

The screenshot shows a presentation slide titled "Adapter Classes". The slide has a blue header bar with the title. Below the header is a sidebar containing navigation links for "Outline", "Delegation Event Model", "Event Classes" (which is currently selected), and "Event Sources". The main content area contains a bulleted list of points about adapter classes, followed by a table mapping adapter classes to their corresponding listener interfaces. At the bottom of the slide are navigation icons and the date "November 9, 2020".

Adapter Classes

Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent
WindowEvent

Event Sources
Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener
WindowListener

■ Adapter classes are used to simplify the process of event handling.
■ An adapter class provides an empty implementation of all methods in an event listener interface.
■ Adapter classes are useful when you want to receive and process only some of the events that are handled by a particular event listener interface.
■ To use adapter class-define a new class to act as an event listener by extending one of the adapter classes and implement only those events in which you are interested.

Adapter Class	Listener Interface
ComponentAdapter	ComponentListener
ContainerAdapter	ContainerListener
FocusAdapter	FocusListener
KeyAdapter	KeyListener
MouseAdapter	MouseListener
MouseMotionAdapter	MouseMotionListener
WindowAdapter	WindowListener

November 9, 2020 Atul Chaudhari 47/51

AC

AP

Atul Chaudhari



Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent
Event Sources
Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener

MouseEvent class contd...

■ Methods:

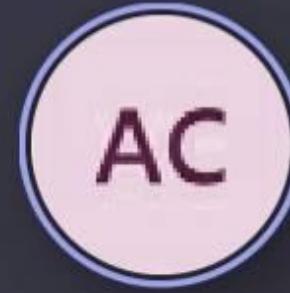
`int getX()`- return the X coordinates of the mouse within the component when the event occurred

`int getY()`- return the Y coordinates of the mouse within the component when the event occurred

`Point getPoint()`- used to obtain the coordinates of the mouse.

`int getClickCount()`- used to obtain the number of mouse clicks for this event.

`int getButton()`- It returns a value that represents the mouse button that caused the event.





Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent
Event Sources
Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener

MouseEvent class contd...

Constructor:

MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)

Here,

src:- reference to the component that generated this event

type:- indicates type of the event

when:- indicate system time at which the mouse event was occurred

modifiers:-indicates which modifiers were pressed when this mouse event occurred

x:- contains x coordinate of the mouse

y:-contains y coordinate of the mouse

clicks:-contains count Contains clickcount

triggersPopup:-flag indicates if this event causes a pop-up menu to appear on this platform





MouseEvent class contd...

Outline

Delegation Event Model

Event Classes

ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent

Event Sources

Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener

Atul Chaudhari

November 3, 2020

Atul Chaudhari

AC

AP

■ **Constructor:**

`MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)`

Here,

src:- reference to the component that generated this event

type:- indicates type of the event

when:- indicate system time at which the mouse event was occurred

modifiers:-indicates which modifiers were pressed when this mouse event occurred

x:- contains x coordinate of the mouse

y:-contains y coordinate of the mouse

clicks:-contains ~~contains~~ count

triggersPopup:-flag indicates if this event causes a pop-up menu to appear on this platform

11:48:38

HD 4G 31%

Unit-III Event Handling.pdf - Adobe Acrobat Reader DC

File Edit View Sign Window Help

Home Tools Unit-III Event Hand... Lab Manual_AJP_2...

28 (28 of 43) 145% Sign In

MouseEvent class

Outline Delegation Event Model

Event Classes ActionEvent ItemEvent AdjustmentEvent TextEvent ComponentEvent ContainerEvent FocusEvent KeyEvent MouseEvent

Event Sources Event Listeners ActionListener ItemListener AdjustmentListener TextListener ComponentListener ContainerListener FocusListener KeyListener MouseListener MouseMotionListener References

■ A MouseEvent is generated when:
Mouse button is pressed, Mouse button is released, Mouse is clicked, Mouse entered the component, Mouse exited the component, Mouse is moved, Mouse is dragged

■ MouseEvent is a subclass of InputEvent.

■ MouseListener and MouseMotionListener are used to process MouseEvent.

■ Constants:
There are eight types of mouse events, which are identified by these integer constants:

MOUSE_PRESSED: The mouse was pressed.
KEY_RELEASED: The mouse was released
KEY_CLICKED: The user clicked the mouse.
KEY_ENTERED: The mouse entered a component.
KEY_EXITED: The mouse exited from a component.
KEY_MOVED: The mouse moved.
KEY_DRAGGED: The user dragged the mouse.
KEY_WHEEL: The mouse wheel was moved.

Other constants defined by MouseEvent class:
NOBUTTON Button1 Button2 Button3

Atul Chaudhari November 3, 2020 Atul Chaudhari 28/43

Type here to search

Scanned by TapScanner

11:10

⌚ ⌂ ⌂ HD 4G 77%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

25 (25 of 27) 154% Tools Sign Comment

Bookmarks

- Delegation Event Model
- Event Classes
 - ActionEvent
 - ItemEvent
 - AdjustmentEvent
 - TextEvent
 - ComponentEvent
 - ContainerEvent
- Event Sources
- Event Listeners
 - ActionListener
 - ItemListener
- References

ItemListener interface

Outline

Delegation Event Model

Event Classes

ActionEvent

ItemEvent

AdjustmentEvent

TextEvent

ComponentEvent

ContainerEvent

Event Sources

ActionListener

ItemListener

References

October 28, 2020 Atul Chaudhari

This interface defines the itemStateChanged() method that is invoked when an item event occurs.

Syntax:

```
public void itemStateChanged(ItemEvent ie)
```

Atul Chaudhari

Search the web and Windows

AC AP

11:06

⌚ ⌂ ⌂ HD 4G 78%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

24 (24 of 26) 154% Tools Sign Comment

Bookmarks

- Delegation Event Model
- Event Classes
 - ActionEvent
 - ItemEvent
 - AdjustmentEvent
 - TextEvent
 - ComponentEvent
 - ContainerEvent
- Event Sources
- Event Listeners
 - ActionListener
- References

ActionListener interface

Outline

Delegation Event Model

Event Classes:

- ActionEvent
- ItemEvent
- AdjustmentEvent
- TextEvent
- ComponentEvent
- ContainerEvent

Event Sources

Event Listeners

ActionListener

References

This interface defines the actionPerformed() method that is invoked when an action event occurs.

Syntax:

```
public void actionPerformed(ActionEvent ae)
```

October 25, 2020 Atul Chaudhari

Atul Chaudhari

Search the web and Windows

AC AP



Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
Event Sources
Event Listeners
ActionListener
References

ItemEvent class

- An ItemEvent is generated when:
 - Checkbox is clicked
 - List item is clicked
 - Choice selection is made
 - Checkable menu item is selected or deselected
- **ItemListener** is used to process ItemEvent.

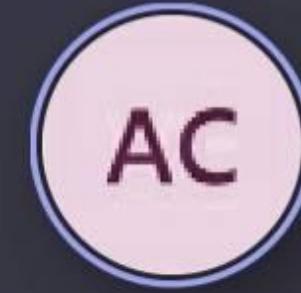
- **Constants:**

There are two types of item events, which are identified by the following integer constants:

SELECTED: The user selected an item.

DESELECTED: The user deselected an item.

ITEM_STATE_CHANGED: Signifies a change of state





AJP: Unit-II Event Handling

24:26 105 attendees



The slide title is "ItemEvent class contd...". It shows a UML class diagram with `ItemSelectable` as the superclass, and `getSelectable()` and `getEntry()` methods. `EventObject` is another superclass, and `AVTEvent` and `ItemEvent` inherit from it. `ItemEvent` has its own method `getStateChange()`.

