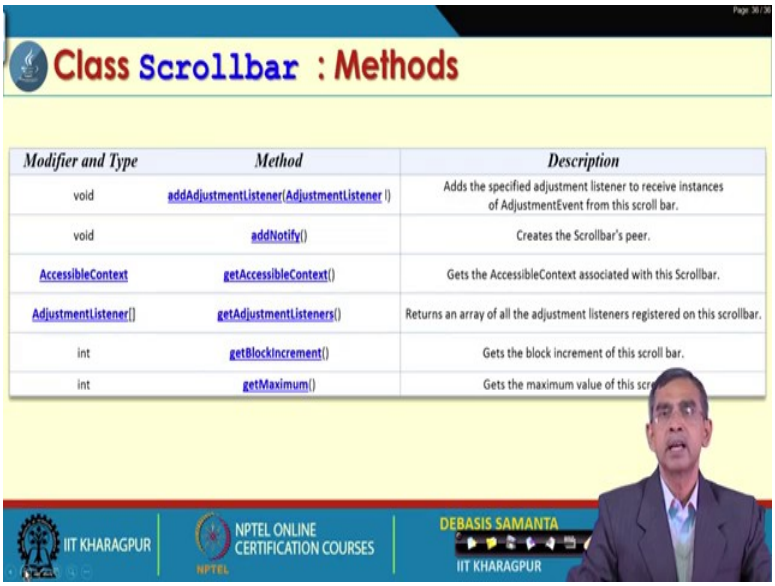
So, these are the elements that is the scrollbar and it can be a vertical as well as horizontal, as you see vertical this is the vertical one this is the horizontal one.

(Refer Slide Time: 36:40)



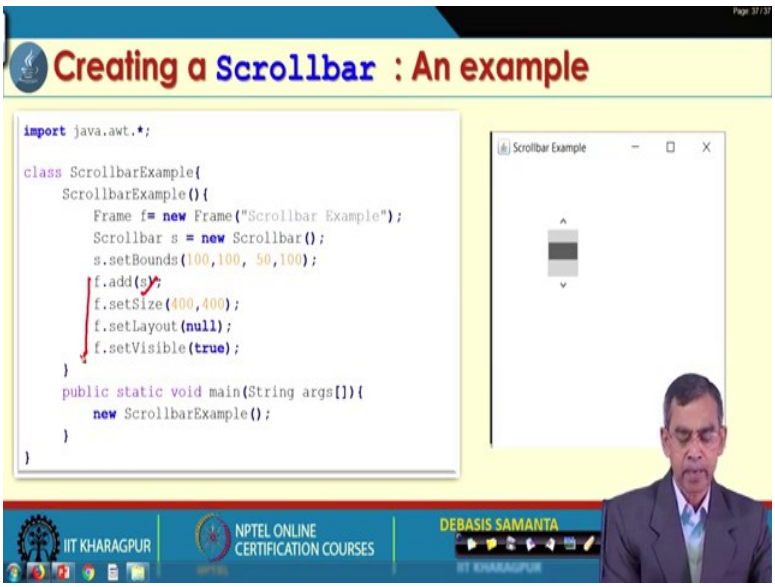
Now, let us see how the scrollbar can be created. Scrollbar for this there is a class called the Scrollbar class we in AWT package. It has 3 constructors, and few methods are there.

(Refer Slide Time: 36:44)



And then this is an example, and this example shows how a horizontal vertical scrollbar cab be created.

(Refer Slide Time: 36:46)



Now, here frame this is as usual previous there and scrollbar this is the instance by which a scrollbar objects can be created. And you see the method we have used the default method, if we use the default method a it will automatically the vertical scrollbar only and no other parameters are there, but here we can define the vertical or horizontal, and then the size, and then visibility, and the range all these parameters can be defined. There

are 3 constructor we have shown using any one constructor the different way the scrollbar can be instantiated. And then this scrollbar can be added into the frame, the frame can be resized, frame can be displayed. This is basically a simple AWT program that can use the scrollbar, and then you can use it.

So, this is the scrollbar that we have discussed about it. And few elements that we have discussed, there are many more elements that we are yet to cover it. So, in our next module we will discussed about many other components those are very much essential for our AWT programming.

Thank you.