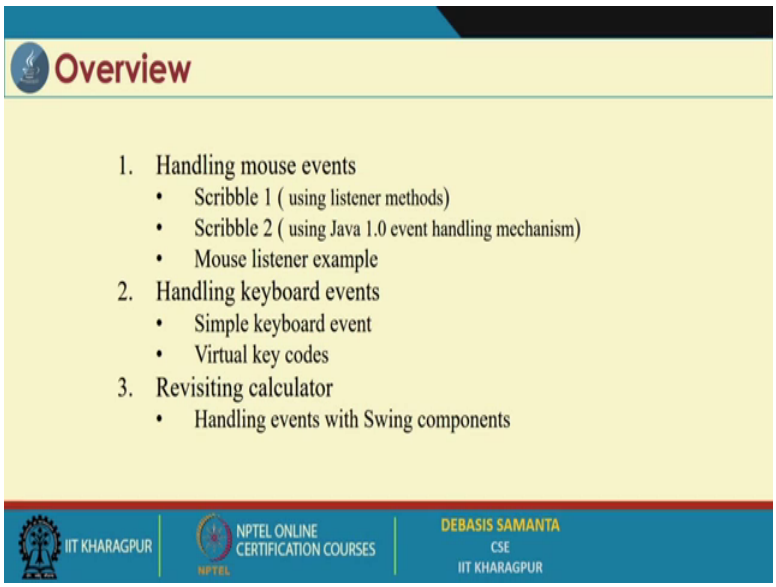
**Programming in Java**
**Prof. Debasis Samanta**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kharagpur**

**Lecture – 45**
**Demonstration –XVI**

So, I welcome you to this Demonstration session. This demonstration is based on the topic on event handling demo event handling mechanism in Java, which we have covered in the last module under it AWT programming part 3.

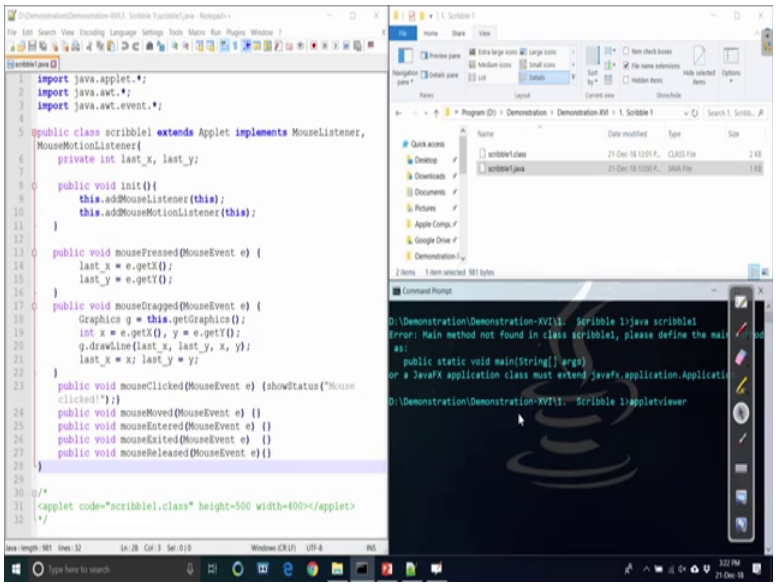
(Refer Slide Time: 00:35)




Now, we know exactly what is an event handling concept it is there and in our theoretical class also we have discussed about few examples like Scribble 1, the Scribble 2 and then the MouseListener example and also we have demo you have explained how the keyboard events can be handled and then finally, we have given an idea about the calculator.

So, in this demonstration we will again repeat the same thing, but in detail manner so, that you can understand how the event handling activities are carried out.

(Refer Slide Time: 01:13)




So, let us have the first demonstration this is regarding the design of a scribble applet. As you know scribble applet basically gives user draw something on the applet or I mean on the compone container itself either on the frame or it is on the applet. Now this scribble applet scribble example that we are going to discuss is basically applied version.

So, first we want to extend this program as an applet class and then we add the event list event and we have to register the event handler routine namely, in this example basically related to the mouse event handling. So, we have to add or you have to register two listener namely MouseListener and MouseMotionListener. And then we have to define implement the event handling method. So, in this case the method which are very importance are the mouse based method and then MouseDragged method.

Now, these two methods are defining the interface MouseListener and the mouse MotionListener interface. So, you have to implement it here. Now we will implement there are some other methods also those are there in these interfaces namely mouseClicked mouseMoved mousePressed mouseExited and then mouseReleased oh sorry mousePressed is already you have implemented. So, mouseClick mouseMove mouseEnter mouseExited mouseReleased all these things. So, those interfaces can be also implemented we will just give an idea how the impli such a in interface method can be implemented.

Now, let us see the program here as you have displayed here, and this program as we see this is the init method this init method is very important in case of applet the init method is coming to the picture in case of other AWT based programming they are new there is no init method, but in the class where we are implementing in that class itself we can add. So, we just add I this is a part of the registration as you know the first step of inner handling.

So, we just add MouseListener and then add MouseMotionListener this is the syntax that you have to follow here this means is basically for the current threat and here we just define the mousePressed e that that is the even handling mechanism here. So, mousePressed and is the mouse if e means event as you have already mentioned that, whenever an event occur related to the mouse event or whatever it is there the automatically the AWT will generate an event object.

So, in this case if a user pressed a mouse button then automatically the event e will be generated. So, if this is the event then from this event we can extract many information for example, for this event dot getX it basically tell what is the x coordinates of the mouseClick position similarly e dot get y this y coordinates of the mouseClick position. So, these two positions we easily store because we have to draw the scribble; that means, whenever user draw by dragging the mouse, it will be displayed on the area of the applet.

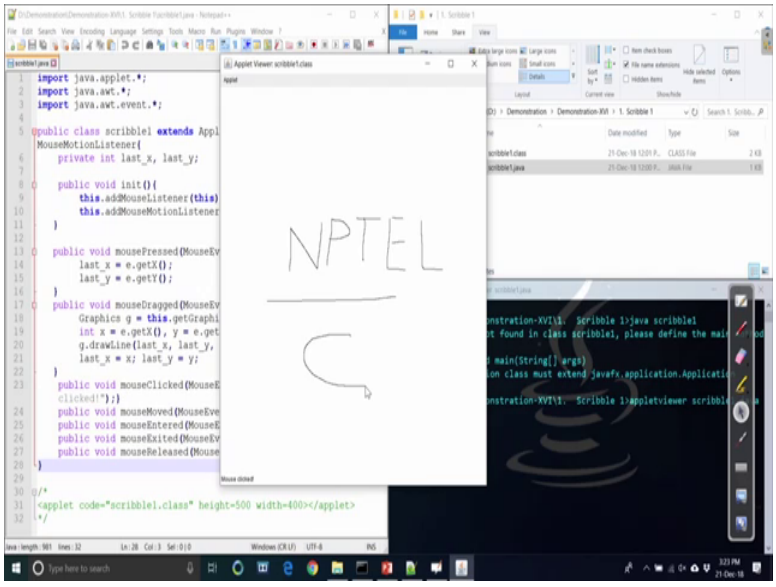
Now, so, this is the mouse press it is very simple in this mouse press even we have to just record what is the current location where the mouse has been paste and then another method that we are we have implemented here mouseDragged. That means, if we press it and then just move the mouse then it is basically called the mouseDragEvent and in this event what we are doing is basically we draw a straight line from the last position to the current position and incrementally if all the positions are like this drawn then a state a line according to the point or tracking of the mouse will be displayed on the screen.