Constructor:

```
ItemEvent(ItemSelectable src, int type, Object entry, int state)
```

Here,
src:- is the object that generates an event
type:- indicates type of the event
entry:- contains the specific item that generated the item event
state:- represents current state of item that generated the event

Methods:

- `ItemSelectable getItemSelectable()` - used to obtain a reference to the ItemSelectable object that generated an event.
- `Object getItem()` - used to obtain a reference to the item that generated an event

For example, if checkbox is selected or deselected it returns label associated with checkbox that generated event.

`int getStateChange()` - method returns the state change (that is, SELECTED or DESELECTED) for the event.

October 25, 2020 Atul Chaudhari 15/26



Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

23 (23 of 26) 154% Tools Sign Comment

Event Listeners

Outline Delegation Event Model

Event Classes ActionEvent ItemEvent AdjustmentEvent TextEvent ComponentEvent ContainerEvent

Event Sources

Event Listeners ActionListener

References

October 25, 2020 Atul Chaudhari 23/26

■ Listeners are created by implementing one or more of the interfaces defined by the java.awt.event package.

■ When an event occurs, the event source invokes the appropriate method defined by the listener and provides an event object as its argument.

Windows Search the web and Windows 10:57 AM 10/27/2020

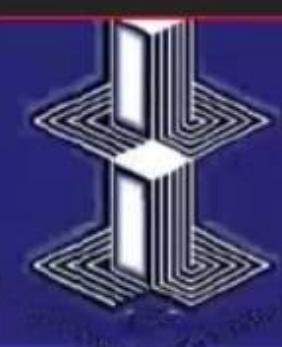


The slide title is "ActionEvent class contd...". It shows the class hierarchy: ActionEvent (src, type) inherits from EventObject and AWTEvent. The class has three constructors: ActionEvent(Object src, int type, String cmd), ActionEvent(Object src, int type, String cmd, int modifiers), and ActionEvent(Object src, int type, String cmd, long when, int modifiers). The third constructor is annotated with handwritten notes: "src:- is the object that generates an event", "type:- indicates type of the event", "cmd:- contains command string", "when:- specifies when the event occurred", and "modifiers:- indicates which modifier keys (ALT, CTRL, META, and/or SHIFT) were pressed when the event was generated." The slide also lists methods: getActionCommand() and getWhen(). The footer includes a date (October 25, 2020), author (Atul Chaudhari), and page number (13/26).



...

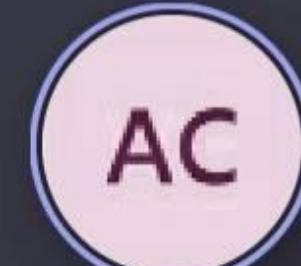
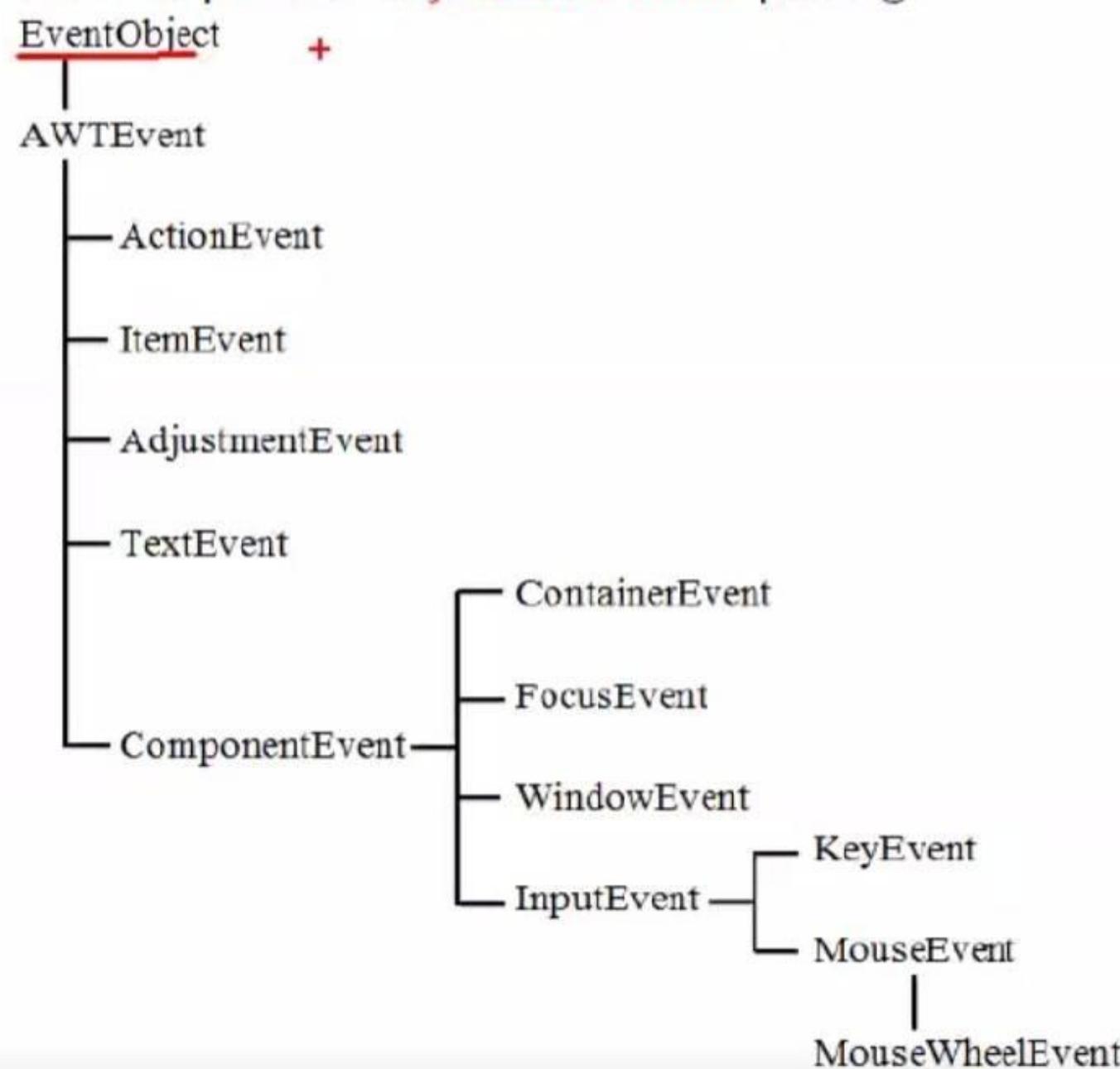




Event Classes

Outline
Delegation
Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
Event Sources
Event
Listeners
ActionListener
References

- Event classes represents an event
- All event classes are present in `java.awt.event` package





Outline

Delegation
Event Model

Event Classes

ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent

Event Sources

Event
Listeners
ActionListener
References

AWTEvent class

- It is subclass of EventObject defined in java.awt package.
- It is superclass (directly or indirectly) of all AWT-based events used by delegation event model.

