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

Lecture – 42
A W T Programming – III

Java is truly exceptional because it offers many supports to the programmers one offering in the sense is A W T package. AWT package helps a programmer to develop window programs very convenient conveniently and we have discussed many features those are there in an AWT package and today we are going to discuss more important possibly to my understanding is very important one features it is called the event handling.

So, today our topics basically to discuss event handling mechanism in Java, this is essential for any window based program development, now first we will discuss about the concept of the event.

(Refer Slide Time: 01:08)

And as we know exactly what is an event in fact, in the context of any program execution. So, as we know we are already familiar with few things about say applets and as we see in case of applet there is a keyboard like right and then the applet contains say calculator applet it contains a key now that key can be typed or key can be pre-pressed.

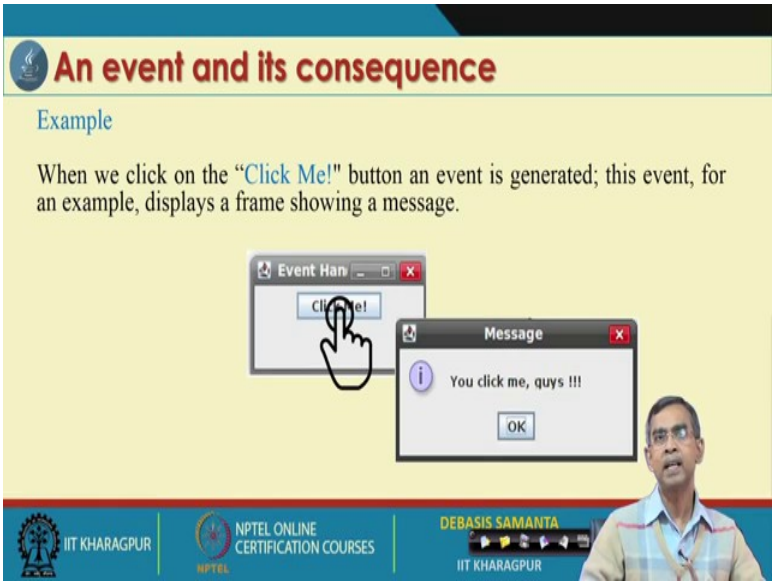
Now whenever key can be pressed then it will basically sense that which key has been pressed. Similarly, if we move a mouse and then point a mouse on a component like say button then and then the button is clicked, then the event is occurs in the sense that button has been clicked and then for this event there will be some execution and then system will do something whenever a user clicked a button.

Now, this basically is an event look like, now so for the graphical leisure interface is concerned even it looks like this, but the event has more many implications in many application areas. For example, when you are driving a car or a driver there are many things are to be controlled and actually whenever the driver wants to control, for example, to break a car. So basically, an event recreates similarly if we accelerate then another event is created.

Now, in case of driving a car at the same time, many events can be generated can be triggered, then the car has those machineries the mechanism by which all the events which occur at any instant at any moment can be handled. So, there is basically handling the events by means of some hardware mechanism or by means of some software mechanism.

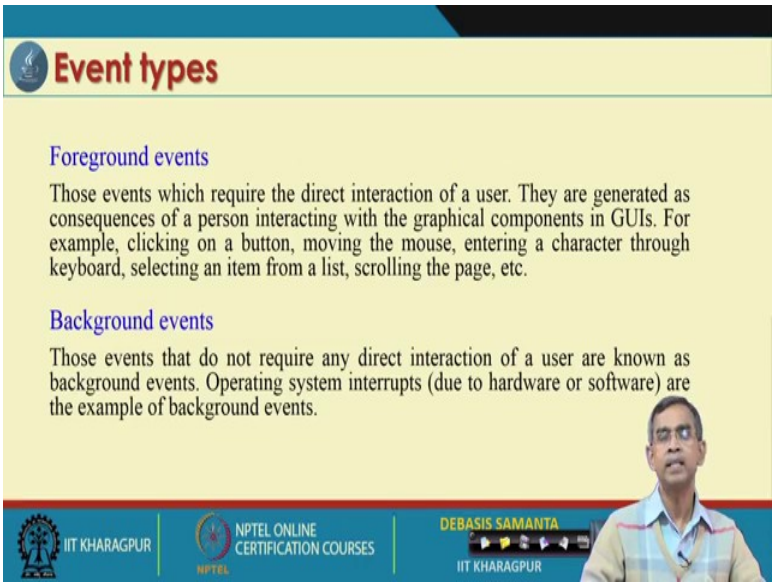
So, event handling possible in Java is basically to manage the event using some software mechanism and that is also particularly in the context of GUI.

(Refer Slide Time: 03:31)



Now, here exactly this is a small example that you can see as you see here this is one frame or you can say a plate which contains a button which is a Click Me. Now if the user clicks this button then what will happen automatically the system can sense that user has clicked a button and then it pops up a message another frame may be which is shown here. So, it is basically the event that may be triggered by the user and corresponding to the event there will be some action that will be performed by the system.

(Refer Slide Time: 04:12)

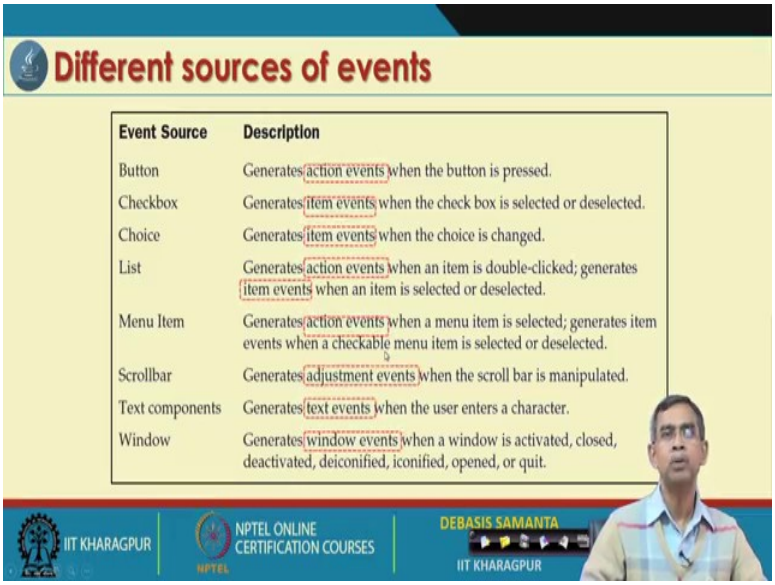


So, this is the idea about the events so for the event different types of events are concerned there are two all the events that can be categorized into two broad categories they are called the foreground event and background event. The events that we are going to handle or discuss, this is in the context of graphically or interface is called the foreground event.

On the other hand there are some events occurs in our computing system allow it they are called the background event like operating systems handles background event whenever press say any start button or alt tag alt to delete or there is a some response from the timer or some other software when it is executing it faces some execution errors divide by 0 errors like it basically causes some interaction we say and this interaction is basically triggering an event and corresponding to the event system know exactly what to do. So, this is called the background event.

Now, we will discuss mostly the foreground event and in Java what are the event handling facilities that are most important.

(Refer Slide Time: 05:19)

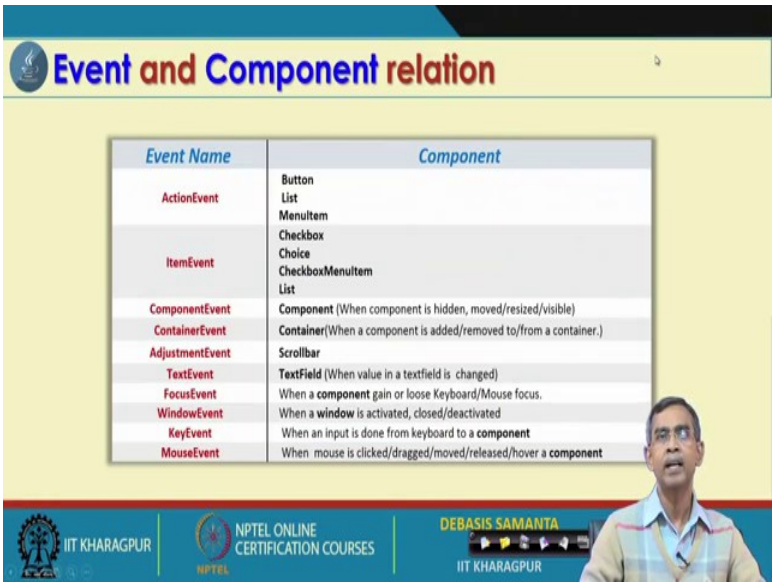


Now, in the context of graphical user interface as we know a container may contains many components here I have listed whatever the main components are possible that is in GUI like button, checkbox, choice and all these things are there.

Now, so, here whenever a button is there this is basically button, checkbooks, the choice all these components are created as a source of the event, that mean event can be generated from this points only. An event can be generated from the scrollbar and event can be generated from the list item, an event can be generated from the checkbox like this. Now whenever a particular source of the event is excited; that means, triggered then that basically the automatically one program that is there in the Java system which basically watches that whether an event occurs in any component or not.

Whenever it says that some event has been triggered has been occurred, then it generates one event and there are many events corresponding to the many different components. For example, as we see in this slides from the bottom if it is a source it generates action events. Similarly, if it is the list then also it is an action event wherever it is a check item or choice the events that are generated automatically is called the item events likewise scrollbar, it is called the adjustment event or text field it is for the text event.

(Refer Slide Time: 06:52)

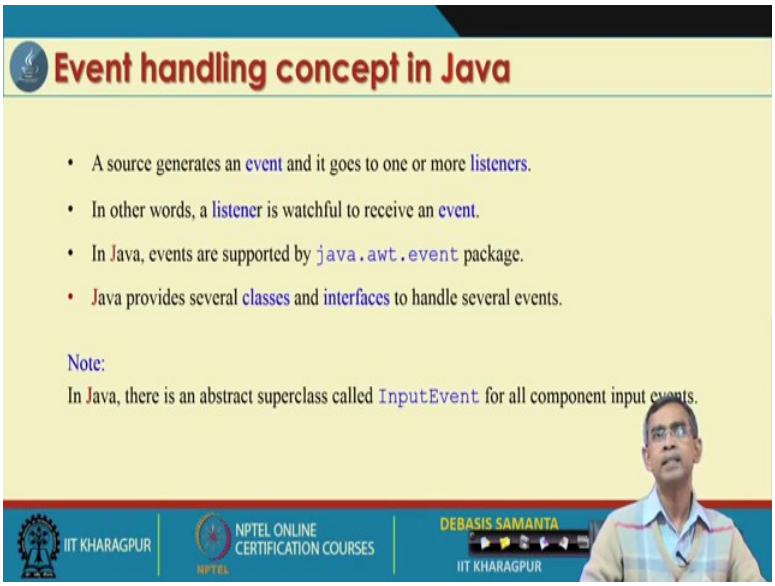


What I want to say is that all the events are basically categorized according to that here. So, what are the possible events that are possible in a GUI is listed here, as you see action event, item event, component event and then container event, adjustment event, text event focus, event windows event key event and mouse event.

The last two events are very much important because usually user gives any event I mean creates your creates many events using these two device devices namely keyboard and mouse. So, if the event is generated by the keyboard it is called the key events if the event generated by the mouse it is called the mouse event and you note that they are not exactly the component, but they can interact with the component that is the thing what I want to me mention here.

And all the action items for example, related the button, list and then menu items like this similar item event is related to checkbox, choice, checkbox or checkbox menu item list like this. Now this list is very important you should know that which is the name of the event corresponding to the which is the source, this slight in this is very important and you should have the complete familiar about because the event that I have mentioned is created by a particular class which is defined in AWT package.