So, this is the two method that we have implemented here and the other methods namely MouseClicked mouse mouseMoved mouseEntered mouseExited and then mouseReleased can also be defined. In this case we have not defined anything explicitly they are implicit things are there; that means, if any event occur related to these activities there will be no action taken actually because as you have not done anything here or

even if it is done also, no event can be caught and as a result no action will be initiated ok. So, this is the example the scribble one.

Now, let us see the output and then we will come back to the example again; we will do certain modifications so, that you can understand impact of that modification. I am just a little bit increasing the size of the applet so, that we can view little yeah.

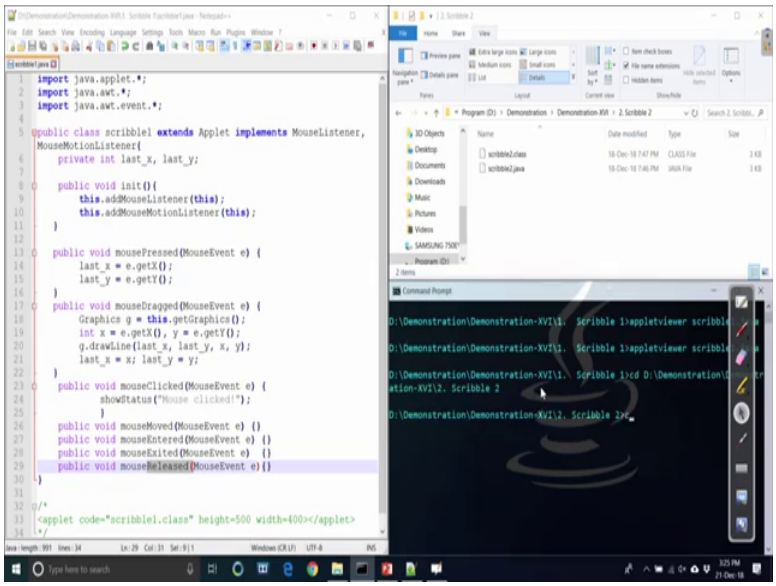
(Refer Slide Time: 06:27)




So, this is basically the scribble applet and at the moment we do not have any event, now I am just going to drag the mouse and as we drag it as you see line will be drawn like and drag not initially any order you can draw here whatever it is there yeah. So, you can see using the scribble applet, we are able to I mean by means of mouseDrag event handling we are able to draw any things there any curvy line also we can draw and we can take it whatever it is there that is all.

So, this is the idea about the scribble 1 version 1 we can say, now I am coming back to the problem again and here is the one event say suppose mouse clicked.

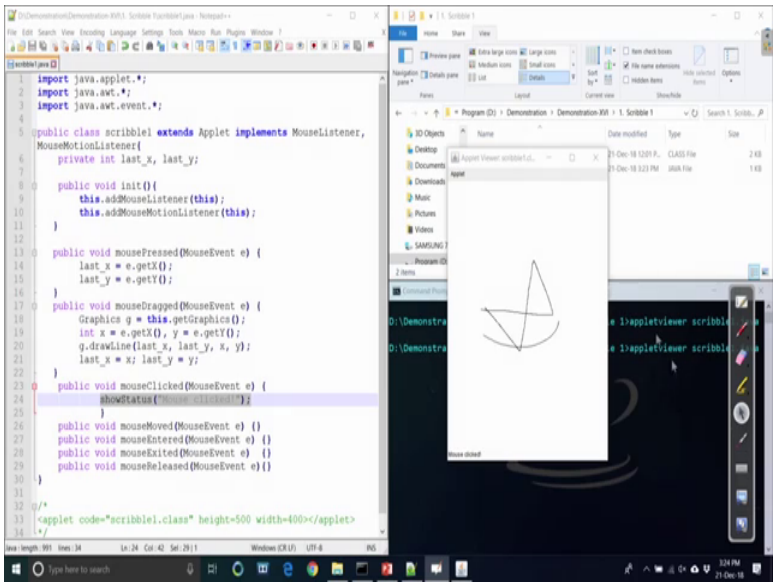
(Refer Slide Time: 07:19)




And as there is no action it has been implemented. So, it is not doing anything in the same applied also if we click nothing will takes place because there is no event handling routine is there. Now I am just adding in this mouse event handling routine a sa simple quote like showStatus as you know showStatus method will basically pin something whatever it is happened here.

Now, in the mouseClick option I just use the. showStatus method and then we just message mouseClicked. So, what will happen is, if we just click it and automatically in the status of the plate viewer it will basically display that mouse has been clicked.

(Refer Slide Time: 07:59)

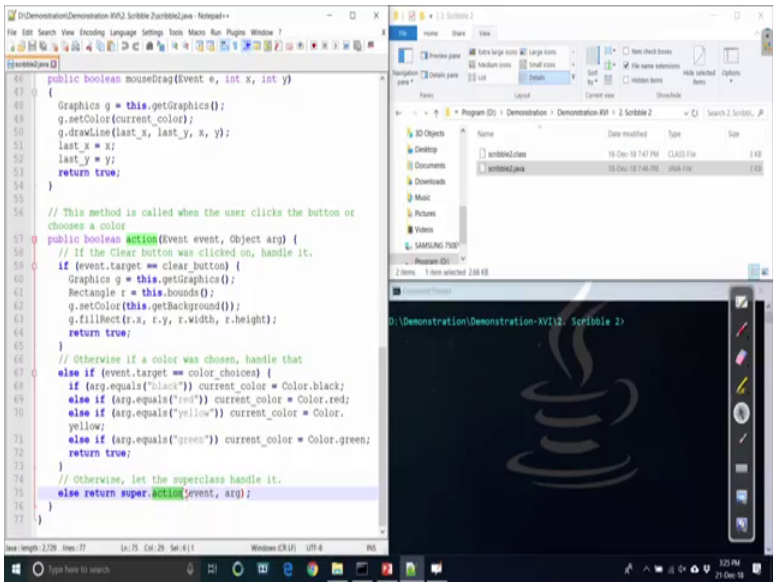



Now, here you can note it here ok. So, they are actually MouseClicked it is there. Because you have clicked it and then if we drag it you can see no action will be there and then clicked ok. So, initially as you see there is no action. So, just it appeared started and then whenever it is clicked is there then ok. So, it is basically no mouseClick event is there now if we clicked it, then we can see around it is showing the mouse has been clicked like. So, this is idea about.

And then mouseMoved is different from the mouse that in the sense that, if we clicked and then just simply move without dragging that is a mouse moved actually and then mouse exited also just opposite to the click; that means, is just releasing these are the things are there. So, if we write some code here and accordingly that code whenever that event will occur, it can be generated and then we can show it. So, it is left as an exercise you can write some code and then for every event that it may takes place in this context you can see the corresponding output.

So, you can use a g drosting method to display corresponding to each event handling mechanism event handling routine whatever you have mentioned here. Now this is the scribble one which basically extends and applet the scribble applet the another example also same scribble applet, but using the lower version of the JDK toolkit where actually the different methods were there those are defined here namely action method that mouseDrag method mouseDown method namely. So, those methods are deprecated now.