- Constructors:

AWTEvent(Object src, int type)

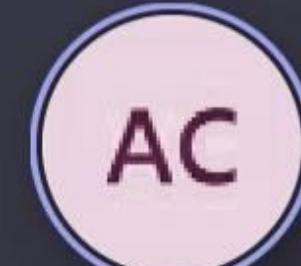
Here,

src:- is the object that generates an event

type:- indicates type of the event

- Methods:

int getID()-It is used to determine the type of event





Outline

Delegation
Event Model**Event Classes**ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent**Event Sources**Event
Listeners
ActionListener
References

EventObject class

- It is root of the Java event class hierarchy.
- It is present in java.util package
- It is superclass for all event classes

- **Constructors:**

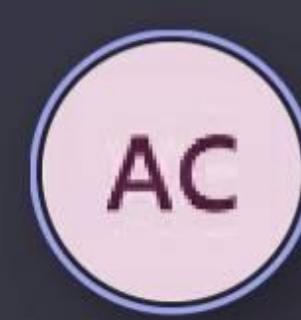
EventObject(Object src)

Here,

src:- is the object that generates an event

- **Methods:**

Object getSource()-This method returns source of the event.





Outline

Delegation
Event Model

Event Classes

ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent

Event Sources

Event
Listeners

ActionListener

References

Events, Sources and Listeners



addTypeListener()

Event (Class)	Source (Class)	Listener (Interface)
ActionEvent	Button, MenuItem, List	ActionListener
ItemEvent	Choice, List, Checkbox, CheckboxMenuItem	ItemListener
AdjustmentEvent	Scrollbar	AdjustmentListener
TextEvent	TextField, TextArea	TextListener
ComponentEvent	Any Component	ComponentListener
ContainerEvent	Any Container (Applet, Frame, Panel)	ContainerListener
FocusEvent	Any Component	FocusListener
WindowEvent	Window (Frame)	WindowListener, WindowFocusListener
KeyEvent	Any Component	KeyListener
MouseEvent	Any Component	MouseListener, MouseMotionListener
MouseWheelEvent	Any Component	MouseWheelListener

Atul Chaudhari

October 25, 2020

Atul Chaudhari

AC

AP

11:35

HD 4G 60%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

8 (8 of 26) 150% Tools Sign Comment

Events, Sources and Listeners

Outline

Delegation Event Model

Event Classes

- ActionEvent
- ItemEvent
- AdjustmentEvent
- TextEvent
- ComponentEvent
- ContainerEvent

Event Sources

Event Listeners

ActionListener

References

October 25, 2020 Atul Chaudhari

Event (Class)	Source (Class)	Listener (Interface)
ActionEvent	Button, MenuItem, List	ActionListener
ItemEvent	Choice, List, Checkbox, CheckboxMenuItem	ItemListener
AdjustmentEvent	Scrollbar	AdjustmentListener
TextEvent	TextField, TextArea	TextListener
ComponentEvent	Any Component	ComponentListener
ContainerEvent	Any Container (Applet, Frame, Panel)	ContainerListener
FocusEvent	Any Component	FocusListener
WindowEvent	Window (Frame)	WindowListener, WindowFocusListener
KeyEvent	Any Component	KeyListener
MouseEvent	Any Component	MouseListener, MouseMotionListener
MouseWheelEvent	Any Component	MouseWheelListener

Atul Chaudhari

Search the web and Windows

AC AP

11:12

HD 4G 66%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

Tools Sign Comment

Listener

Outline

Delegation Event Model

Event Classes

- ActionEvent
- ItemEvent
- AdjustmentEvent
- TextEvent
- ComponentEvent
- ContainerEvent

Event Sources

Event

Listeners

ActionListener

References

October 25, 2020 Atul Chaudhari

■ A listener is the object that is notified when an event occurs.

■ It has two major requirements:

- ① It must have been registered with one or more event sources to receive notification about specific type of event.
- ② It must implement methods to receive and process these notifications.

Atul Chaudhari

Search the web and Windows

AC AP



AJP: Unit-II Event Handling

37:28 84 attendees



Source

Outline

Delegation Event Model

Event Classes

ActionEvent ItemEvent AdjustmentEvent TextEvent ComponentEvent ContainerEvent

Event Sources

Event Listeners

ActionListener

References

October 25, 2020 Atul Chaudhari 6/26

A source is an object that generates an event. This occurs when internal state of object changes some ways.

Sources may generate more than one type of events.

A source must register listeners in order for the listener to receive notification about specific type of event.

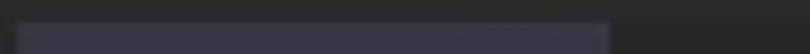
When event occurs all the registered listeners are notified and receive copy of event object

Each type of event has its own registration method. The general form of Method is:

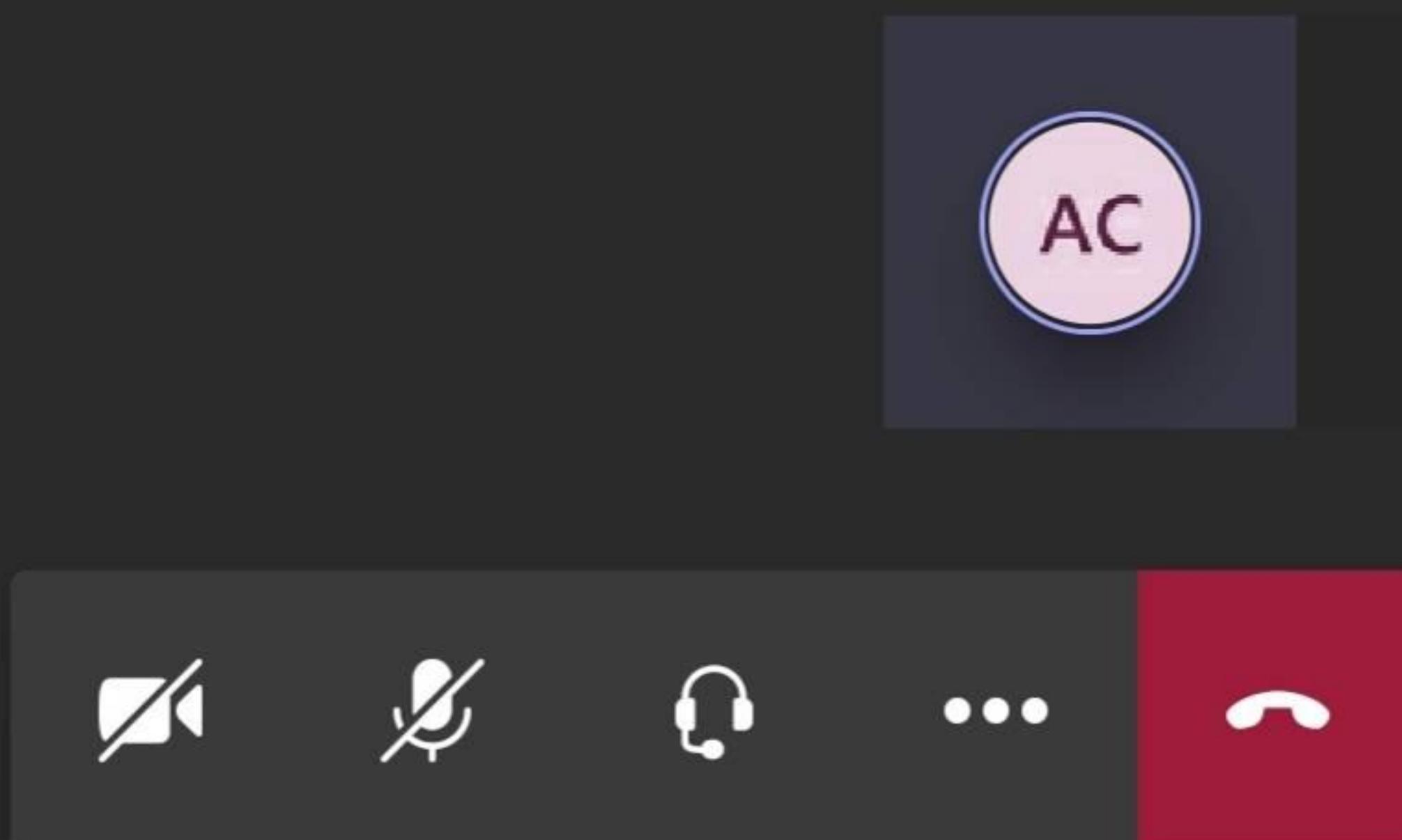
```
public void addTypeListener(TypeListener el)
```

Here,
Type- Name of Event
el- reference to the event listener

Example:
Method that register keyboard event listener is called `addKeyListener()`



6/26



10:28

⌚ ⌂ ⌂ HD 4G 75%

The screenshot shows a presentation slide titled "Event". The slide has a blue header bar with the title "Event" and a sidebar on the left containing navigation links. The main content area contains a red checkmark icon followed by the word "Red", and a red box labeled "Plane". Below this, there is a bulleted list of three items. The footer of the slide includes the author's name, date, and a navigation bar.