(Refer Slide Time: 08:19)



Now, I am going to discuss about what are the different classes and interfaces that are possible in an event. In a nutshell, what I want to mention here is that a source like a say button or a checkbox or any window whatever it is there can generate an event and whenever there is an event occurs it goes to listener items. Now, therefore, we should have the two things in your mind, that event and correspond to an event the listeners know what exactly the listener it is.

Now, event will be triggered by the user and then there should be some mechanism by which if an event occurs any time at any moment it should be listened by someone else. So, listener is another what is called the object I should say which basically listens to any event if it occurs at any instant anywhere.

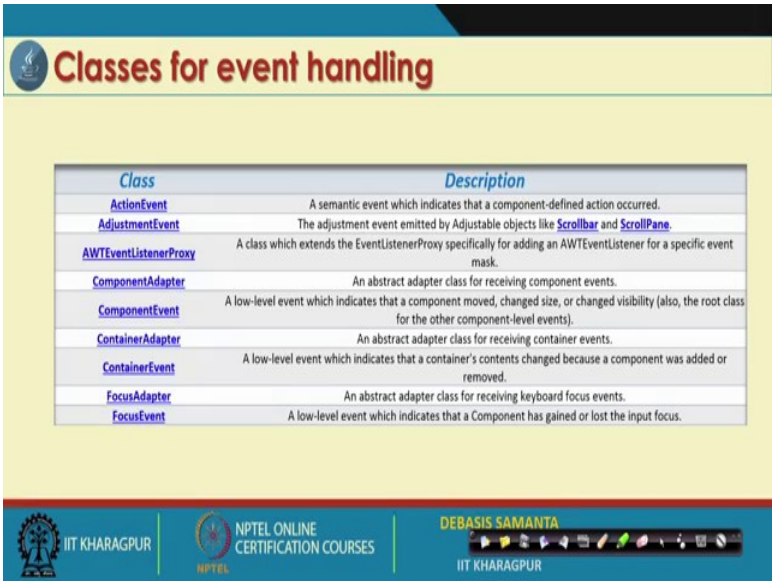
Now, so there is a listener who is watchful to receive any event if occurs in the device in any windows elements. Now in Java all these events are basically; that means, that the see basically whenever a source the source can create an event. So, it basically create an object actually, the object is of type event and that object is defined I mean type of the object is basically event, this is defining java.awt.event is a class, so there everything is defined there.

Now, here the most important thing so for the java.awt.package is concerned is that all the classes and interfaces which are basically responsible for either creating an event

whenever a source is interacted by a user or whenever an event is created is listened by the listener.

So, these two things; that means, the event creation and then listening to an event is basically handled by two items in the Java, that classes and interfaces which are defined in java.awt.package.

(Refer Slide Time: 10:24)



Class	Description
ActionEvent	A semantic event which indicates that a component-defined action occurred.
AdjustmentEvent	The adjustment event emitted by Adjustable objects like Scrollbar and ScrollPane.
AWTEventListenerProxy	A class which extends the EventListenerProxy specifically for adding an AWTEventListener for a specific event mask.
ComponentAdapter	An abstract adapter class for receiving component events.
ComponentEvent	A low-level event which indicates that a component moved, changed size, or changed visibility (also, the root class for the other component-level events).
ContainerAdapter	An abstract adapter class for receiving container events.
ContainerEvent	A low-level event which indicates that a container's contents changed because a component was added or removed.
FocusAdapter	An abstract adapter class for receiving keyboard focus events.
FocusEvent	A low-level event which indicates that a Component has gained or lost the input focus.

Now, I will discuss first about event handling class truly there are many different types of events possible and according to these events each events there are different classes are there. Now, for example, there is one class called action event, now action event is class basically responsible to hand to create an action event. Now action event occurs because of the button, because of the checkbox, because of the choice like this one. So, as we have already discussed that this is a different event from this different source.

Now, here again this continues or adapter, this is another type of event that occurs from the container itself if there something changes happens or that means, any event is triggered there.

(Refer Slide Time: 11:01)

Classes for event handling

Class	Description
HierarchyBoundsAdapter	An abstract adapter class for receiving ancestor moved and resized events.
HierarchyEvent	An event which indicates a change to the Component hierarchy to which Component belongs.
InputEvent	The root event class for all component-level input events.
InputMethodEvent	Input method events contain information about text that is being composed using an input method.
InvocationEvent	An event which executes the run() method on a Runnable when dispatched by the AWT event dispatcher thread.
ItemEvent	A semantic event which indicates that an item was selected or deselected.
KeyAdapter	An abstract adapter class for receiving keyboard events.
KeyEvent	An event which indicates that a keystroke occurred in a component.
MouseAdapter	An abstract adapter class for receiving mouse events.
MouseEvent	An event which indicates that a mouse action occurred in a component.
MouseMotionAdapter	An abstract adapter class for receiving mouse motion events.
MouseWheelEvent	An event which indicates that the mouse wheel was rotated in a component.
PaintEvent	The component-level paint event.
TextEvent	A semantic event which indicates that an object's text changed.
WindowAdapter	An abstract adapter class for receiving window events.
WindowEvent	A low-level event that indicates that a window has changed its status.

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

So, likewise there are few more classes also here and I have listed all the classes that are for your information and I should advise you to go through all the classes those are there which basically causes an event. And whenever you are whenever dealing with a particular elements particular components, then corresponding to those particular components which event may occurs and corresponding to that event which class is responsible that information you should have, so that you can do all the programming correctly and effectively. So, these are the classes that I have already listed here.

(Refer Slide Time: 11:35)

Classes for event handling : ActionEvent

An **ActionEvent** is generated when a button is pressed, a list item is double-clicked, or a menu item is selected. The **ActionEvent** class defines four integer constants that can be used to identify any modifiers associated with an action event:

`ALT_MASK, CTRL_MASK, META_MASK,` and `SHIFT_MASK`.

In addition, there is an integer constant, `ACTION_PERFORMED`, which can be used to identify action events

Constructors
`ActionEvent(Object src, int type, String cmd)`
`ActionEvent(Object src, int type, String cmd, int modifiers)`
`ActionEvent(Object src, int type, String cmd, long when, int modifiers)`

Here, `src` is a reference to the object that generated this event. The `type` of the event is specified by type, and its command string is `cmd`. The argument `modifiers` indicates which modifier keys (ALT, CTRL, META, and/or SHIFT) were pressed when the event was generated. The `when` parameter specifies when the event occurred. The third constructor was added by Java 2, version 1.4

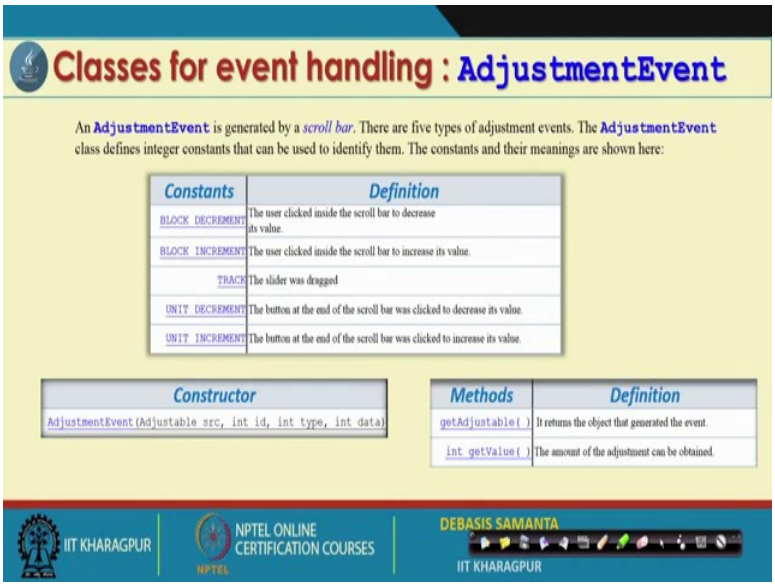
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA, IIT KHARAGPUR

Similar to the classes there are I will just discuss about ActionEvent classes those are there is basically handling any ActionEvent which may occur whenever a user clicks a button or a list item or menu item these are the things, so this is related to all this components.

Now, the class details I do not want to discuss elaborately in this I have mentioned the slides here I advise you to go through the slides read everything. And what I can say is that for every class that is there in related to the handling event there is a constructor, there is some constants involved and some methods involved and for this action event as we see here in this slide 3 constructors are there and this constructor is basically as you know the ActionEvent is automatically called.

So, there no need to call create any object explicitly; that means if you click a button automatically the action event object will be created and we have to just say exactly hold these objects for our further processing we will demo we will illustrate the concept clearly whenever we discuss some example.

(Refer Slide Time: 12:47)



So, for this action event these are the constructors, apart from this constructors there are only these constructors are there is no other methods and all these things for this an action event. Now, this is another AdjustmentEvent it is the event that is related to the scrollbar item whenever user negotiated the scrollbar then this event occurs and also this

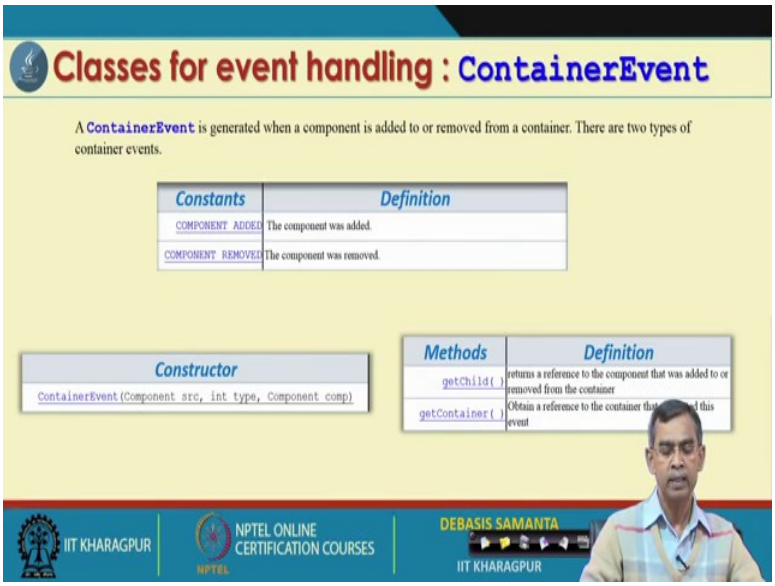
class deals with constants and then constructors only 1 constructor and 2 methods are there.

(Refer Slide Time: 13:12)

Now, ComponentEvent whenever some component is changed or component is resized component is configured and all these things is there and so component event is involved and these are the constructor there which is there in the ComponentEvent and then constants also few constants as we have mentioned COMPONENT HIDDEN, COMPONENT MOVED, COMPONENT RIZE, COMPONENT SHOWN.