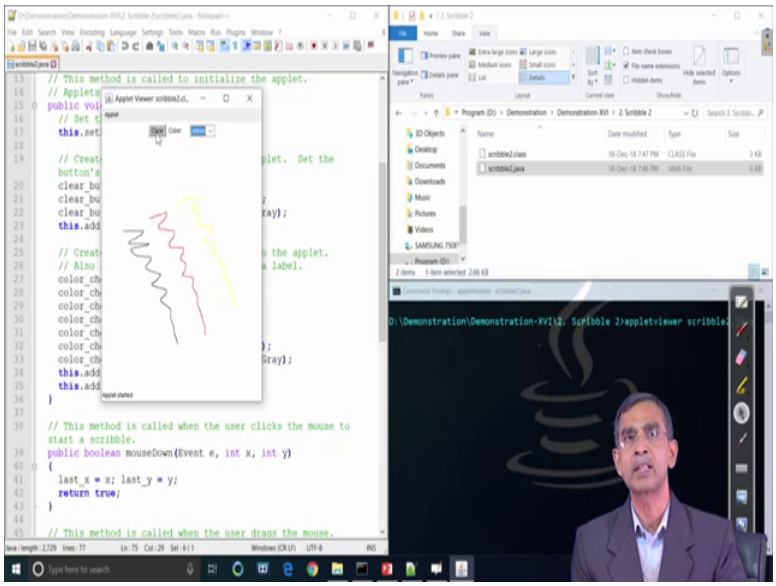
(Refer Slide Time: 09:41)




Now, we have a new methods as you have discussed here. Now this is basically based on the old one just for a simple favor of how the event handling routine can be written in a program and that can be handled by java system runtime manager.

So, in this example as we see the little bit is advanced example we can say, it has other than the scribbling only it also has the two more options namely cleaning; that means, it will basically clear whatever the portion that you have scribbled so, far and then another option is that choice in the color. So, we ca if we see that applet first then you can see exactly how the applet will look like and then you can understand the code that has basically has planned to implement these kind of applets.

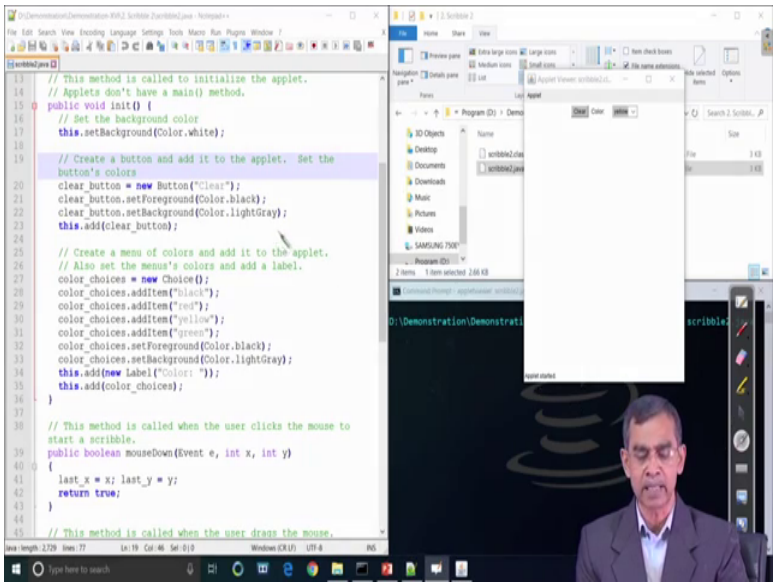
(Refer Slide Time: 10:29)



So, let us have a quick view of this applet here yeah. As you see here this is an applet window; in this applet window you see two buttons here is the one clear and another is the button is called the color ok. Now the currently color is the called as black; now if we draw something just a scribbling we see something it is there exactly it is black; now if we change the color. So, you just is a is a choice item you see and in the choice box we can select some color say red and it is showing that red is there.

Now, if we select the fine another color right you can see the different color now if we see the clear button, then what the things that you have drawn it will basically repaint. So, repaint everything with the previous background. So, its clean it. Now this is our applet now let us see how we have designed this applet what are the means methods that we have used it here.

(Refer Slide Time: 11:33)



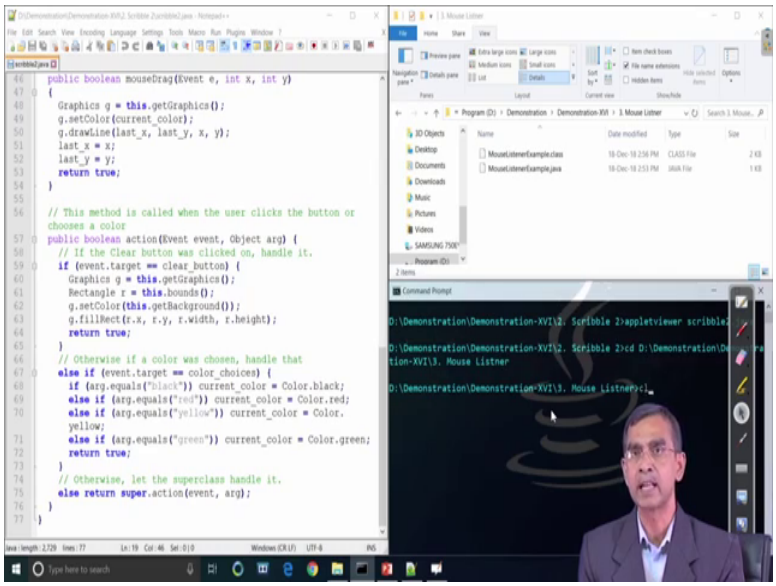

And is this program as we see we have declared two components namely button, clear button and then color choices as a choice button the choice component these two buttons these two components have been declared. And in the init method as we see we have just decide the background of these applet is as white, and then we create the clear button with a level clear and then clear button the foreground and background has been decided as a black and the light gray as you can see the background the button appears with a font as the la font as the black and then background of the button as the light gray and then we add this button into the applet screen.

So, the add method is called for that. And similarly the color choices we have included several color like black, red, yellow and green and then color choice items has been created and they have been added into the add item like. So, these are choice list is created and finally, we decide we set the foreground of this color choice as black and then background is a light color.

So, you can see the both button and then choice component looks very similar in the sense of color settings and others, and then we deci we add 1 level as a color to indicate this choice is basically to relate to the color only. So, a level has been added side by side this 1 and we add all these things into the applet.

Now,. So, this is. So, far the applet design is concerned as a.

(Refer Slide Time: 13:17)




Now,. So, if we do it do not include any even handling mechanism the applet will appear it is like this there. Now we have to implement the listener method here, now here we first implement the mouseDown mouseDown method is same as the previously described mouse the mouse pa paste method that is discussed in the scribble 1 and the mouseDrag is also very similar to the mouseDrag method that we have discussed in the scribble 2. So, we do not want to discuss it explicitly and next another is that action event that we have to write it is basically handling the whatever any event occurs.

So, event is basically any object of type event if it is created and object art is basically tell about which is the event actually we have called it there. Now if event or targets that mean if we click say button so, that is a event of target is clear button then it basically clean the repaint the entire what is called the applet area. So, that is why set color field rate by that default color is been set.