Outline

Delegation Event Model

Event Classes

- ActionEvent
- ItemEvent
- AdjustmentEvent
- TextEvent
- ComponentEvent
- ContainerEvent

Event Sources

Event Listeners

ActionListener

References

Event

Red

Plane

- An event is an object that describes a state change in source.
- It can be generated when user interacts with elements in GUI.
- Some activity that causes event to be generated are:
pressing button,
entering character via k/b,
clicking mouse,
selecting item in list.

Atul Chaudhari

October 25, 2020

Atul Chaudhari

AC

AP

10:04

HD 4G 80%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

Tools Sign Comment

Delegation Event Model

Outline

Delegation Event Model

Event Classes

- ActionEvent
- ItemEvent
- AdjustmentEvent
- TextEvent
- CompositeEvent
- ContainedEvent

Event Sources

Event Listeners

ActionListener

References

The modern approach to handle event is based on Delegation Event Model.

It provides standard and consistent mechanism for generating and processing events.

Concept: A source generates an event and sends it to one or more listeners. In this scheme listener simply waits until it receives an event. Once received, the listener processes the event and returns.

There are three main parts of delegation event model:

- 1 Event
- 2 Source
- 3 Listeners

October 25, 2020 Atul Chaudhari

Atul Chaudhari

Search the web and Windows

AC AP

(i) You're joined as an attendee. X

The screenshot shows a presentation slide titled "MouseEvent class contd...". The slide content includes a section on the constructor of the MouseEvent class, detailing parameters like src, type, when, modifiers, x, y, clicks, and triggersPopup. A sidebar on the left lists various Java event-related classes and interfaces. At the bottom, there's a footer with the date November 8, 2020, the name Atul Chaudhari, and a page number 29/48.

MouseEvent class contd...

Constructor:

```
MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)
```

Here,

- src:- reference to the component that generated this event
- type:- indicates type of the event
- when:- indicate system time at which the mouse event was occurred
- modifiers:- indicates which modifiers were pressed when this mouse event occurred
- x:- contains x coordinate of the mouse
- y:-contains y coordinate of the mouse
- clicks:-contains click count
- triggersPopup:-flag indicates if this event causes a pop-up menu to appear on this platform

November 8, 2020 Atul Chaudhari 29/48



AJP: Unit-III Event Handli...

07:56 53 attendees



You're joined as an attendee.



The screenshot shows a presentation slide with a blue header bar containing the title "MouseEvent class contd...". Below the header is a sidebar with navigation links for "Outline", "Delegation", "Event Model", "Event Classes" (which is currently selected), and "Event Sources". The main content area contains a section titled "Methods:" with the following descriptions:

- int getX() - return the X coordinates of the mouse within the component when the event occurred
- int getY() - return the Y coordinates of the mouse within the component when the event occurred
- Point getPoint() - used to obtain the coordinates of the mouse.
- int getClickCount() - used to obtain the number of mouse clicks for this event.
- int getButton() - It returns a value that represents the mouse button that caused the event.

At the bottom of the slide, there is footer text: "November 8, 2020", "Atul Chaudhari", and "30/48".



...





AJP: Unit-III Event Handli...

18:58 81 attendees



You're joined as an attendee.



WindowEvent class contd...

■ Constructor:

```
WindowEvent(Window src, int type)
WindowEvent(Window src, int type, Window other)
WindowEvent(Window src, int type, int fromState, int toState)
WindowEvent(Window src, int type, Window other, int fromState, int toState)
```

Here,
src:- reference to the component that generated this event
type:- indicates type of the event
other:- specifies the opposite window when a focus or activation event occurs
fromState:-specifies the prior state of the window
toState:- specifies the new state that the window will have when a window state change occurs

November 9, 2020 Atul Chaudhari 32/48



i You're joined as an attendee. X

The screenshot shows a presentation slide titled "MouseEvent class". The slide content includes:

- A MouseEvent is generated when:
Mouse button is pressed, Mouse button is released, Mouse is clicked, Mouse entered the component, Mouse exited the component, Mouse is moved, Mouse is dragged
- MouseEvent is a subclass of InputEvent.
- MouseListener and MouseMotionListener are used to process MouseEvent.

Constants:
There are eight types of mouse events, which are identified by these integer constants

- MOUSE_PRESSED: The mouse was pressed.
- MOUSE_RELEASED: The mouse was released.
- MOUSE_CLICKED: The user clicked the mouse.
- MOUSE_ENTERED: The mouse entered a component.
- MOUSE_EXITED: The mouse exited from a component.
- MOUSE_MOVED: The mouse moved.
- MOUSE_DRAGGED: The user dragged the mouse.
- MOUSE_WHEEL: The mouse wheel was moved.

Other constants defined by MouseEvent class:
NOBUTTON Button1 Button2 Button3

Atul Chaudhari

Atul Chaudhari

Scanned by TapScanner

TextEvent class

- An TextEvent is generated when the value of a text area or text field is changed
- **TextListener** is used to process TextEvent.

- **Constants:**

`TEXT_VALUE_CHANGED`

- **Constructor:**

`TextEvent(Object src, int type)`

Here,

`src:-` is the object that generates an event

`type:-` specifies the event

- **Methods:** The TextEvent object does not include the characters currently in the text component that generated the event. Instead, other methods associated with the text component can be used to retrieve that information.

(i) You're joined as an attendee. X

AdjustmentEvent class

- An AdjustmentEvent is generated when scrollbar value is manipulated
- **AdjustmentListener** is used to process AdjustmentEvent.

- **Constants:**

There are five types of adjustment events.

The AdjustmentEvent class defines integer constants that can be used to identify them.

BLOCK_DECREMENT: The user clicked inside the scrollbar to decrease its value.

BLOCK_INCREMENT: The user clicked inside the scroll bar to increase its value.

TRACK: The slider was dragged

UNIT_DECREMENT: The button at the end of the scrollbar was clicked to decrease its value.