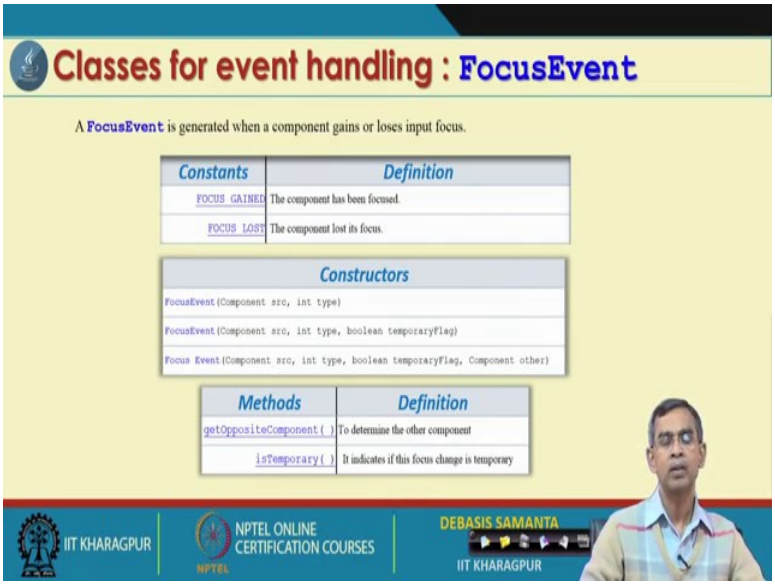
Here component means a frame like, so if you resize a frame it basically triggers an event if your component is closed there is an event occurs. So, corresponding to the event it is basically these are some activities that action can be listened by some listener objects like.

(Refer Slide Time: 13:51)



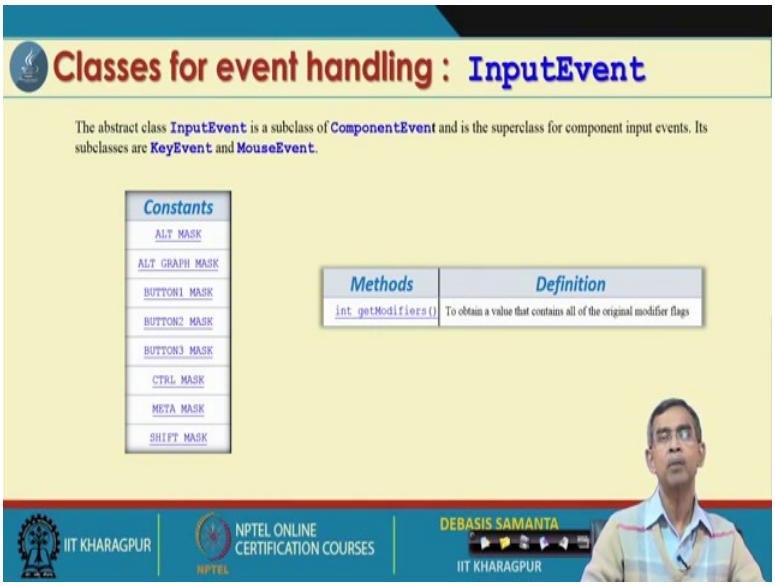
Now, so like this the container in an event is also there, a container is similar to the component. So, here either container can be added or container can be removed, whenever we close that event occurs, whenever we open that time it will occur if the like this.

(Refer Slide Time: 14:07)



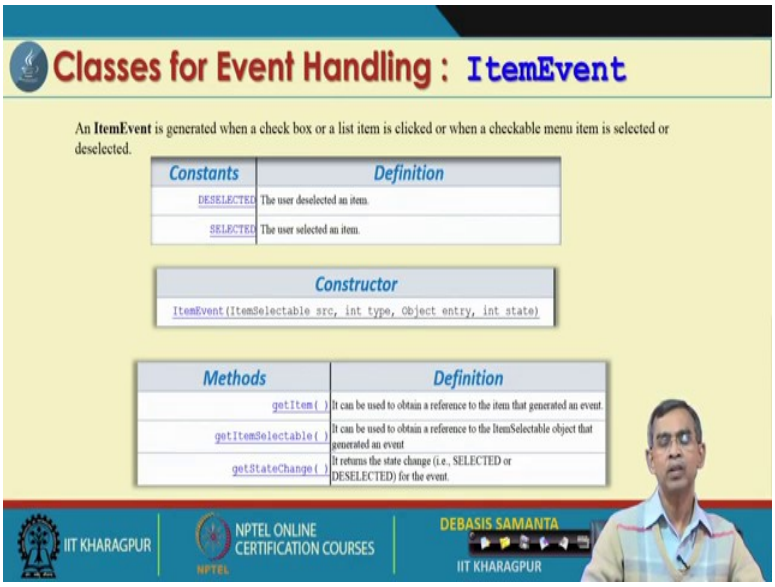
The FocusEvent is basically whenever there is a focus say keyboard needs to be focused like whenever some event that has to be listened to like this. So, FocusEvent is there, it has few values constant it is called and then constructors and methods.

(Refer Slide Time: 14:24)



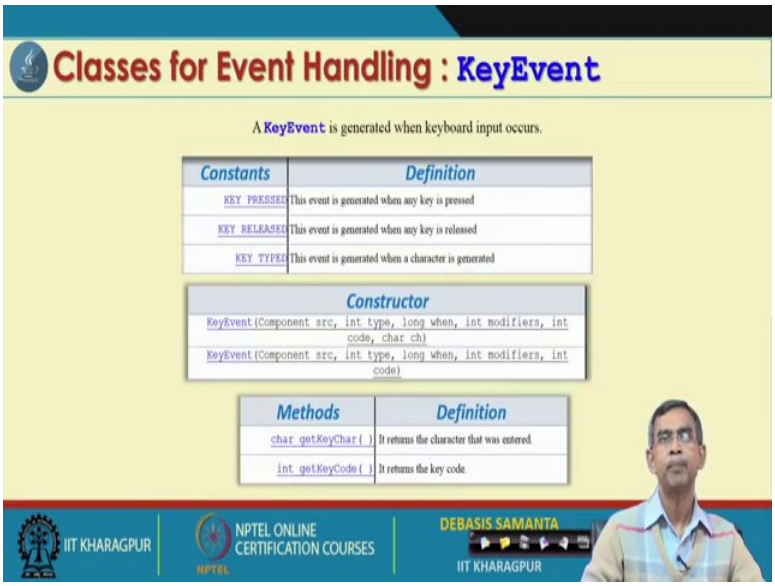
And then InputEvent is basically is a superclass, is a superclass of that KeyEvent and MousEevent which we are going to discuss shortly and this is the two most important event handling mechanism by which the key and mouse can be controlled.

(Refer Slide Time: 14:35)



An ItemEvent is also, ItemEvent is related to the menu items or list or checkbox and dealing with this and this is this class contains some constants constructors or method which has been mentioned in the slides.

(Refer Slide Time: 14:52)

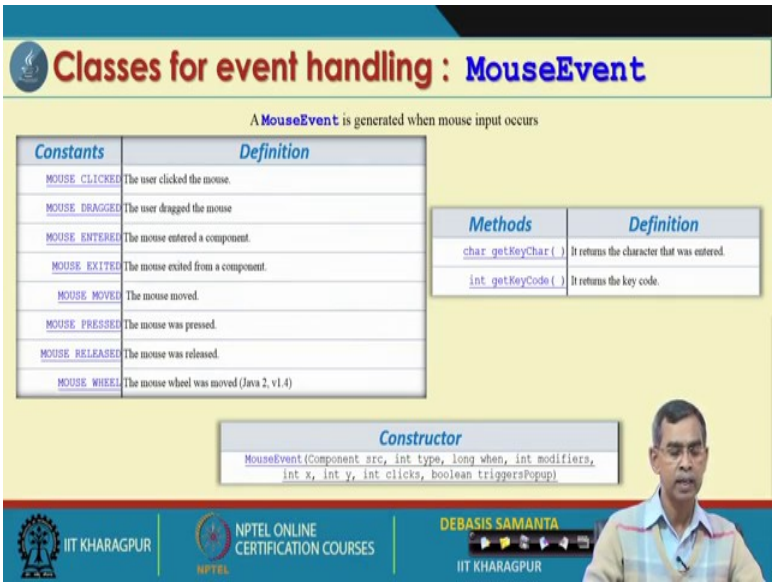


And then KeyEvent; KeyEvent means whenever there is a KEY PRESSED or there is a KEY RELEASED or just typing a key. So, these are the three things that may occur they are called modifiers. So, whenever I have the things is say per either KEY PRESSED or KEY RELEASED or KEY TYPED, this event is generated. And for this event there are two constructors as they see here the two constructors has the different parameters all those parameters has their own meaning, say components source means from which component it occurs because there may be more than one component in a program, it is (Refer Time: 15:29).

And integer type means what type of component, whether KEY PRESSED or KEY RELEASED or key typing long event means, it is at what time that event occurs and integer modifiers means what is the modifier mode actually an integer code; that means, what is the character that can be coded from this KeyTypes and a character key if it is a KeyType like this one.

So, these are the different what is called the parameters that are there in the constructor, what I want to say is that whenever one keypress is occurring all these values are basically pa initialized; that means, object is created means all these values actually and those values are very important to use in our program.

(Refer Slide Time: 16:05)

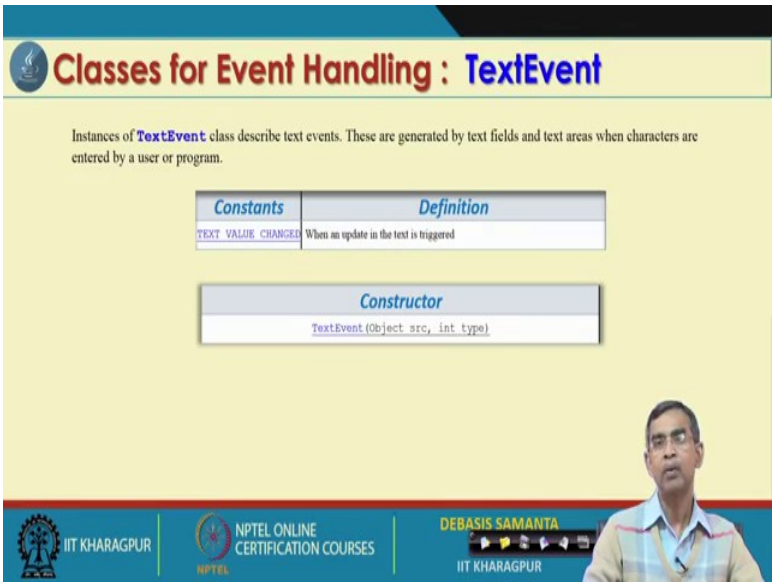


Now, MouseEvent like this KeyEvent, MouseEvent has new constants as we have mentioned in a MOUSE CLICK, MOUSE DRAGGED, MOUSE ENTERED, MOUSE EXITED, MOUSE MOVED, MOUSE RELEASE, MOUSE PRESSED MOUSE WHEEL; MOUSE WHEEL is for the three-button mouse actually.

Now, so, these are the things that may happen; that means, the out of many types. So, anyone of this things related to the mouse may occur and those things can be handled and there are two methods I get KeyChar and then get KeyCode these are the basics if you select one character, then which is the get key all these things can be selected and can be obtained. And it has only 1 constructor as the name of the constructor is MouseEvent same as the name of the class and the different parameters that is there in the constructor it is there.