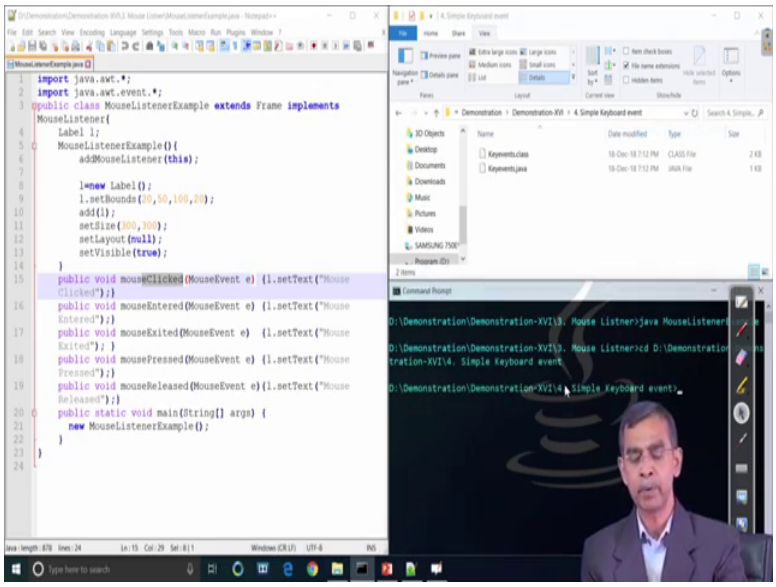
Now if event or target is color choice. So, this is basically the action event is related to the choice selection actually. So, if the event dot target is equals to color choice and then it will basically what it will do, depending on arts basically if it is black then it will set color black and otherwise whatever the different color will be choice according the color will be set and it will return this action to the caller actually.

So, this is the way by which the 3 actions related to the mouse handling the mouseDown and in mouseDrag and then another 2 actions related to the clicking on the button or

clicking on the choice. So, action handling return has been designed and you have already shown obtain the output that it will show the output this one.

So, my advice again is that you can try this program with your own settings and then run of your own and then change whatever you want to do and let us see what is the effect of the change that you can get it. So, change in the sense that if you change in the method it will basically reflect it automatically ok. So, this is the scribble tool.

(Refer Slide Time: 15:37)




And then our next example is MouseListener it is very simple example it is very simple rather very trivial example we should say. So, there are different mouse re mouse event actually likes a mouseClick mouseEnter mouseExited mousePress release some fa features or you have already familiar we have already experienced in scribble 1 applet.

Here again we have to do; it we will just do what exactly the thing is that if some event occur we use the so, status method to show this. Here actually we do not instead of this 1 a level will be there and for this level that text will be changed according to the event that it occurs. So, the accordingly the code that is here as you know it is basically the AWT program without any applet.

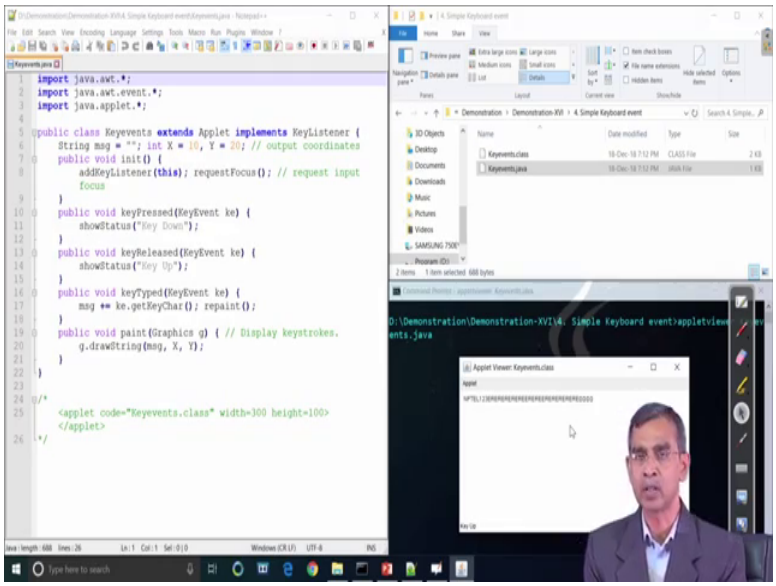
So, it is extensive frame. So, whatever the things it will be components and other things it basically will be will be include in the frame itself and as it include mouseListener. So, MouseListener method or interface are to be here and then we first do the registration by

add MouseListener this and then we decide the level 1 the component, we display this component and then finally, we write the event handling method namely MouseClicked here simply setting the text for the level whenever the text as the basically related to the particular event like loop.

So, this is the 1 example very simple example as you can understand it I hope and now here is the output you can see. Let us resize the window so, that we can see it yeah. So, now, here as you see here. So, if we clicked then you can see the level will be mousePressed like and if we add if we mouseReleased also you can see the mouseReleased be there and so, mouse play pressed mouseReleased MouseClicked all these message are coming because I am here controlling I am pressing clicking deleting all these kind of things are doing as a result you can see that display is also changing. So, this is an example related to the mouse event handling.

Now, I think it is time to discuss about key hand KeyEvent handling one example and this is a very simple example here the.

(Refer Slide Time: 17:57)




So, whenever there is a KeyEvent occur, all the KeyEvent can be categorized into either keyPressed keyReleased or keyTyped. So, there are three events. So, if we can devi de define these 3 events handling routine in our program. So, it is enough and all the other things that we have to add the; we have to do registration the listener object. So, in this case we have to addKeyListener; addKeyListener is the listener here in this case the

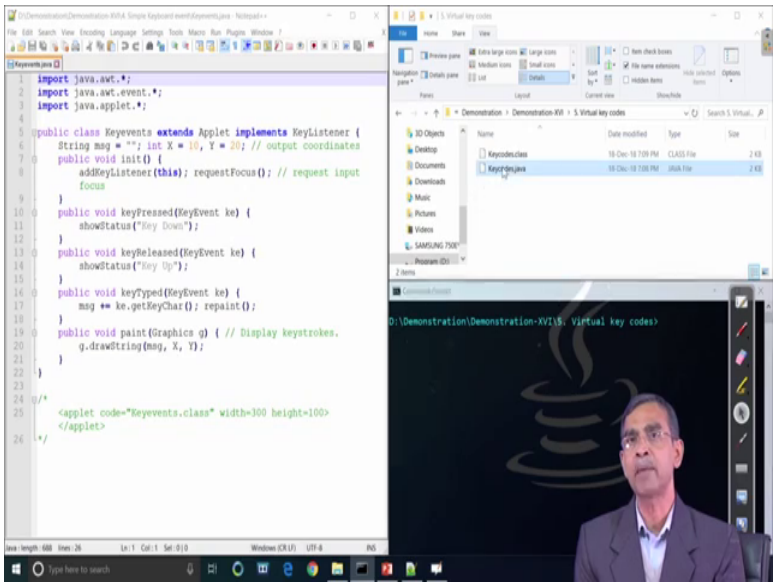
listener will be watchful to see if any event occurs at any instant and then also one thing you have to use it to call the request focus; that means, the AWT should instruct the runtime manager to request focus for the keyboard in ta if any.

So, this is the thing and in this case the definition of is the method keyPressed keyReleased and then Key tip a simple method. So, status method has been called for it and then it will basically display the status at the status bar like. So, if we and these are also an another applet program so, that is i so, status is accessible to us as by means of extending it. So, it is a very simple. So, status method we are calling it here and as you see here ok. So, initially as you see so, status applet started. So, there is no event occurred.