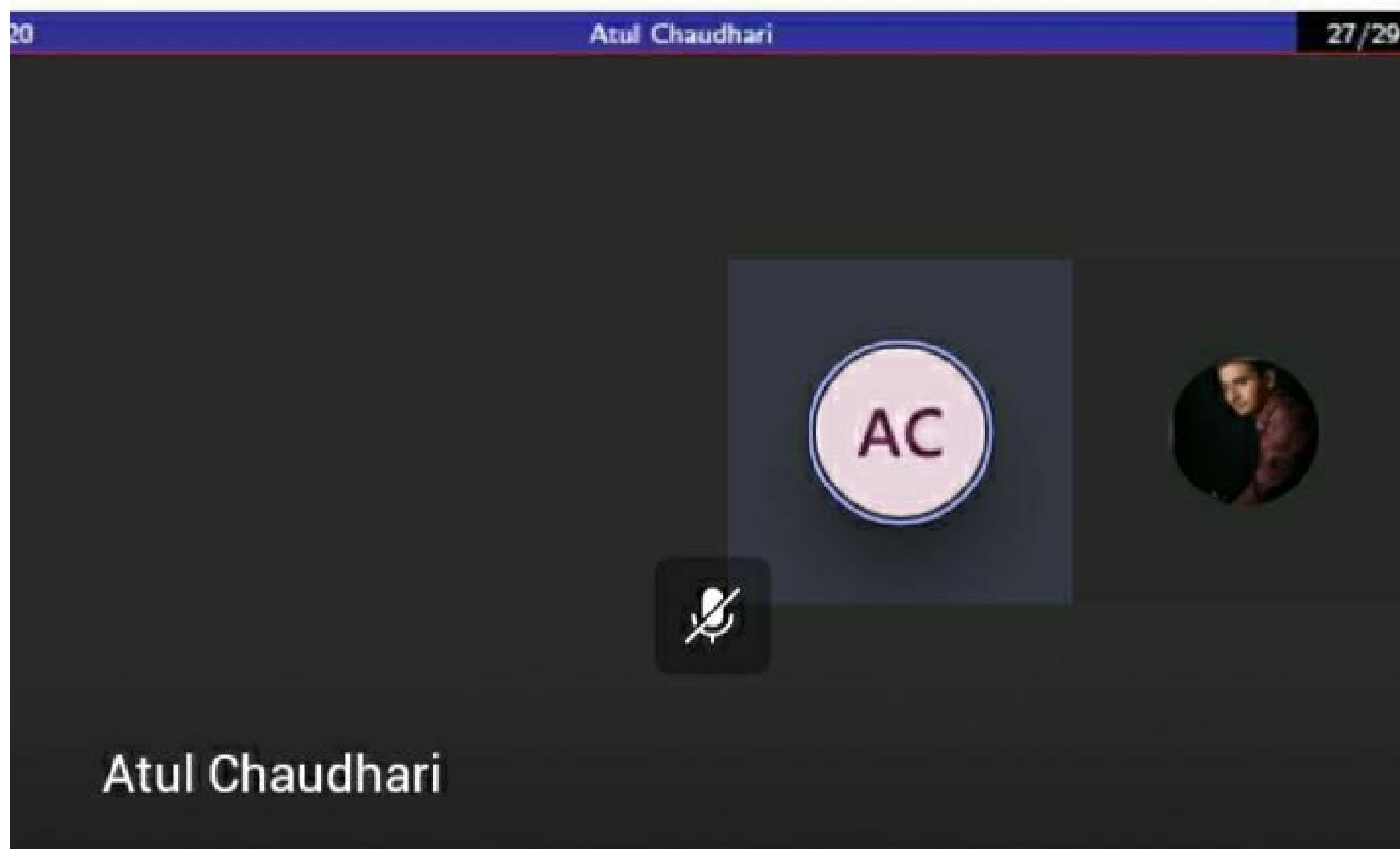
UNIT_INCREMENT: The button at the end of the scrollbar was clicked to increase its value.

ADJUSTMENT_VALUE_CHANGED: indicates that a change has occurred.

TextListener interface

- This interface defines the `textChanged()` method that is invoked when a change occurs in a text area or text field.
- **Syntax:**

```
public void textChanged(TextEvent ae)
```



(i) You're joined as an attendee. X

The screenshot shows a Java API documentation page for the `WindowListener` interface. The title bar says "WindowListener interface". The left sidebar contains navigation links for "Outline", "Delegation Event Model", "Event Classes" (listing various event classes like `FocusEvent`, `MouseEvent`, etc.), and "Event Sources" (listing various event sources like `Event Listener`, `Focus Listener`, etc.). The main content area starts with two bullet points: "This interface defines seven methods." and "Syntax:". Below the syntax, there is a code block:

```
public void windowOpened(WindowEvent we)
public void windowClosed(WindowEvent we)
public void windowClosing(WindowEvent we)
public void windowIconified(WindowEvent we)
public void windowDeiconified(WindowEvent we)
public void windowActivated(WindowEvent we)
public void windowDeactivated(WindowEvent we)
```

At the bottom of the slide, there is a footer with the date "November 9, 2020", the name "Atul Chaudhari", and the page number "45/48".



AJP: Unit-III Event Handli...

20:31 82 attendees



You're joined as an attendee.



WindowEvent class contd...

Methods:

- Window getWindow() - It returns the Window object that generated the event.
- Window getOppositeWindow() - returns the opposite window
- int getOldState()) - returns previous state of window
- int getNewState() - returns the current state of window

November 9, 2020 Atul Chaudhari 33/48



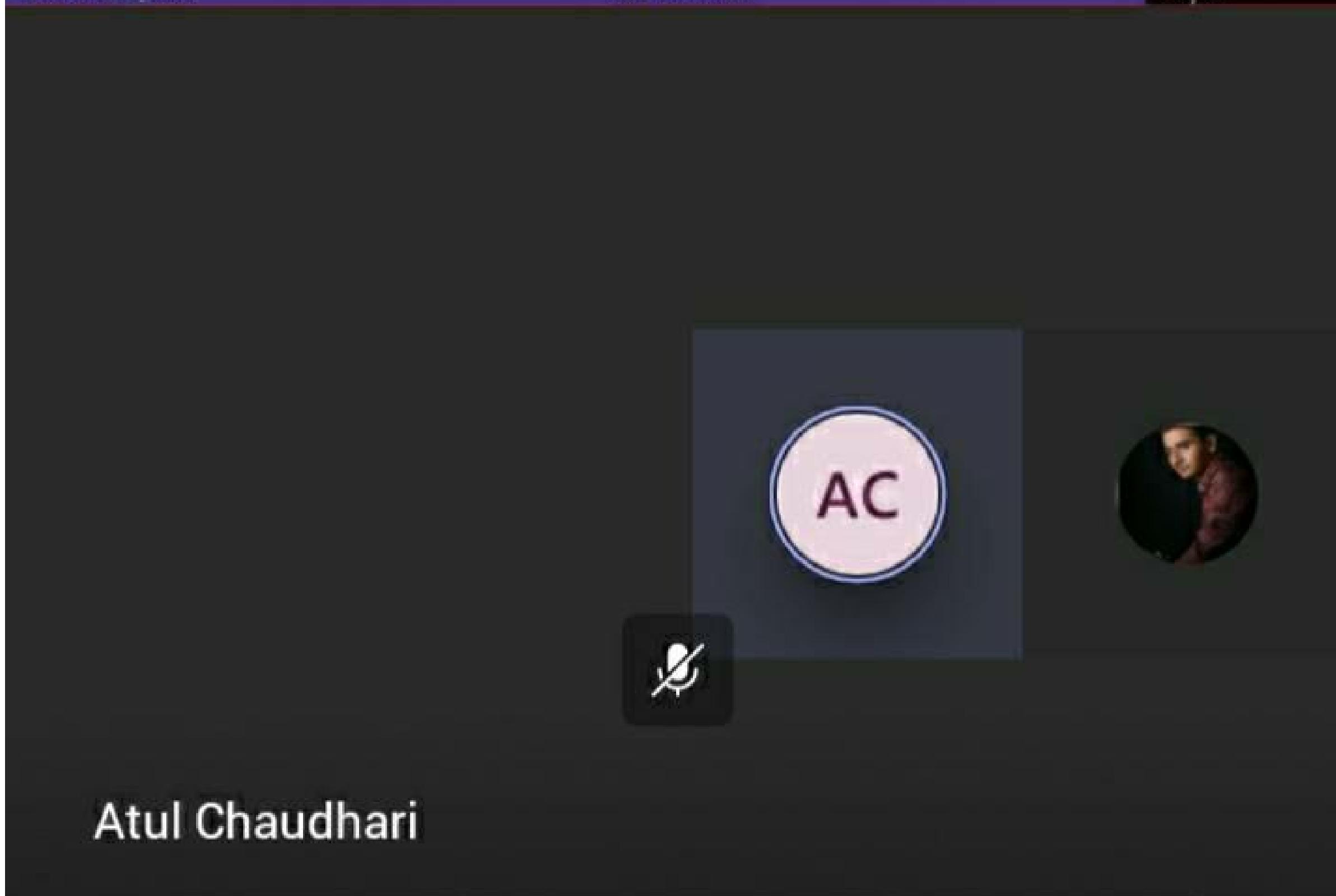
The screenshot shows a mobile application interface. At the top, there is a dark header bar with signal strength, time (10:56), battery level (38.3 KB/s), and network information (4G, 42). Below this is a blue navigation bar with a logo on the left and the title "AdjustmentListener interface". The main content area is white and contains the following text:

■ This interface defines the `adjustmentValueChanged()` method that is invoked when an adjustment event occurs.

■ Syntax:

```
public void adjustmentValueChanged(AdjustmentEvent ae)
```

At the bottom of the content area, there is a toolbar with icons for back, forward, search, and other navigation functions. The footer of the screen shows the date "November 1, 2020", the author's name "Atul Chaudhari", and the page number "26/29".

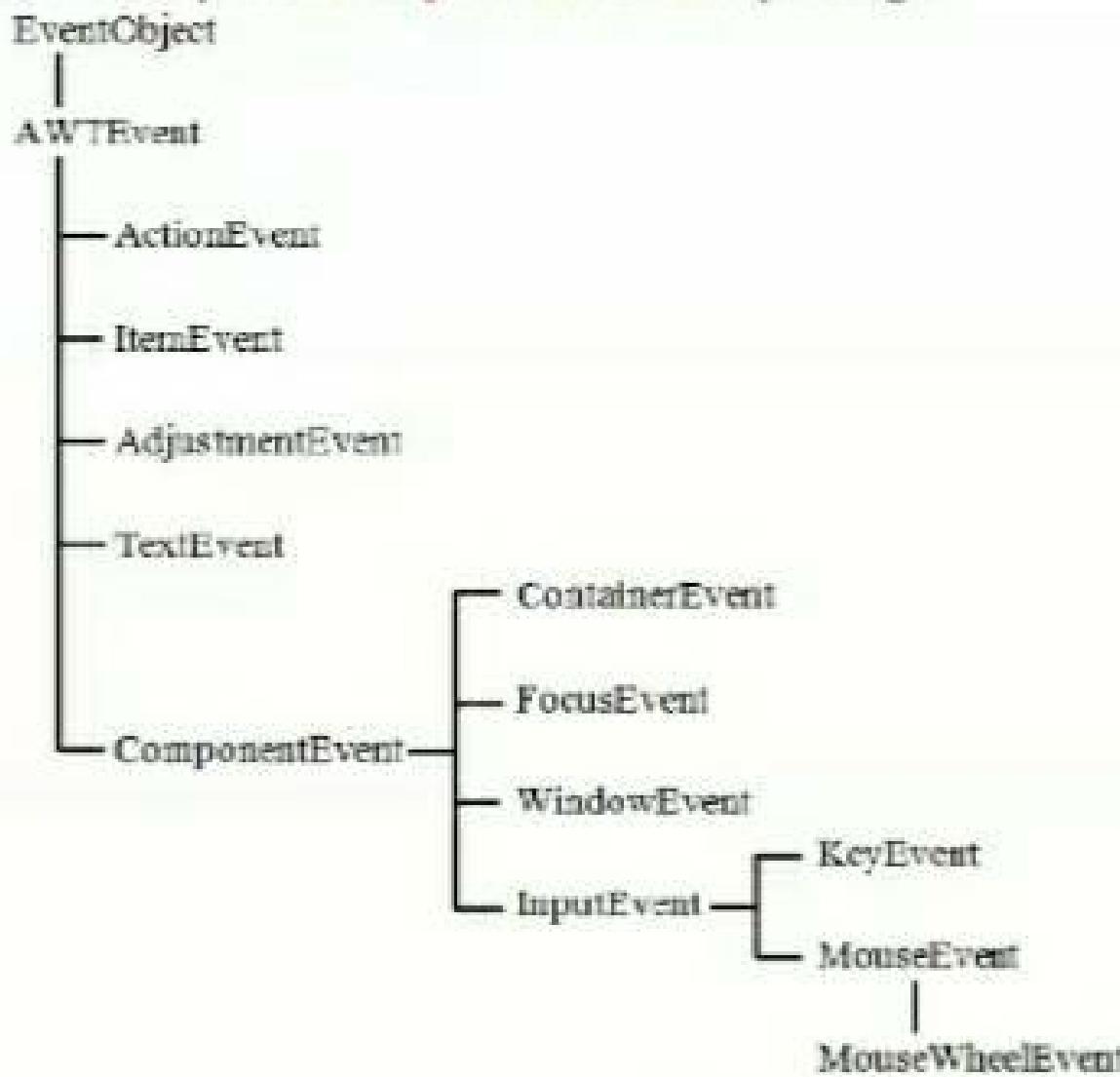


(i) You're joined as an attendee. X



Event Classes

- Event classes represents an event
- All event classes are present in `java.awt.event` package



Model
Classes
Events
Input
MouseEvent
TextEvent
AdjustmentEvent
ComponentEvent
ContainerEvent
FocusEvent
WindowEvent
InputEvent
MouseEvent
MouseWheelEvent
Listeners
Listeners
MouseListener
KeyListener
WindowListener
InputListener
MouseListener
KeyListener
WindowListener
InputListener
MouseListener
KeyListener
WindowListener
InputListener