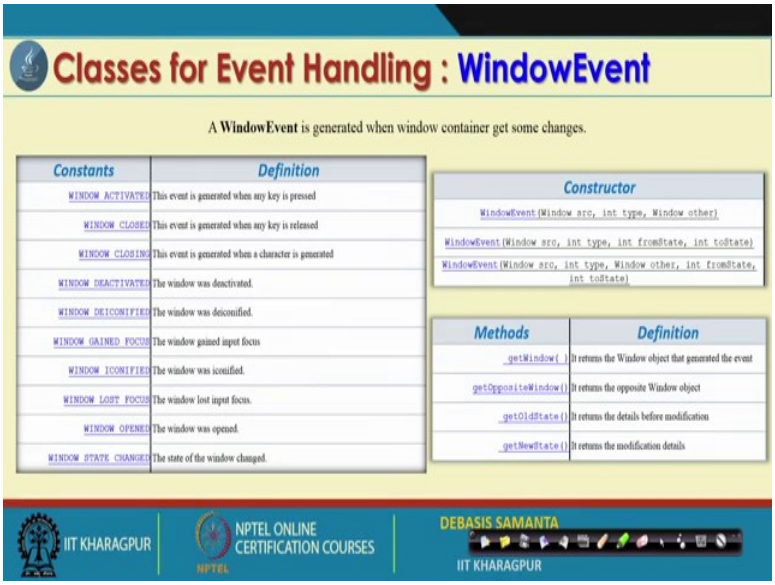
So, these are the constructor means these parameters can be generated automatically whenever an event occurs and then the value. For example, if you want to note that at what time mouse has been clicked I can get it from the long win. So, win value gives you that if there is a MOUSE CLICKS, a MOUSE PRESSED or MOUSE RELEASED then at the what is the time. So, these are the very important information that can be obtained intake seen to the eye; that means, in which location in the component the mouse has been clicked or released those kinds of things are there.

(Refer Slide Time: 17:39)



So, thus values are variations here whenever we have to process the event and TextEvent is related to the handling the text filled area. So, it has only one value TEXT VALUE CHANGED or not and then is it true or false like and it has only 1 constructor that the constructor is basically what is the source and what is the type of the things there?

(Refer Slide Time: 17:55)

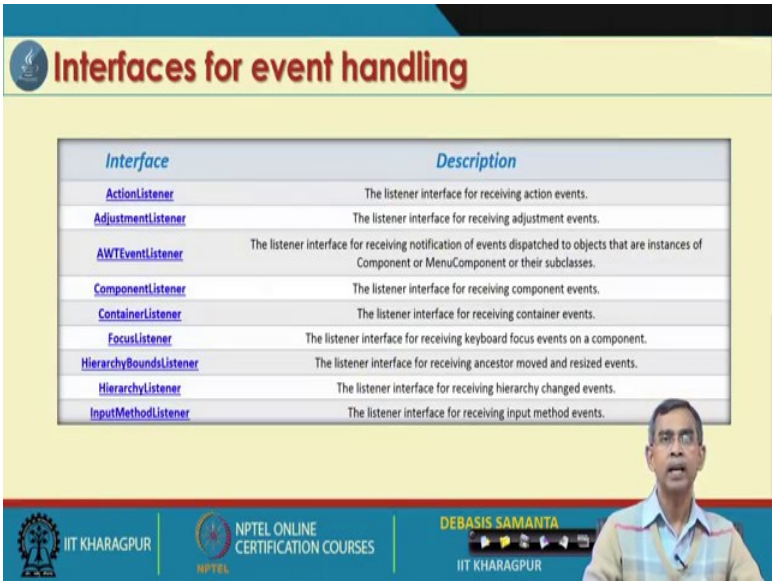


And this is a last one class that is there it is called the WindowEvent it has also few constructors and then constants methods all these things related to this one. Now, related to these classes so far the event handling is concerned whenever an event occurs the

event of any type of the class which I have discussed an object of that type will be automatically created that is all.

Now, this is so, for the event so event occurrence is concerned. Now there is a for every event there should be certain what is called the consequence now this consequence like event generation it is not automatic. So, a user has to mention that what is the consequence or what is the action that should be corresponding to an event. So, interface basically gives the programmer facility to define their actions for every what is called the events occurrence.

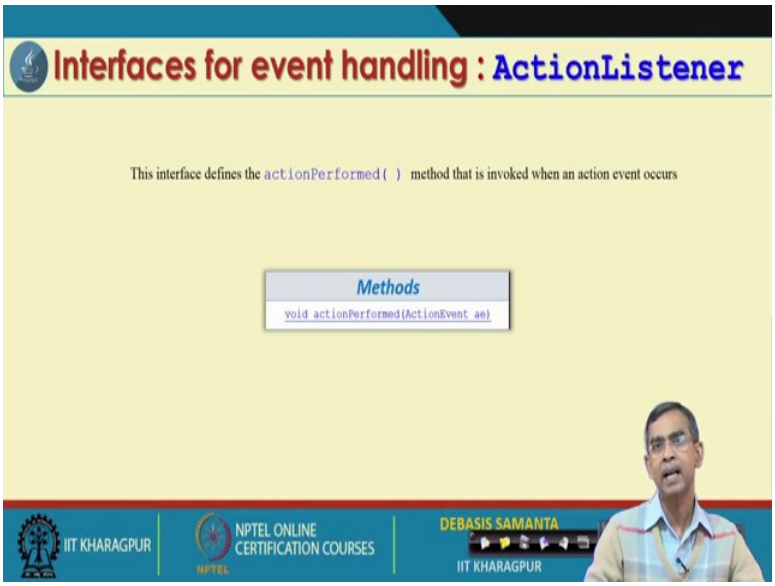
(Refer Slide Time: 18:54)



Now, here the different interfaces are there in those are defined already in the Java dot AWT package as you know interface is basically in there is method is there, but a methods are abstract. So, we have to implement the methods, we have to define the methods according to our own need; that means if this event occurs then what will be the action the consequence that you have to discuss.

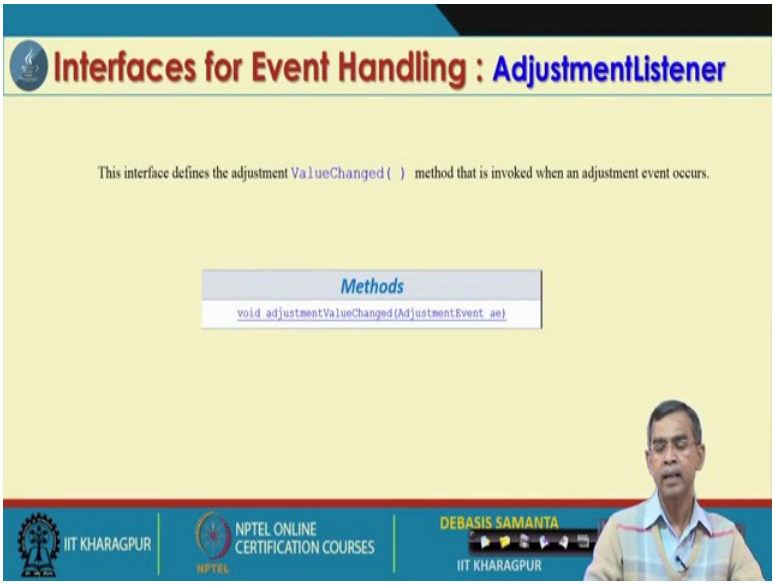
So, using this interface co object you will be able to create our methods provided that we know the name of the methods which are to be defined precisely corresponding to each event.

(Refer Slide Time: 19:36)



Now, like say action listener is an interface only the method it is there action performed ActionEvent. So, that is basically which action event that is the object automatically created we have to pass an argument to that and then action perform interface will be defined accordingly.

(Refer Slide Time: 19:53)



And then the adjustment listener is basically the method is there value change that needs to be defined by the programmer.

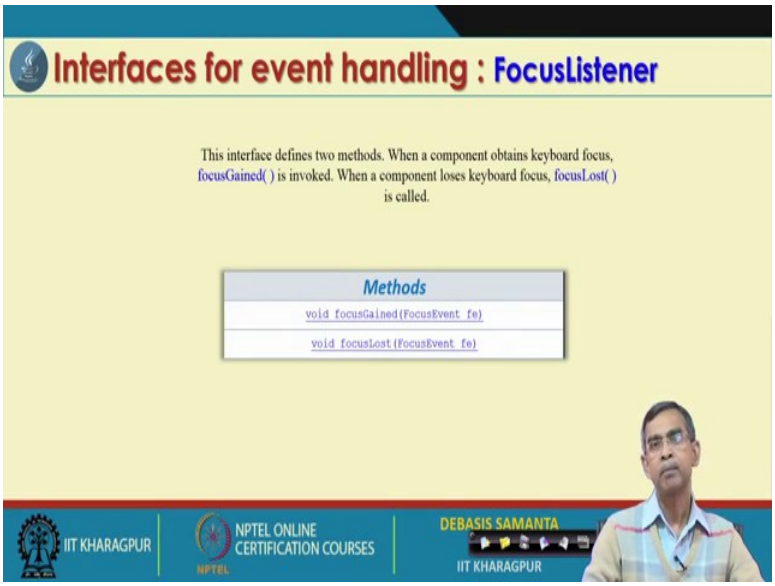
(Refer Slide Time: 19:59)

The component listener has many methods like ComponentResize, ComponentMoved, ComponentShown, ComponentHidden and those methods are need to be implemented.

(Refer Slide Time: 20:08)

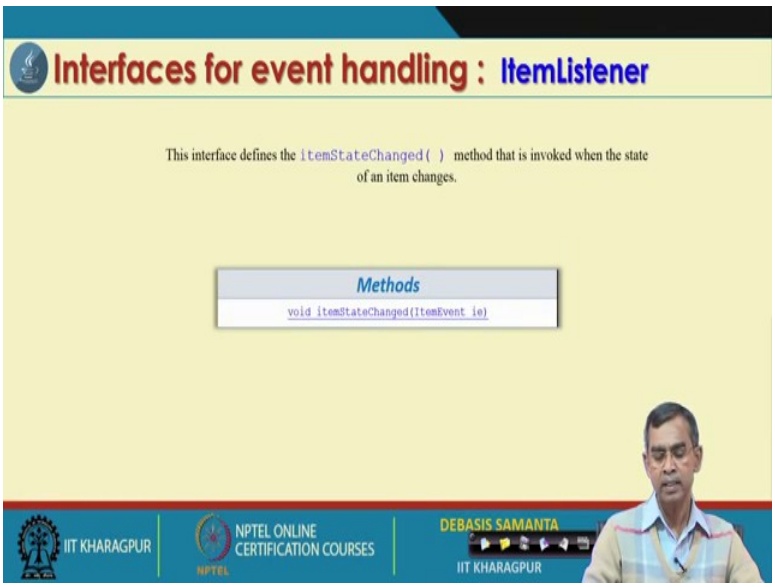
Similarly, container listener, it has only 2 methods ComponentAdded and ComponentRemoved, this has to be considered and then the object that has to be passed is basically container event.

(Refer Slide Time: 20:21)



And then focus listener the object that has to be passed as a focused it event; that means, this listener related to the FocusEvent.

(Refer Slide Time: 20:33)

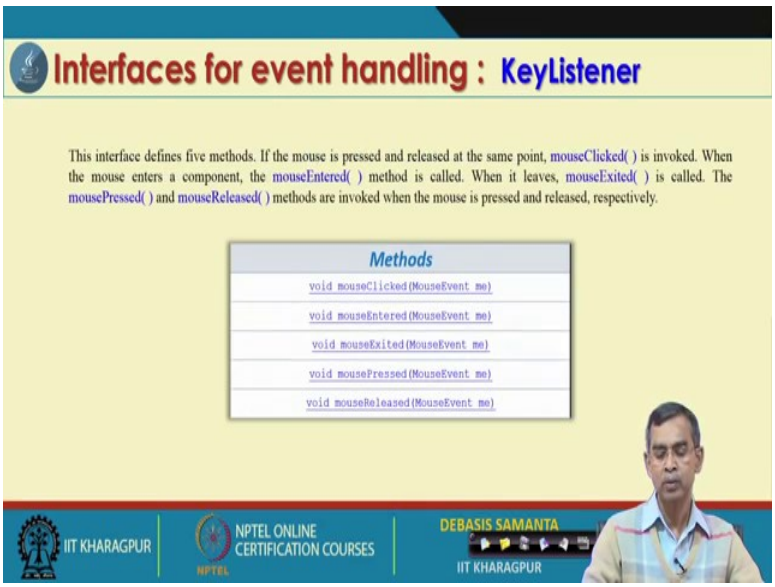


And then FocuGain or FocusLost is a method and then item listener is the focus change is the method needs to be mod overwritten.