Now, I am just going to k keep list as a keep list as you see the Key Up or Key Down key release all those things are coming here ok. So, it is there and if we write something yeah Key Up Key Down at an actually whenever the KeyEvent occurs there are 2 methodologies Key Up and key din because you easily key press and then Key Down at is like this. So, Key Up Key Down automatically side by side actually and then keyTyped also if you see if we type some character that character will be displayed on the front area.

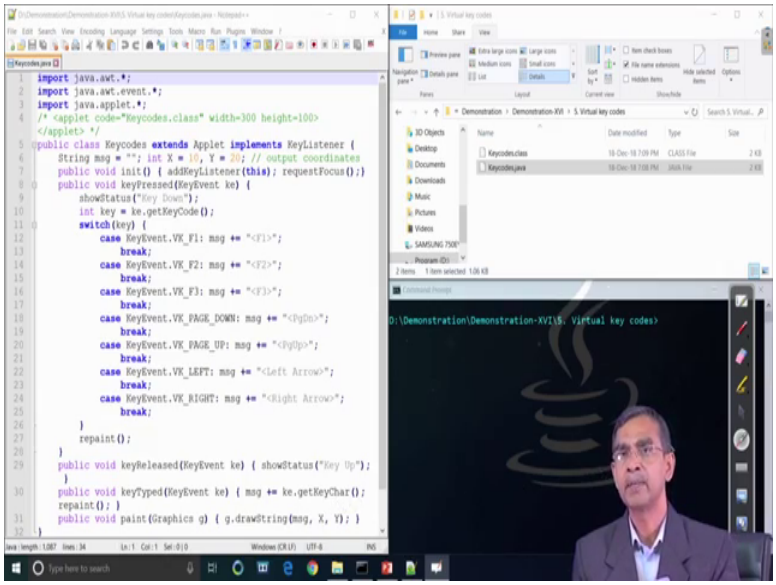
Here for example, we are typing like NPTEL sort of thing as it is coming here on this applet screen. So, you ca hope that although is a very simple example trivial example, but only for the sake of illustration and then demonstration I hope it is understandable.

(Refer Slide Time: 20:17)




Now, we will discuss about another see there are three different type of keys that is possible.

(Refer Slide Time: 20:21)




And if we types any function key or some control alter shift and everything corresponding to these there will be no character should be displayed because they do not have any character assigned for that.

On the other hand a to z and some numeric characters they have been assigned. So, if we type it they are correspondingly key type and then for that key type the event will be

occurred; that means, we will be able to read from the event that it happens and then from this event what is a character that it has been typed can be obtained.

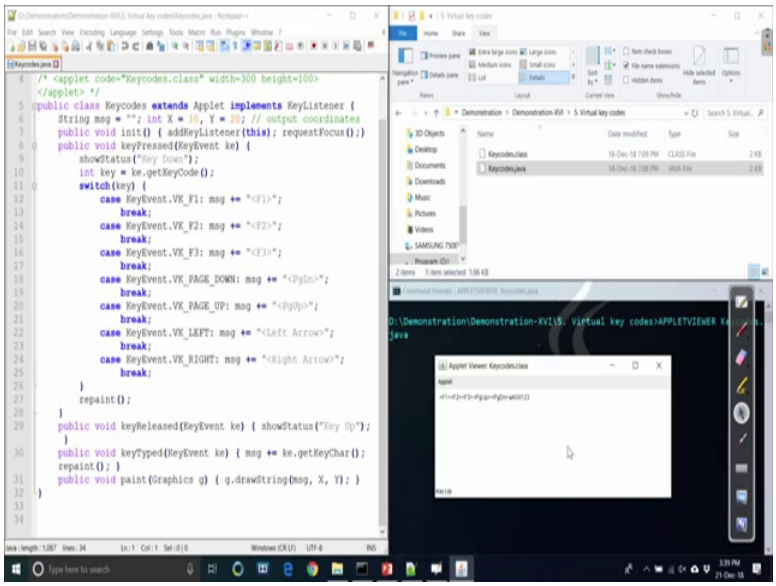
Similarly, what are the function keys or the control or other keys has been obtained that can be also understood from the event itself that it happens there. Now for this event class has defined many variables many constants actually. So, we can read all those values and then according we can understand that, which keys has been pressed by the user. So, this example illustrate the fact that there are different type of key can be pressed and according to these the different output can be obtained through extracting the event that it has caught by the program.

So, this example as we see KeyEvent dot VK underscore a form; that means, if the function 1 key is pressed; that means, event is related to this one then it will just give a print that function 1 is there and like this functional function to function see page up page down page up all the keys those are not having ASCIII code returnable. On the other hand if we type some other characters key, then you will type it and then it will display this one accordingly our key typed event methods needs to be modified.

Here we have done it keyReleased. So, status actually Key Up and then keyType it basically typed whatever the character that we have typed gate key character from the event that it had happened corresponding to this type key, we can get it and just to go on appending and that message can be displayed on the area there applied area and it basically that steam that message steam can be displayed by means of paint method it is there.

So, these are example now have the demo so, that you can see how this program works here yeah.

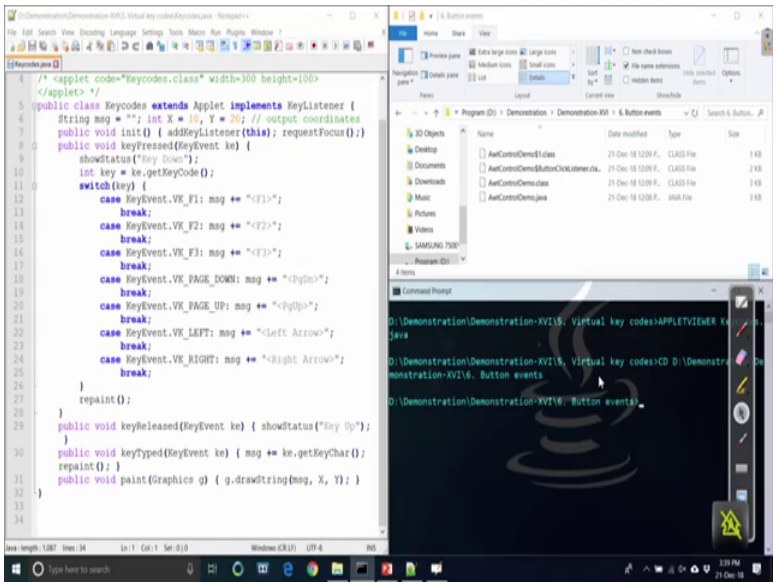
(Refer Slide Time: 22:41)




So, as you see here. So, these are applet has started right now, no event initiated right now I am going to type a font. So, if we press [FL] 1 as you see this one similarly F 2 and then F 3 and so, on. So, you can see now I am typing a capital A, as you see typing capital A it will sense it and then it also displayed there and whatever the type character is there. So, 0 1 2 3 also we can see it is also going to sense it.