ember 8, 2020

Atul Chaudhari

9/48

AC



Atul Chaudhari

Scanned by TapScanner

4G | 4G 09:11 42.0 KB/s

4G 62

AJP: Unit-III Event Handli...
18:23 81 attendees

(i) You're joined as an attendee. X

1 of 48 | - + 134% | Tools Sign Comment

A Presentation
On

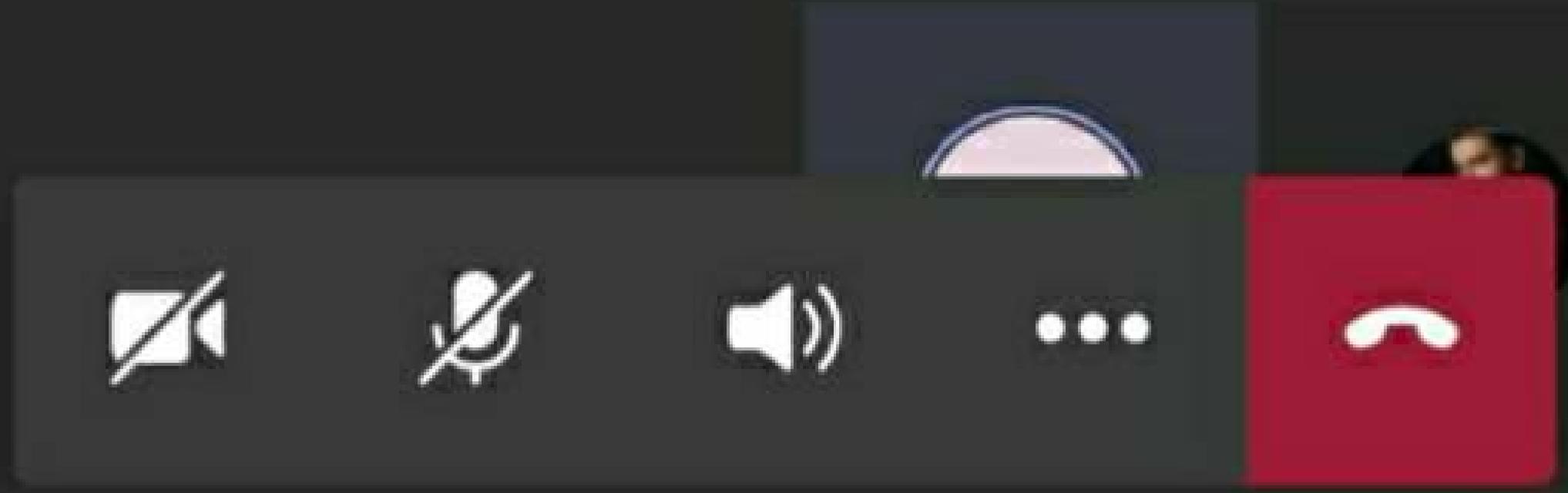
Advanced Java Programming

By
Atul S. Chaudhari
M.E.(Computer), C-DAC Pune

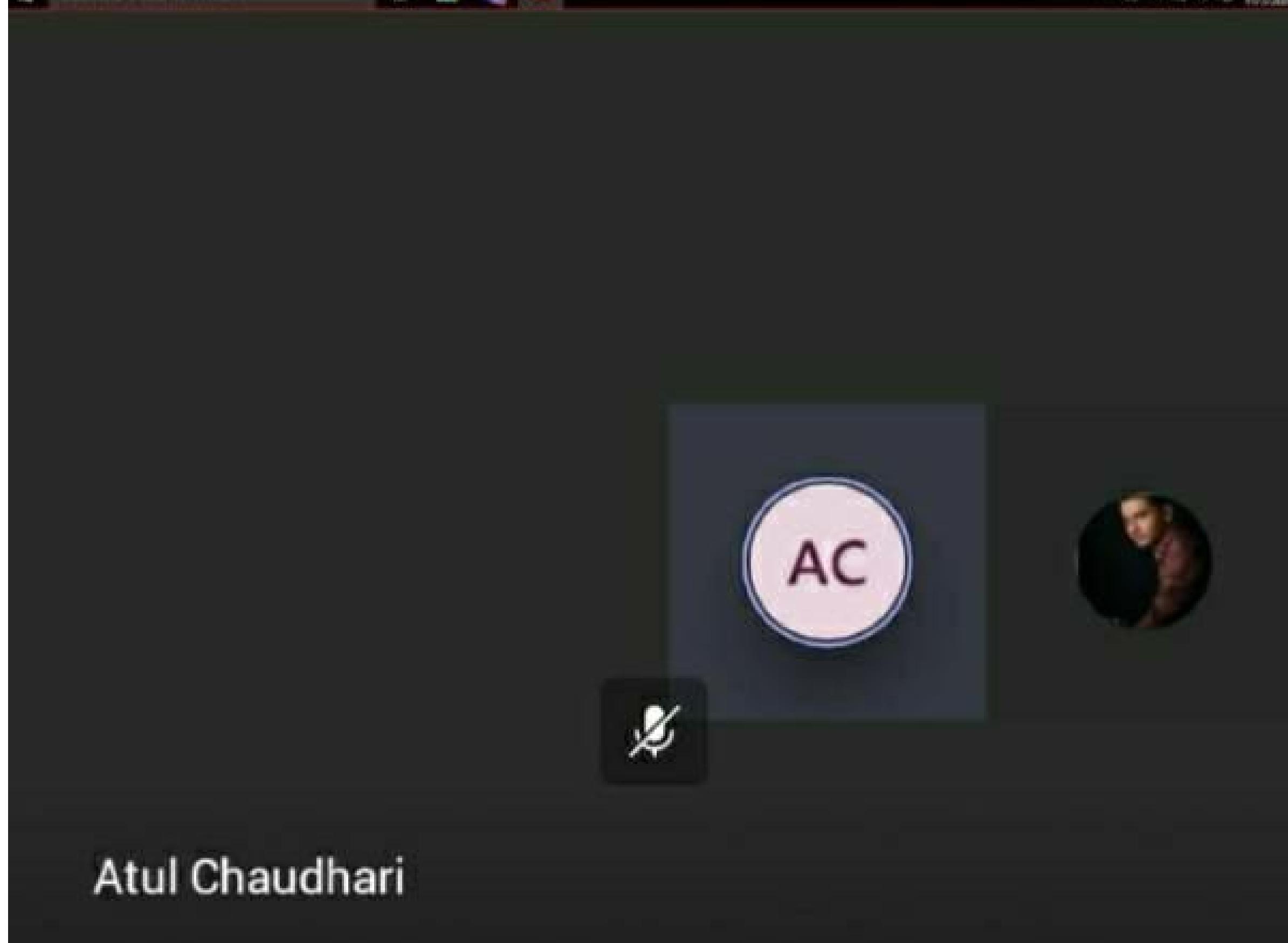
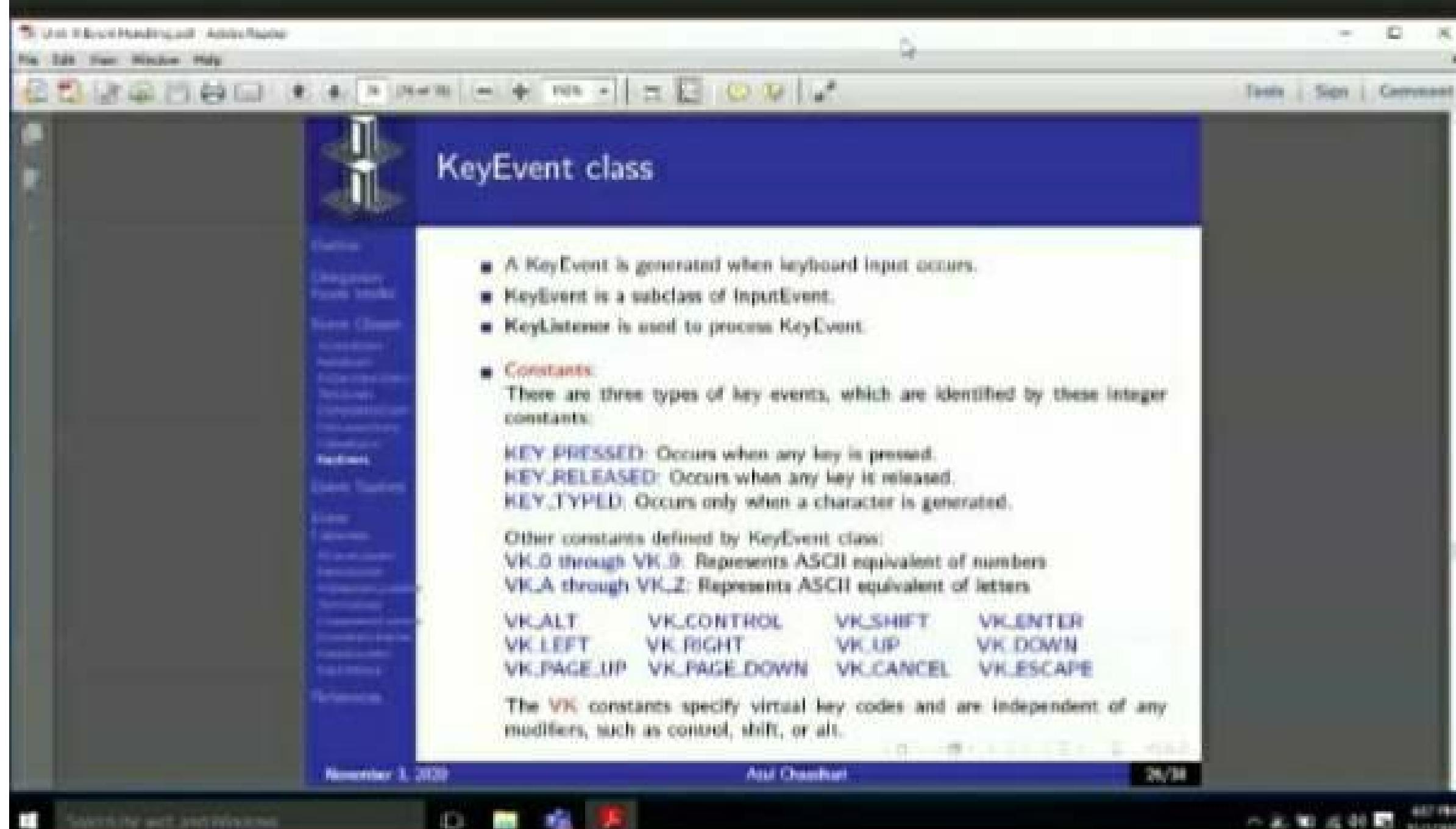
Department of Computer Engineering
S. S. V. P. S's B. S. Deore Polytechnic, Dhule