(Refer Slide Time: 20:39)

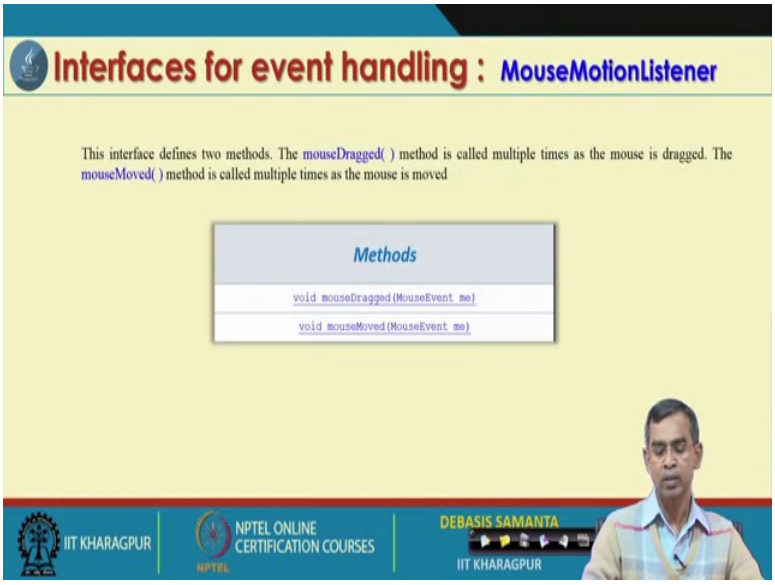
And then key listener is the keyPressed or keyReleased or keyTyped, these are the three I mean methods that user can modify in their program. So, there according to keyPressed what will be the action, according to the keyRelease what will be the action, if a key is typed then what will be the action like this one. So, the method can be customized by the programmer according to their needs.

(Refer Slide Time: 21:01)



KeyListener is another important or listener is there it has method mouseClicked, mouseEnter, mouseExited, mousePressed and mouseReleased those methods are need to be modified.

(Refer Slide Time: 21:13)



And mouseDragged and mouseMoved also are the two other interface method.

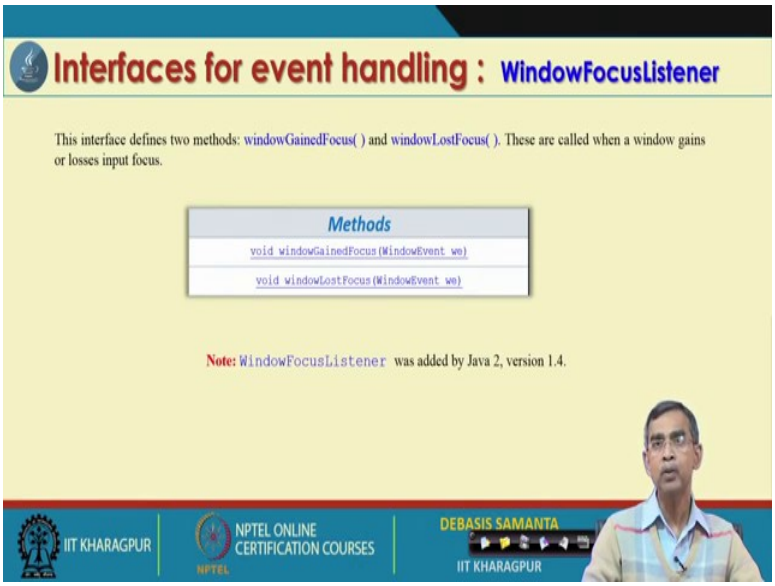
(Refer Slide Time: 21:6)

MouseWheel listener the 3 button mouse you know so the mouse will have the mouse wheel method is the listener here.

(Refer Slide Time: 21:25)

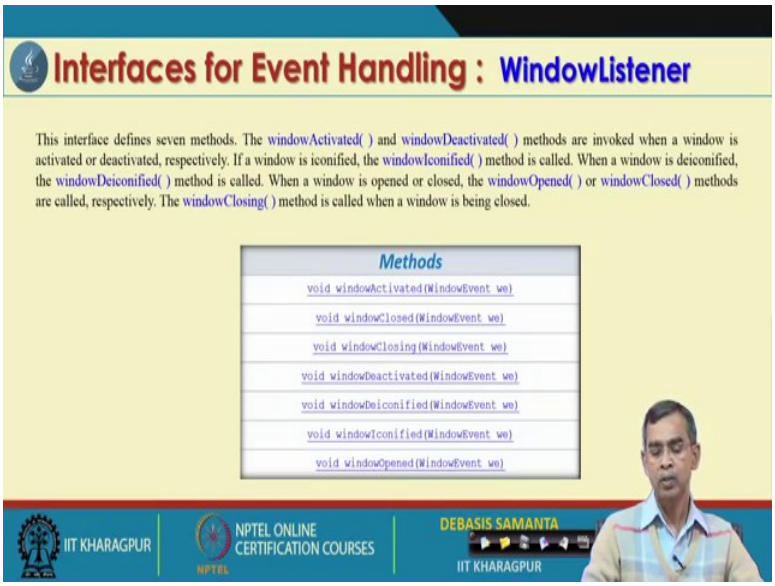
And text listener only one with a textChanged.

(Refer Slide Time: 28:28)



WindowFocusListener it has two method windowGainFocus, windowLostFocus.

(Refer Slide Time: 21:34)



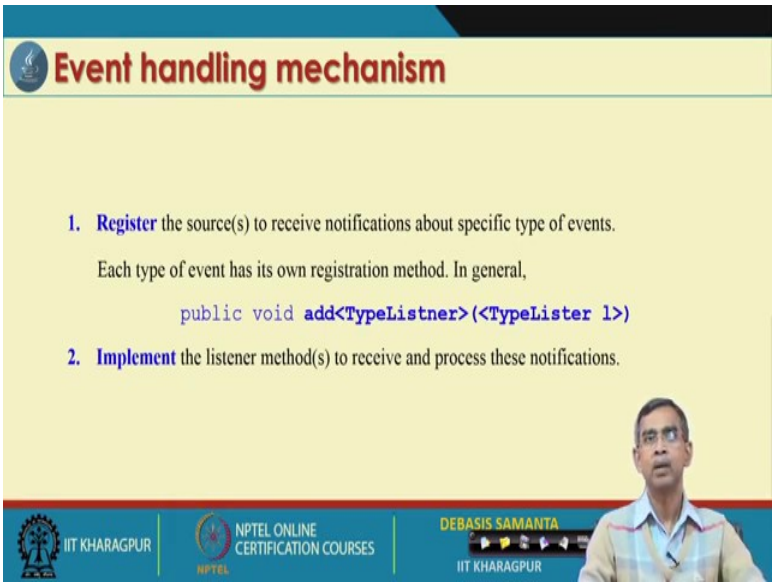
And then window listener is a many method as we have mentioned windowsActivated, windowClosed, windowClosing, windowDeactivated, windowDeiconified and windowIconified and windowOpened.

Now, these are the interfaces we have mentioned here and it is also not possible to discuss each event classes and corresponding to each event classes what are the action or listeners are possible because it is too time-consuming you are advised to refer to some materials, that is I have already mentioned in the first week of my lecture strikes where I have given very a number of sources you can console all the sources to have the full knowledge about it.

And that is this way you only you can learn much and much, but this is at the starting point of course. So, at the starting point you should have the full concept about what are the different classes are there to dealing with the inference, what are the different interfaces are there to generate the event consequence all these things are there.

How to the methods are everything that you can have in your repository at your disposal so that you can process then and accordingly the program can be written. So, I will give you some examples so that you can understand how the events can be handled using this one.

(Refer Slide Time: 22:48)



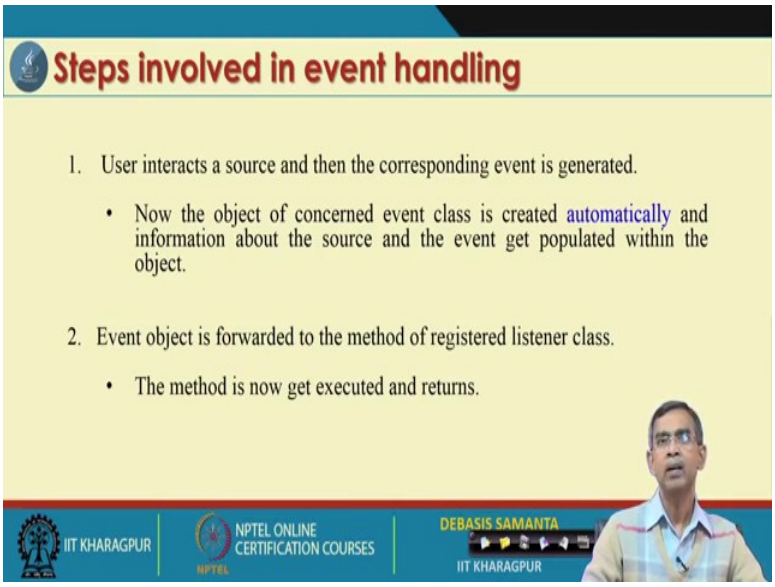
As the time is short I will not be able to cover all type of event source only mouse and keyboard are the two events will be concerned. Now, anyway so what is the basic idea if you have to write a program dealing with the event handling? So, there are only 2 steps these are the very simple 2 steps in the first step that is called the registration procedure, here basically we have to register an event listener.

So, here the idea about just simply in if you use the applet programming than in the init method you just add one method like public wait at type of listener whether it is an action listener is a mouse listener or keyboard listener, key listener or like this one or mouse motion listener whatever it is there. So, public void adds say key listener; that means, we are adding to key listener and what is the type of the object that it can handle, for example, action event for the key mouse event for the mouse these kind of things are there so it is like this.

And then implement the listener method, so, if it is supposed action event then the method that you have to implement with the key pressed, key down key released whatever it is related to this one. Similarly, if it is an add mouse listener is the listener interface; that means, and then mouse event is the event occurs, then we have to implement the method like say mouse pressed, mouse drag, mouse release, mouse move whatever it is there.

So; that means, what will happen if there is a mouse pressed, mouse clicked and accordingly we have to write the listener method, so that it happens then what will be the actions that I mean event corresponding to the event, what will be the action are to be performed. So, that is not, so these are two things are there so registration and then implementation. So, better some illustration is required, so that you can understand these two processes.