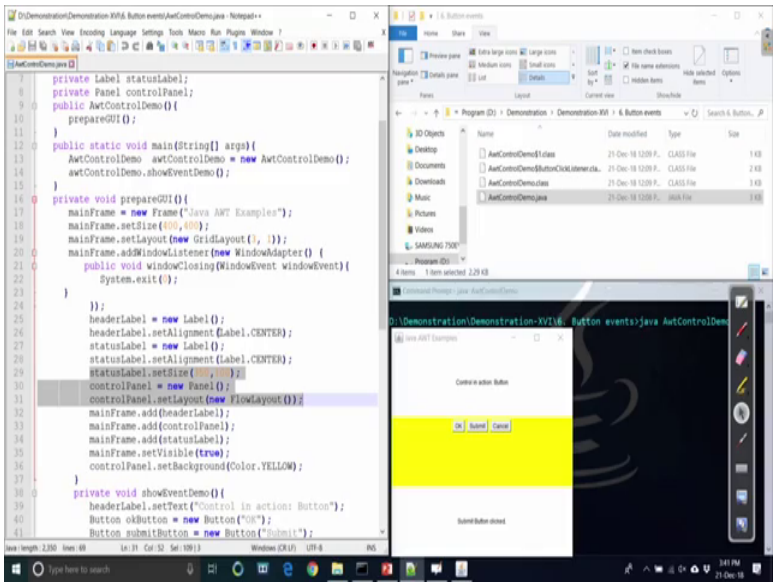
So, it is basically the event is that we are pressing some keys whatever is a function or whatever the character or numeric key whatever it is here. So, these are event that is the source is a key and then event handling (Refer Time: 23:20) hence read it and then they can take the action, the here action is basically printing the same text on the applet screen here.

(Refer Slide Time: 23:31)



So, this is related the KeyEvent and here let us have little bit complex examples.

(Refer Slide Time: 23:37)



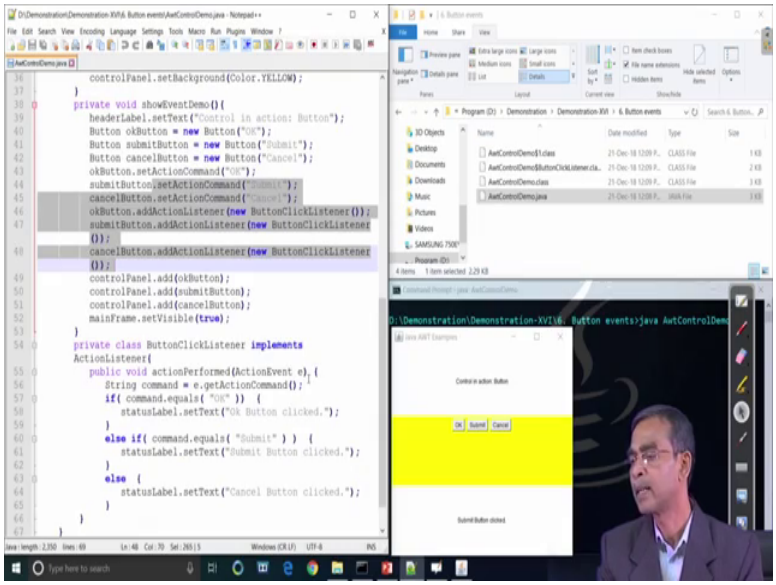
Here actually we want to give that example relate to how the button can be manipulated button can be handled. But this program in addition to button there are few more items are there first let us have the look of this applet, then I can come back to this program so, that you can understand exactly what are the concepts are here it is followed. Yeah as you see here little bit much more components are involved what are the components?

First of all the frame it is and frame contains one panel which is colored as a yellow, and you see in the frame also we include some ta text area the what is called the control in action and then button this is basically the level we can say and so, this display the level, level is included in the frame and fine and in addition to these there are three buttons as you have seen that button and then another is submit and then cancel.

So, that is also if we click as you see that are ok. So, output will be displayed by means of another level and that button clicked. And similarly if we cancel then we can see another the cancel button click. So, you can understand so, how the events are handled. Here we are clicking a particular button corresponding to this button a particular display message appears on the frame itself.

And this is the program we have implemented in this applet here as we see we declare as you see here we declared the frame, level panel whatever the things it is there and they have been initiated and size and everything has been precede. For example, it is for the frame and it is for the level and it is for the panel and everything that we have discussed and then as you have already have an idea about how to configure a panel, how to configure a feel a frame and whatever the every things are there their shape layout pattern and everything all these things are there.

(Refer Slide Time: 25:41)



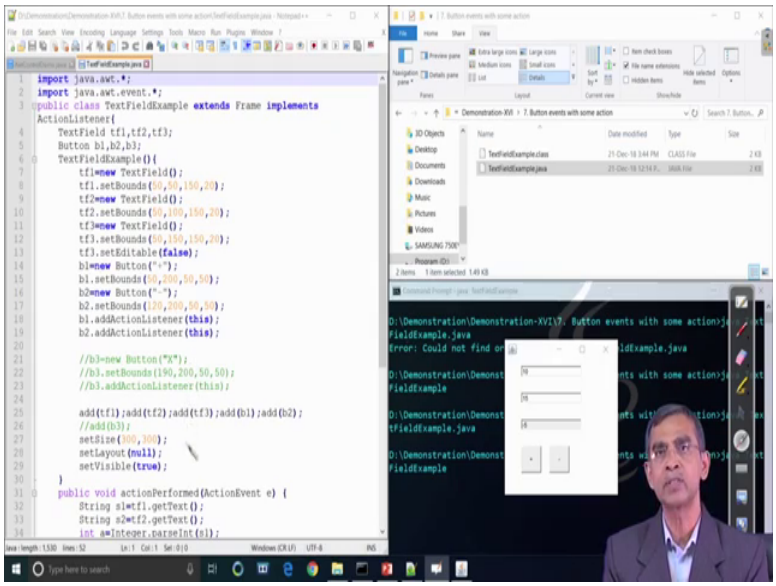

So, if you go through with your own time you will be able to follow it clearly. I will better come to the event that we have increases there. So, here we have declared one

method of our own called the showEventDemo; and in this method we just assign the add setActionCommand setActionCommand and command setActionCommand set action listener and then action listener for the different buttons and different buttons mainly.

So, here for example, button and then submit button and cancel button for all these buttons we defined the different methods that it can handle actually and this basically the registration tasks and then control panel just add different buttons on each. So, button submit button and cancel button and finally, the frame is now set as visible

And then button click listener is basically the method which we have planned you listen to the button click this is the main class in this case actually and in this method as see if you see we have different actionPerform this is the main method that needs to be defined by us and then so, that according to that what will be the text will be there. So, if we press this one button click, then it is a status level dot set text will change at that thisone that is this button is click like this. So, this is a program it is basically event handling.

(Refer Slide Time: 27:07)




So, again you if you run the program of your own, then only you will be able to understand much more things in details. So, this may be only 10 percent of your learning, but you have to run it of your own and then see exactly how the things are coming. If you tend something and then you have to see exactly whether the change that you have done according to your own wish or not.

So, this is the idea about this one and this example as we see here ok. So, now,. So, another example that we are going to discuss about, it is another simple example this example relate to the text field in addition to button. So, here are basically two text field area, it we can say it is a simple calculator the calculator we will have only add and subtract methods are the two buttons add and subtract and that two text field as we see on these two text field for taking an input.