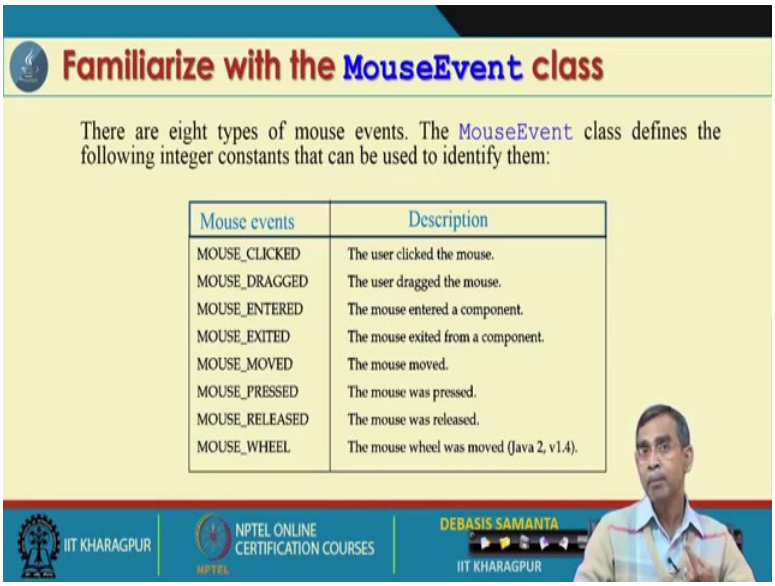
(Refer Slide Time: 24:44)



Anyway, so as you have just started summarizing this for fact is that whenever an event occurs; that means, a user interacts any source like saying button or any list or items whatever it is there automatically to this there will automatically an event will be created. And this event will be created corresponding to this event there are many other information since automatically stored in the objects and then event object is forwarded to the listener method which is already registered in the program.

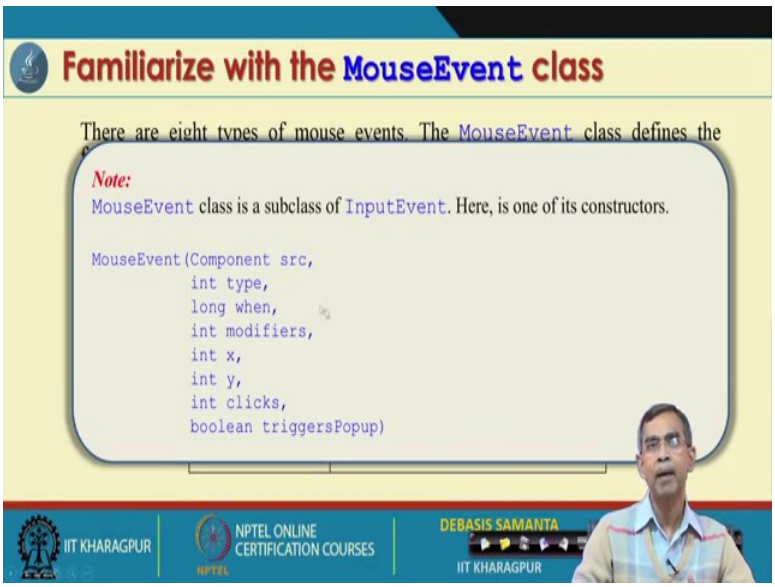
So, you can register as many as listener as it is possible or relevant so your program is concerned. So, we just have to do it and here is let us have some example so that you can understand the first I will discuss about handling mouse events.

(Refer Slide Time: 25:34)



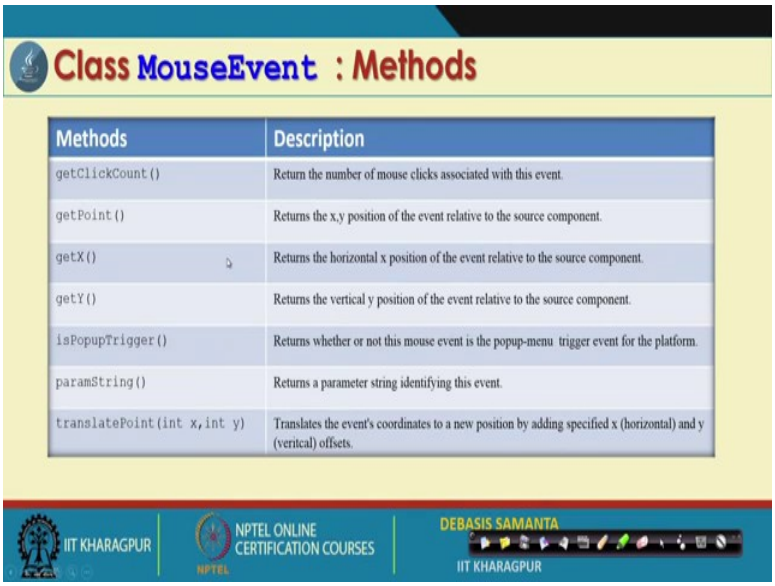
And so for the MouseEvent is concerned as we have already mentioned these are the mouseEvents that may happen MOUSE CLICK, MOUSE DRAG, MOUSE ENTER like and these are the basic field value that can be; that means, if every mouse event have occurred then these are the on value so the can be read by the program. So, that program programmer can understand which it occurs because we have to do it automatically.

(Refer Slide Time: 25:59)



And then there is a constructor which basically, automatically the MouseEvent object will be created, by these problems these are the different value that we can process it.

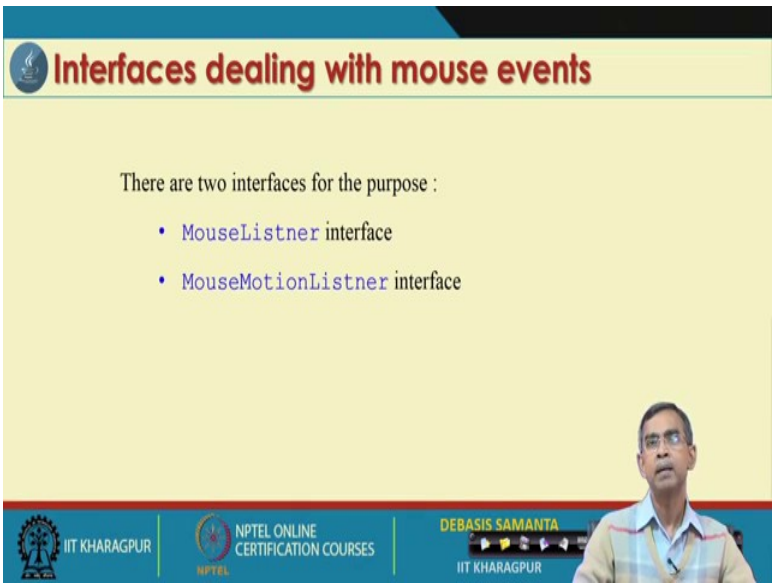
(Refer Slide Time: 26:08)



Methods	Description
getClickCount()	Return the number of mouse clicks associated with this event.
getPoint()	Returns the x,y position of the event relative to the source component.
getX()	Returns the horizontal x position of the event relative to the source component.
getY()	Returns the vertical y position of the event relative to the source component.
isPopupTrigger()	Returns whether or not this mouse event is the popup-menu trigger event for the platform.
paramString()	Returns a parameter string identifying this event.
translatePoint(int x,int y)	Translates the event's coordinates to a new position by adding specified x (horizontal) and y (veritcal) offsets.

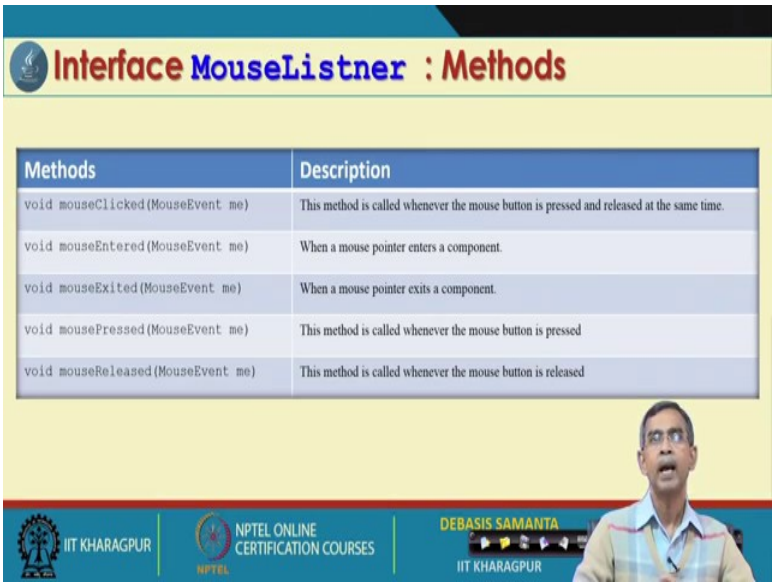
And then the methods that are possible so get x get y; that means, in which location a mouse click or char that can be obtained like this one.

(Refer Slide Time: 26:16)



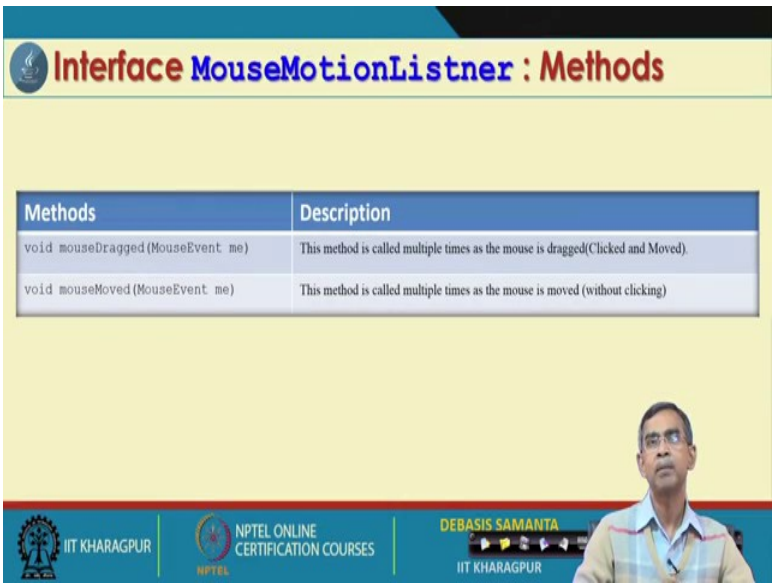
Now, there are two I mean interface as you have already learned about MouseListener and MouseMotionListener interface. So that means, if it is a mouse click mouse pressed, mouse released, a mouse listener if the mouse dragging is involved then mouse motion listener interface are to be registered in your program.

(Refer Slide Time: 26:34)



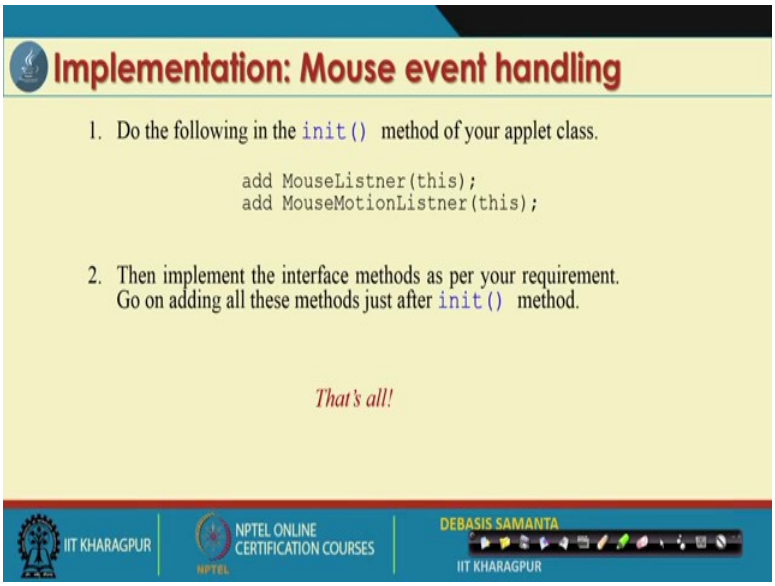
And so for the mouse listener interface is concerned these are the method already you have mentioned.

(Refer Slide Time: 26:39)



And for the mouse motion listener these are the two methods already there this means we have to implement it.