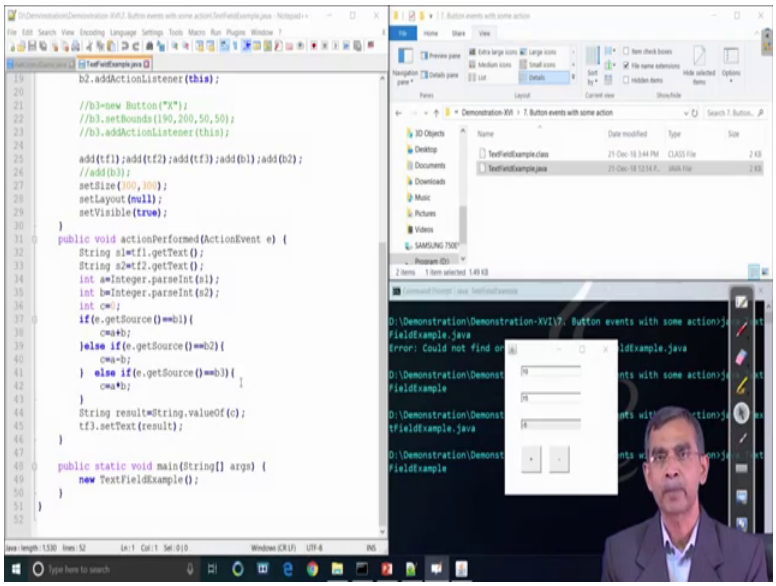
So, it is also even can be added into that text field area, now we will see that how the even can be added in the text field area and then the third text field is basically so, the result. So, he probably types say 10 in the first text field and then f15 in the second text field and then if we try plus and you can see the result will be 25.

If which check in minus as we see the result will be changed to this one. So, this way the it is a simple calculator look like, but this can be extended to have right your I mean much more complex calculator actually.

Now, here you see what we have done first you have to create the 3 text field that we have done here t f 1, t f 2 and t f 3 and usual the routine that we have already learned about it how to create a text field, how to size of the text field can be case it or is that everything that we have done here. So, just first few lines is basically related to how we can create 3 text field and then we have decided that two button, how the two button can be defined.

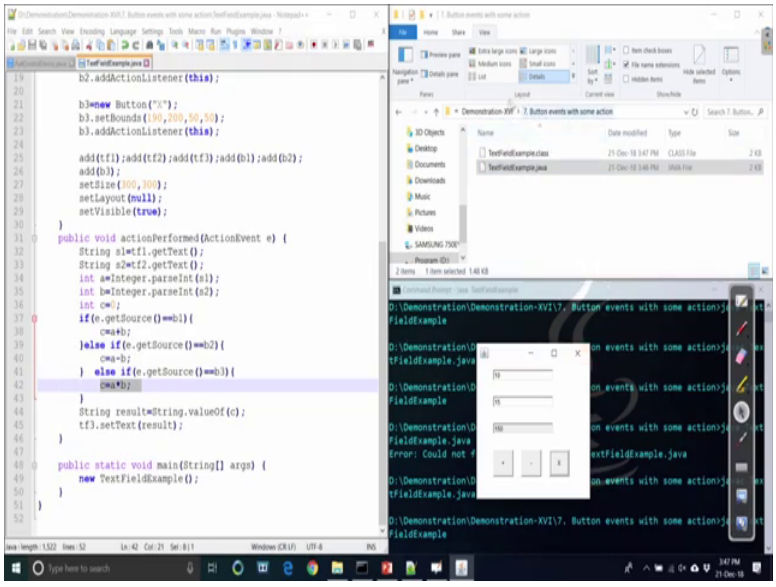
So, here we have declared the two button b 1 and b 2 namely b 1 is for plus and b 2 is for minus, and then we have to add the action listener as a registration. So, b 1 dot add action listener b 2 dot add action listener. So, if any action have if anything happened to this button. So, this action listener will listen to that and accordingly the routine that we have to develop is basically in the next method actionPerform.

(Refer Slide Time: 29:31)



In the actionPerform as you see we read s 1 and s 2 the two string from the two text field and then there needs to be converted integer,we have done it as a by means of integer dot par seeing method. And then we have to take about event if the event is e related to the clicking a button. So, e dot gate source and if it is a b 1 so, it basically at the result and otherwise we just add subtract from a to b a minus b like and finally, we display the result converting into the string and then we display the result on to that text full third text field. So, this is idea about it.

(Refer Slide Time: 30:21)

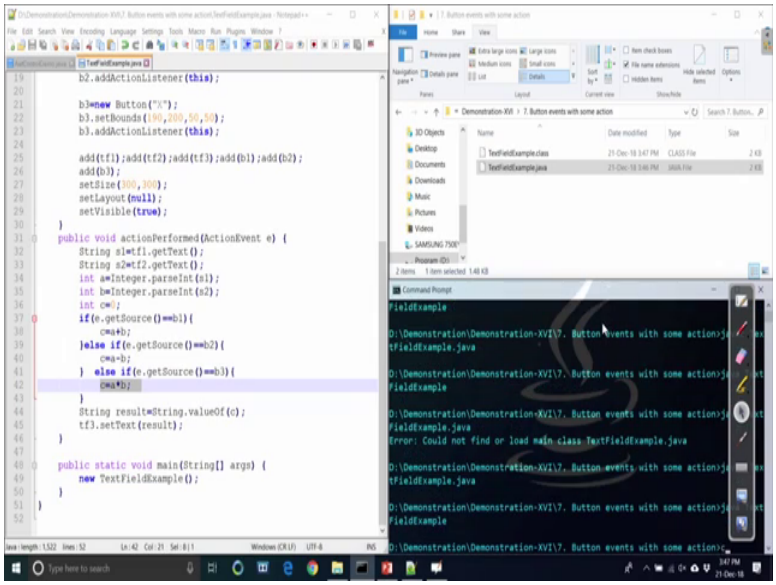


Now, I just want to add it say suppose I want to add one more button, and you will see how the things can work for you. So, one more button lame namely say for multiplication I just add one more button here. Yes I am adding one more button that is all. So, multiplication and then in the action performed routine I have to change it, because we have plus button.

So, if the plus button is there. So, another e field part we can add another else part. So, else if e dot gateSource is equals to equals to b 3 then action that will be performed is c equals to a star b. And then if just compile this and then you see how the change in the applet will be and like this ok.

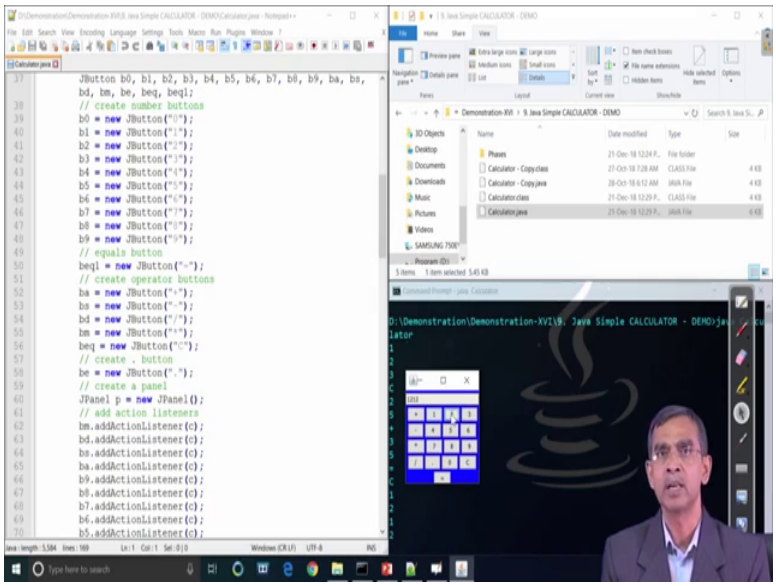
So, now we can see the applet here yeah. So, applet has been changed as you can see if any value you can add in the two text field plus minus and then star as you see the result is coming.

(Refer Slide Time: 31:15)



So, this is the way that we can add components at the same time for each component we can register the listener objects and then according to for each component, we can decide what are the event handling routine that needs to be associated in each component and that is all so, far event handling is concerned.

(Refer Slide Time: 31:33)



Now, I will quickly come to the swing and then we have right idea about the swing actually, and this is the calculator that we can use using the swing. Here the swing calculator we have discussed in our theoretical class. So, it is basically the same thing we have done it, now if we run it then you can see how the swing calculator look like yeah.

So, here is the swing calculator is a typical look here as you see here. Now in the swing calculator as you see 0 to 9 10 keys are there 10 buttons rather they are labeled as 0 1 2 2 9 only. So, basically all these buttons are to allow user to type 0 to 9 any now integers and then the user using this button can type any numbers for example, 2 5 like this.

And then there are some operation button like plus minus then multiplication then subtraction and some buttons like clear c and finally, there is another button equals. So, if we write 2 5 and then plus then 3 3 0 and then and then equals then you can understand what the result it will see. And then if we again click key c then it is basically clean it is there anyway. So, this is the part of this applet calculator as we see here.

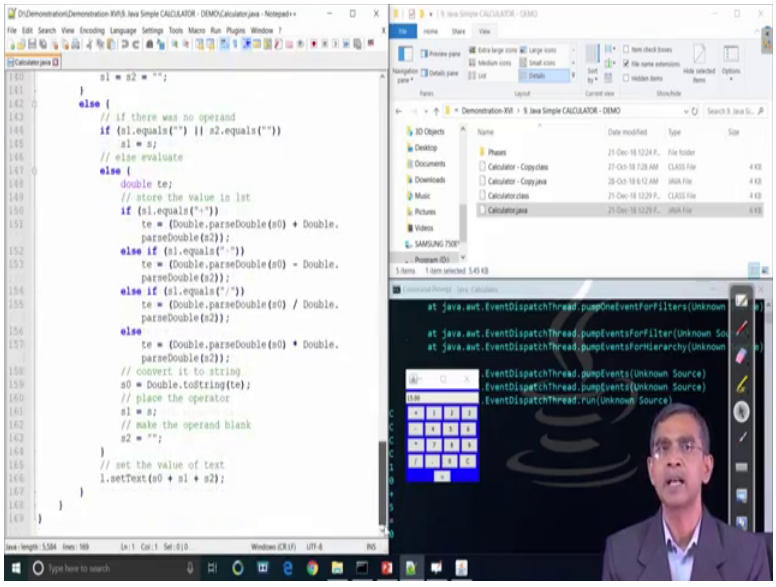
So, here basically idea about designing this calculate, I just leave it with the program you can have it and then you can check the program of your own according to your own convenient time then you can understand much more about it. I just want to see the methodology or the steps that you should follow. In the first step whether it is an applet or frame you decide accordingly the things will be there in this case we have used a frame. So, this is not an applet and then we have to decide the frame and then you have

to I mean create one panel usually it is customary to include all the components inside the panel so, that it can be more better configurable. So, you have to clear the panel also.

And this panel includes so, many keys are there, 0 1 2 all these things button actually there. So, here actually 17th buttons as we see, 16 including operations and on the last one is equal. So, those buttons have been declared and then finally, how the action listener is to be registered here in this case the action listener is basically mouse key as you see. So, mouse clicking actually; we are we are entering the number not from the keyboard rather we are entering number from the keys on the calculator. So, the mouseClick is required here. So, method that we have to add it basically the mouse key MouseListener to be add MouseListener is the method that needs to be added here.

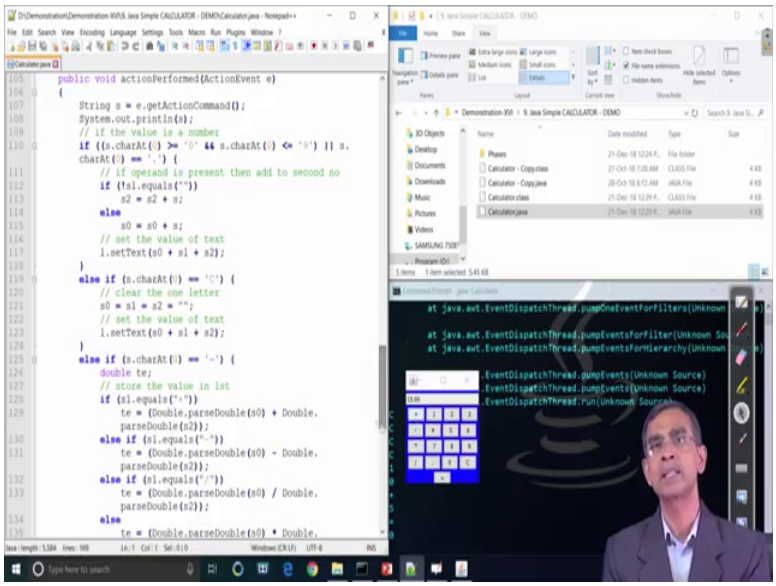
And then once it is there for every keyboard we have to add the action listener and then action listener method is defined accordingly. So, that it will basically send that what is the type that the user has ta I mean keyed rather, and then accordingly it will do the results whether depending on plus button or minus button or multiplication button or division button is the operation will be carried out. So, this is the right code; code for action handling as if you can see later on. You will see that a little bit optimized code it is used here, but you can write the course according to your own understanding and write it there.

(Refer Slide Time: 35:05)



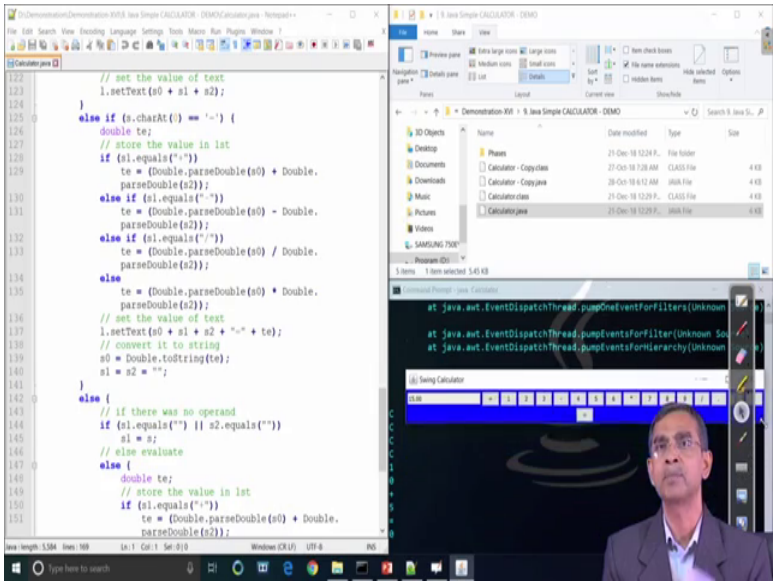
So, I do not want to say that you have to follow these code.

(Refer Slide Time: 35:09)



You may understand these code very easily or you can find it little bit difficult to understand. But anyway, but you can think you can see that how you have to handle the key that is your own idea should be. And then accordingly you can rewrite this code re rerefine these codes and then you can use it.

(Refer Slide Time: 35:29)



So, this is about the event handling this is the start and very beginning of the even handling me mechanisms as I said and more on the things depends on and you can learn

much about if you develop many more programs many more tasks that you have to do it and you can follow it. So, that is all for today.

Thank you very much.