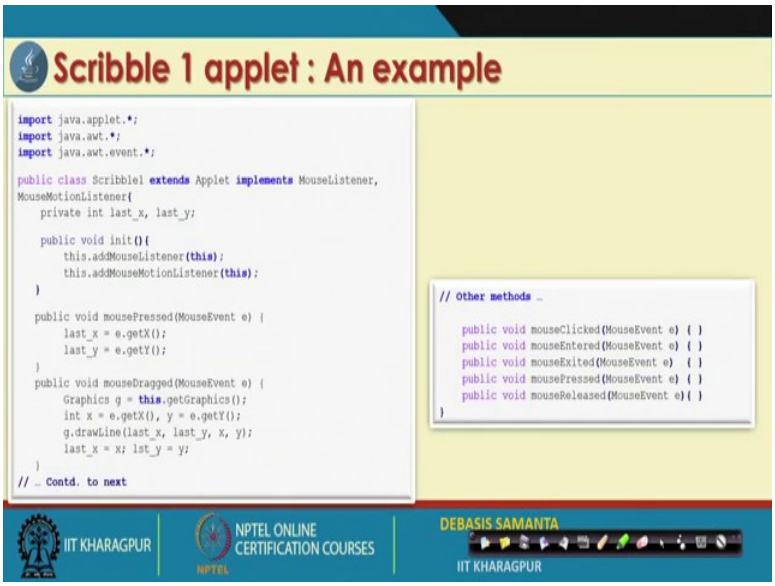
(Refer Slide Time: 26:41)



Now, so for the dealing with a mouse event handling, so first what we have to do is that either in init method or any beginning of any class declaration that is way to event handling program we have to add this one. So, add MouseListener add MouseMotionListener and then this is basically for the current object; current object actually.

So, this is the public wide at mouse listener, there in now should not be any blank space actually if is so blank space should be avoided and then the implement the interface method as far the requirement of the programmer and that can be done immediately after the init method so, that is the two procedures is there.

(Refer Slide Time: 27:24)



Now, let us have the examples for these things, as we see this is a small example that we are going to discuss this is a scribble one applet and as you know the scribble applet basically allows the user to I mean draw something on the some component or container on a frame or maybe on applet. So, in this example as we see we are declaring one class call the Scribble 1 extends Applet. So, it is a basically Applet so Applet will be there and then it implements MouseListener and then MouseMotionListener. So, is two interfaces are involved in this content, so this is a customized thing that you have to do it.

And these are the x and y are the two variables that needs to be maintained so we have discussed it here and now let us see the public init method in this init method we have registered the two interface the two listener actually here the add MouseListener and addMouseMotionListener objects are to be registered.

So, this basically inputs or registration there and the this registration followed by the declaration of the interface methods namely mousePressed and then mouseDragged the two methods we have implemented here. So, it is very simple it will basically get the current location of X value and the current location of Y value and then it will store into last x and last y, so that is all.

So, whenever mouse pressed, is there it will just read what is the value of x and y it is there and you know that e is the event. So, that event means whenever this an Applets

will occur so it will corresponding to this event that event e will be generated. So, that this will basically be written this object event is there.

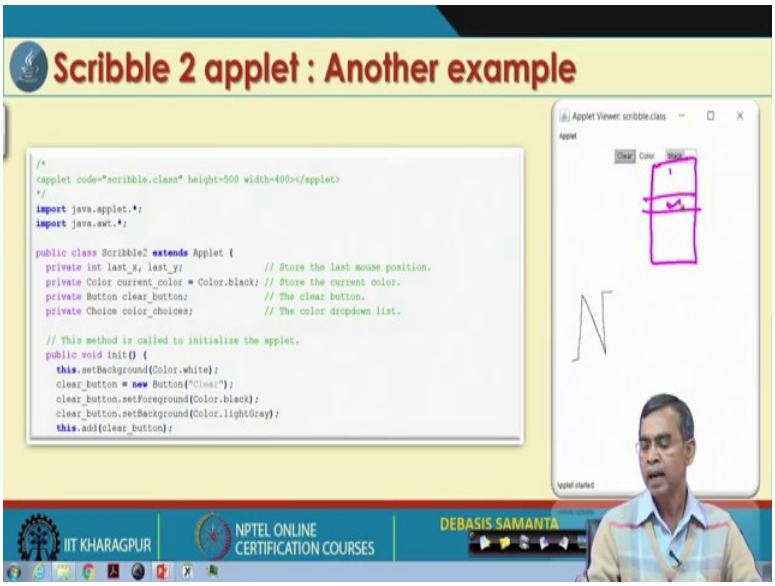
Now, so or here basically mousePressed, MouseEvent e as we see them whenever mouse event occurs, so an event e is created and this is basically for that event the x and y location is the coordinates of the x and y position in the area. And then mouseDragged it is basically what we have done is so basically if this is the last position and this is the current position so dragged means draw a line.

So, if the mouse is moved here and here so it draws a line automatically, so here line will be drawn. So, if the mouse move and like this then automatically the line will be drawn by the system and this is a routine it is here we have mentioned here. So, it basically so just a graphic subject is created; that means, it is basically related to the drawing a graphics when its line drawing will be there and that is why g dot line we have then and for the g draw line we see last x last y and x y.

So, it is basically co what is the previous x y and that is the current x y draw line. So, previous x y current x y like this, so this way you just dragging and as the dragging continues it will draw the line whether the straight line or any curvy line whatever it is there and so these are the methods so for the mousePressed and then mouseDragged is concerned. Now there are few other methods also so for the MouseEvent a there is concerned like mouseClick, mouseEnter, mousExited, mousePressed; mousePressed and mouseReleased mousePressed is there already we have declared so it is no more required

And so these are the other methods those are remaining, now here I just do not have any code for any method I to remain the wide method like. So, otherwise you can just simply ignore all these the only two methods are involved, then you can define the two methods and this completes the scribble applets and if you run the scribble applet you can see it will to give the output and that will be discussed whenever we have the complete demonstration.

(Refer Slide Time: 31:06)

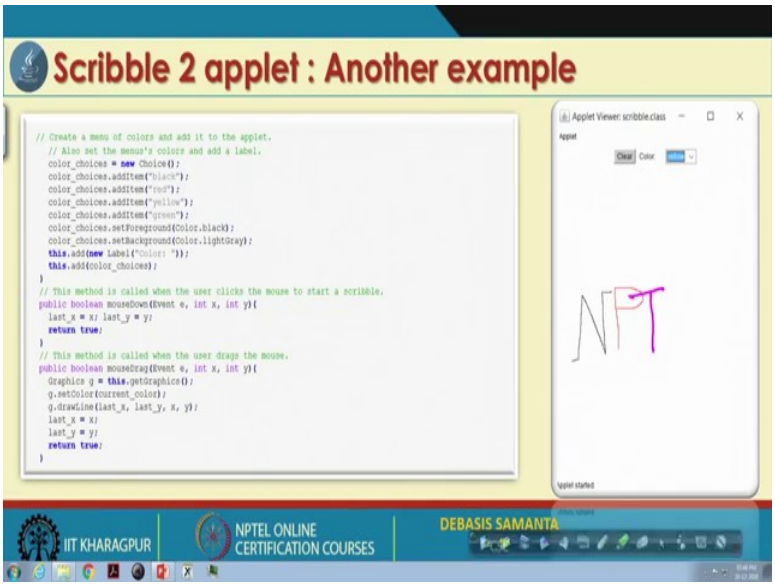


Now, another example let us see the scribble 2 applets it is little bit earlier one it is basically using the latest Java JDK version so and this is the one example which is basically oral one using the Java 1.0 version, but it is pretty useful and then very simple little bit different methods and classes are there I quickly mention about. So, here basically as we see in this example we have to create one-button call the clear and also we have to create one checkbox the checkbox will if you click it the checkbox will give you the number of colors that are there.

So, user it can clear it then basically it will whatever the user has typed it will be cleaned from here and whenever user click this one so a list item will be generated from this list item which is if the user can select any item corresponds this is an item the color will be selected. So, here as we see the 2 components are created one is called the Button and then another is called the current color choice, so these are the two components are created here.

Button this is for the button and this is the choice this is choice basically are this one so these two buttons are created. Now in the init method as we see we have just simply set background all these things we have already known about it here and these are the variable that is declared they have been initialized that one and then all the buttons and then choices those are declared they have been added into the same it is basically this is applet so it is added to the applet.

(Refer Slide Time: 32:46)



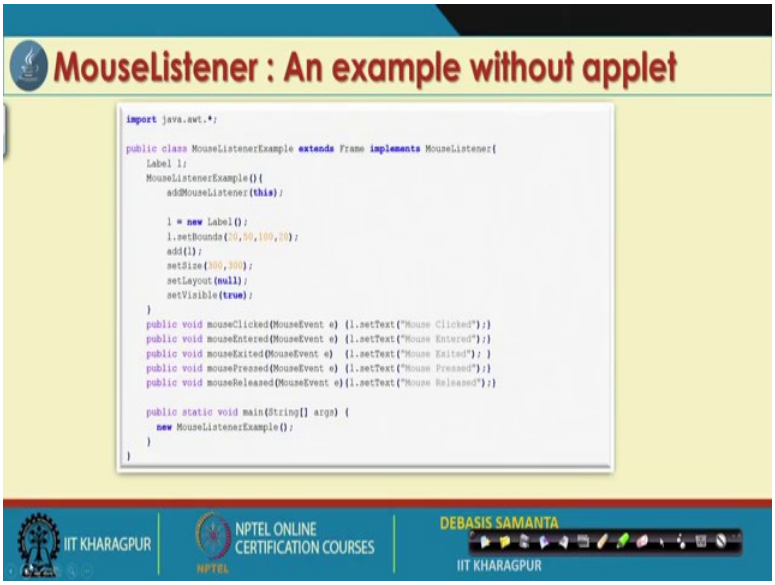
And then finally, we just create the choices so these basically quote to create the choice component it has the so many elements in it and then finally, we add these choices into the container, so this is the method that we have declared here. Now, here see this is basically the method that we have to overwrite this method is defined in Java dot AWT MouseEvent declaration and there Event e, int x, int y is basically showing there what is the event; that means, if it is a mouse event and x and y means in which location the mouse event occurs.

So, this is basically the method as per the old version of the Java 1.0 package and then it is basically same as the earlier one. So, this basically define mouse down; that means, if a mouse if fixed and this is a mouseDragged, it is similar the mouseDragged listener method actually that we have already discussed in the last example Event x and y is a same thing. So it is there setColor basically if the color is chosen by this. So, the current color set color according to this color it will draw; that means, here as we see user can set a color and according to this color if it is drawn then it will draw like.

So, if it is a pink color and then it is pink color will be drawn and so on. So, there is a very nice applet is very simple only few lines quotes are there and that is why this AWT program is called a lightweight process you see you can find many things which are automatically done by you only you have to just play and plug that is all and which is basically not possible in any other programming language actually that you can do it

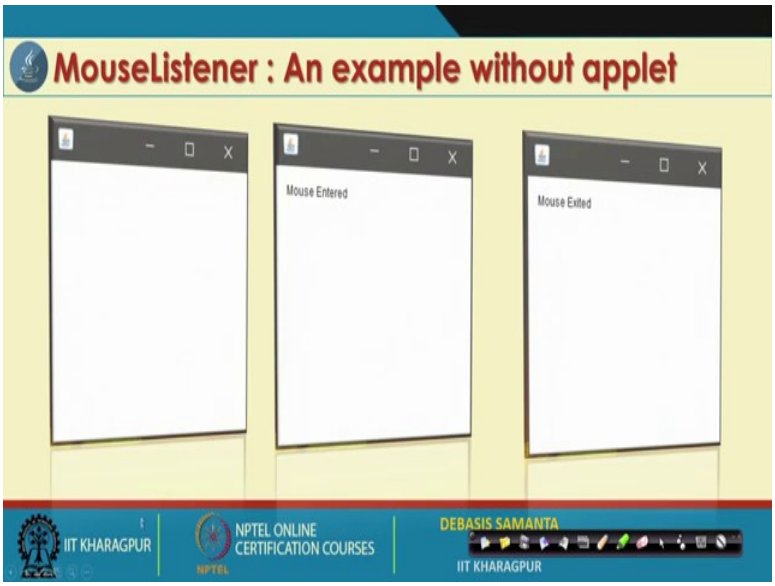
from here. So, this is the idea about; this is the idea about the two examples, so mouse event handling is concerned.

(Refer Slide Time: 34:35)



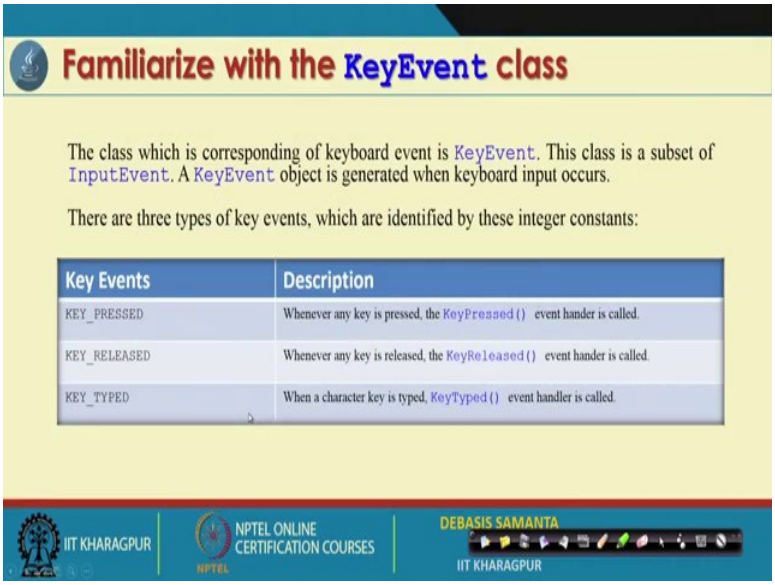
Now, I will discuss about another example this example is MouseListener is very same similar example I will just quickly have it. So, add MouseListener is basically MouseListener in this case, this is basically creating the graphics for us; that means, this is a frame where it is basically create a frame in this example and here these are the method that we have overwritten it is very simple MouseEvent method, mouseClick it is just simply mouseClick and like that.

(Refer Slide Time: 35:07)



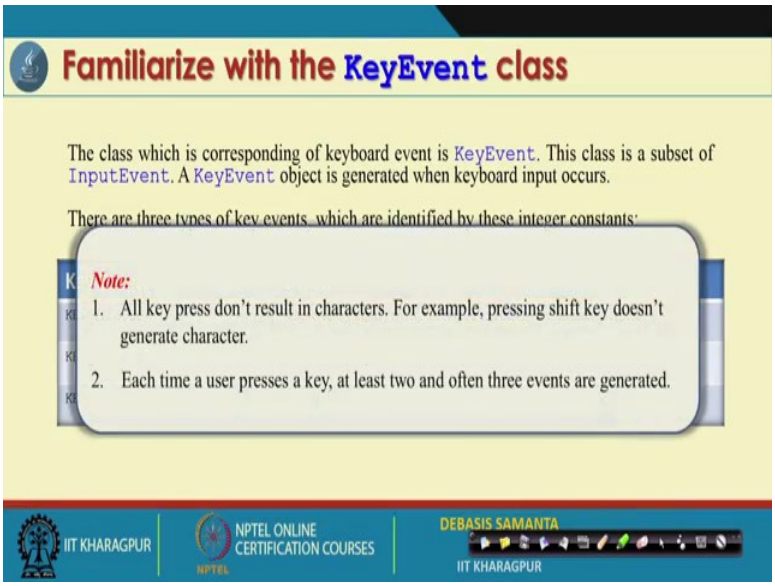
So, this way it will just whenever some clicks or occurrence click is occurred so it will basically response through that and then object will be created like this kind of output you can get it. Now, handling the keyboard event this is then let us see how the keyboard can be handled.

(Refer Slide Time: 35:17)

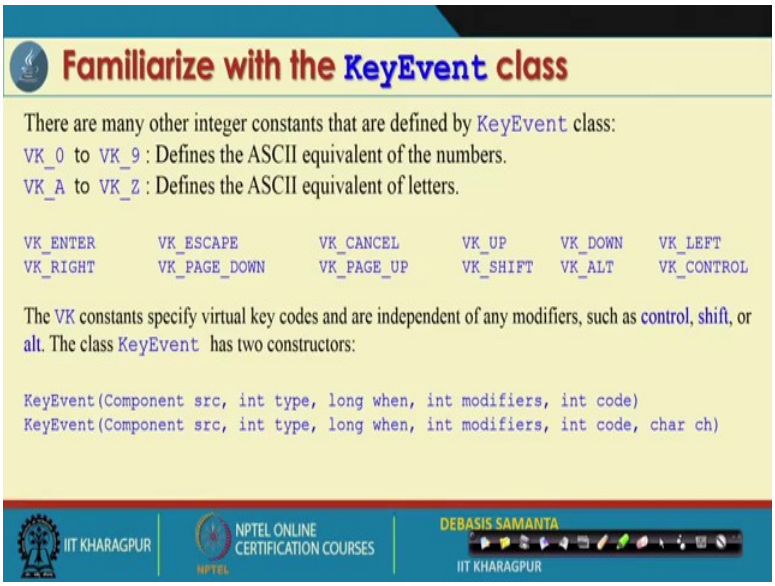


Here as we have already discussed it is basically whenever keyboard event occurs there is a key event object will be created and this object has the three different values are there means keyPressed, keyReleased and keyTyped.

(Refer Slide Time: 35:30)



(Refer Slide Time: 35:32)



And then it also has the many other this is also there so for the different keys are concerned because in your keyboard whatever the qwerty keyboard that we use it in general, it has many function keys alter control c then modifier and then some are non functional keys and then character keys numeric keys all these things are there.

(Refer Slide Time: 35:54)

Class KeyEvent : methods

Methods	Description
`char getChar()`	Returns character that was entered, for invalid character, it returns CHAR_UNDEFINED
`int getKey()`	Returns the ASCII code of the enterd key.

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

So, according to these things it basically deals with the different values that it can need from the keyboard and it has the two methods the get Char and then get Key, basically what is the character that is in the thing and get Key is basically written ASCII values the right.

(Refer Slide Time: 36:04)

Implementation : Key event handling

To handle key events, the same steps as in handling mouse events to be followed.

1. Register the following listener in the `init()` method of your applet class.

```
add KeyListner(this);
```

In addition to this, `init()` must include request input focus.

```
requestFocus(); // This method is defined in the component class. If not,
                then the program will not receive any Key events.
```

2. Implement the listener methods:

```
KeyPressed(KeyEvent ke)
KeyReleased(KeyEvent ke)
KeyTyped(KeyEvent ke)
```

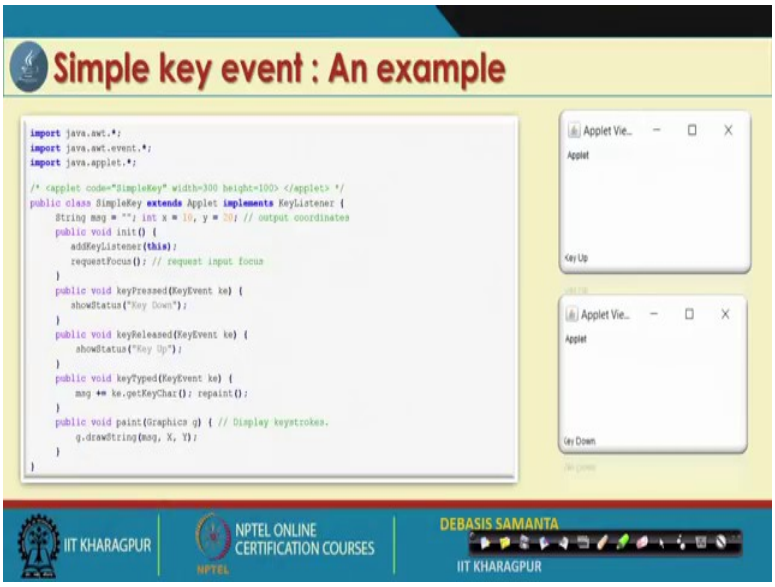
That's all!

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

And so for the listener, it concerned it has only one listener keyListener method is there. So, add a key listener to the accomplished any unique method or any at the beginning of any program that you have to implement and then in addition to this add keyListener that

you have to use also requires focus method to be included here. So, these are the two things that need to be added first and then once it is there this is a registration procedure, then rest for rest of the part is implementing the three methods whichever relevant to your program execution the keyPressed, keyReleased and keyTyped.

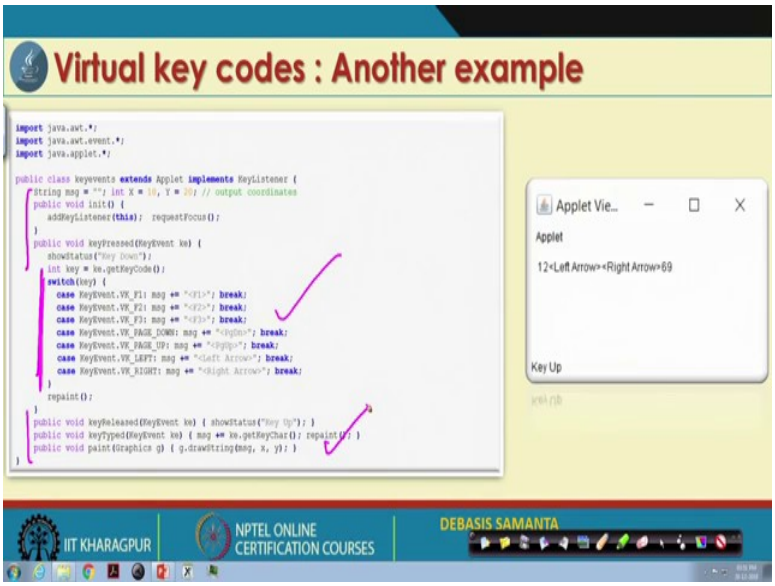
(Refer Slide Time: 36:39)



Now, I will discuss about a small example by which you can understand about my methods here. Now in this example as we see here these are the registration process and these are the basically interface implementation related to the keypress event as we see whenever key event occurs key is automatically generated we can read the values of this key objects whatever the according to the constructor that we have already defined and then we will be able to handle it.

So, if this program is run it will basically all the times it will prompt that what are the key events it occurs in your system.

(Refer Slide Time: 37:13)



And then this is the another example and then if this example is a similar to that one also it has also like registration method as the previous one, but here are different function key that it can be because the different function it can identify and accordingly it can bring it and other keyEvent also keyPressed, keyReleased, keyTyped for the sake of simplicity I have mentioned here. So, that you can write the code of your own according to this event and then you can get the result here.

So this is basically the just a starting point we have discussed because event handling is not a simple task to understand so easily, but as the time is short, so we do not have any other things to discuss here. But in our next demonstration when we will discuss it a lot of many things also will be demonstrated to you so that you can understand about it more in a detailed manner.

Thank you very much.