November 9, 2020 Atul Chaudhari 1/48

Atul Chaudhari



(i) You're joined as an attendee. X



(i) You're joined as an attendee. X

The screenshot shows a Java documentation page for the `WindowEvent` class. The title is "WindowEvent class". The page contains the following content:

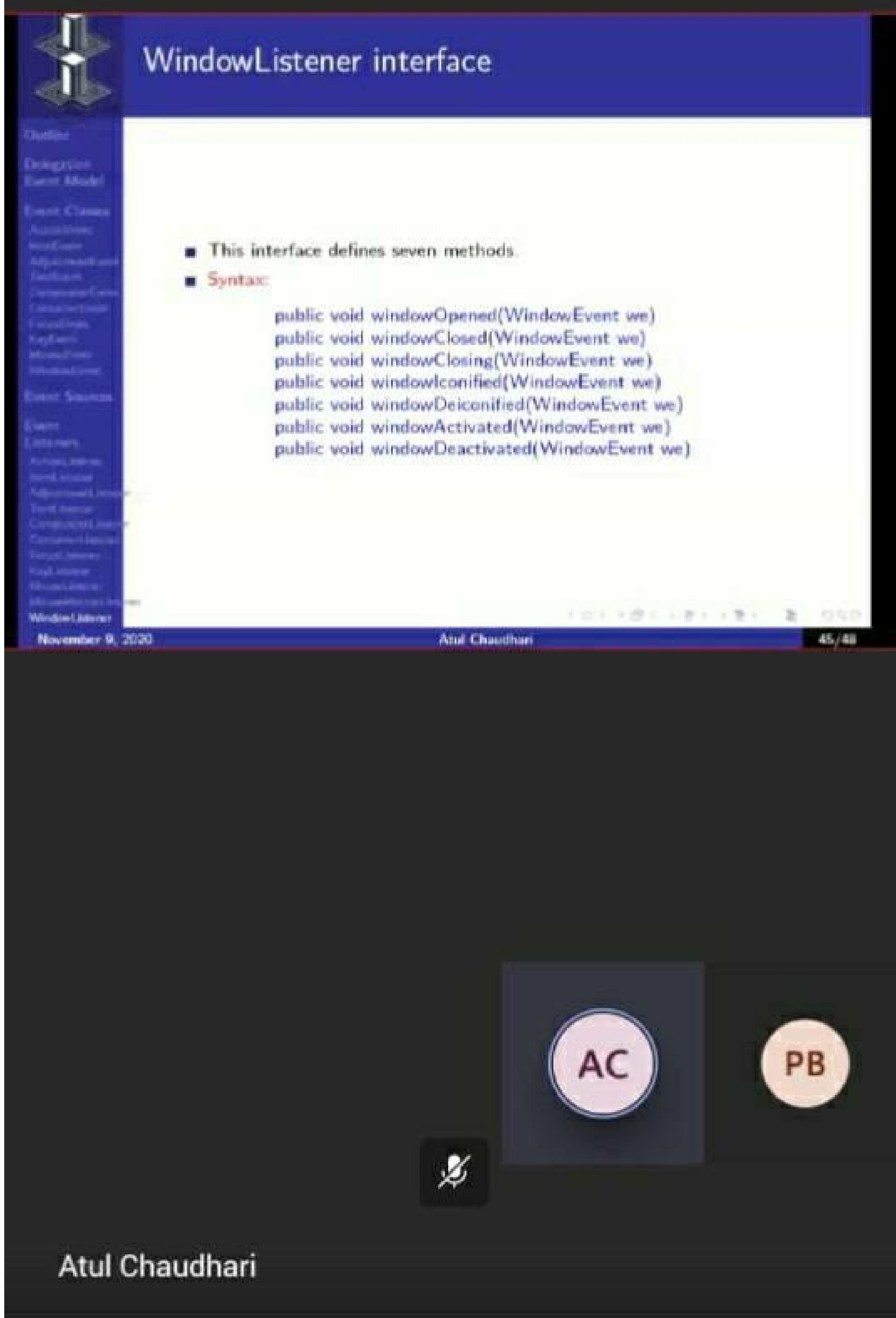
- A `WindowEvent` is generated when:
 - Window is Opened, Window is Closing, Window is Closed, Window is Activated, Window is Deactivated, Window is Iconified, Window is Deiconified.
 - The window gained input focus, The window lost input focus
- `WindowEvent` is a subclass of `ComponentEvent`.
- `WindowListener` and `WindowFocusListener` are used to process `WindowEvent`.

Constants:
The `WindowEvent` class defines integer constants that can be used to identify different types of window events.

- `WINDOW_OPENED`: The window was opened.
- `WINDOW_CLOSED`: The window has been closed.
- `WINDOW_CLOSING`: The user requested that the window be closed.
- `WINDOW_ICONIFIED`: The window was iconified.
- `WINDOW_DEICONIFIED`: The window was deiconified.
- `WINDOW_ACTIVATED`: The window was activated.
- `WINDOW_DEACTIVATED`: The window was deactivated.
- `WINDOW_GAINED_FOCUS`: The window gained input focus.
- `WINDOW_LOST_FOCUS`: The window lost input focus.
- `WINDOW_STATE_CHANGE`: The state of the window changed.

At the bottom of the slide, there is a small video player interface showing a thumbnail of a person and the text "Atul Chaudhari".

You've joined as an attendee.



Meeting in 'AJP'

04:30 81 attendees



ⓘ You've joined as an attendee. X

WindowEvent class contd...

■ Constructor:

```
WindowEvent(Window src, int type)
WindowEvent(Window src, int type, Window other)
WindowEvent(Window src, int type, int fromState, int toState)
WindowEvent(Window src, int type, Window other, int fromState, int toState)
```

Here,

src:- reference to the component that generated this event

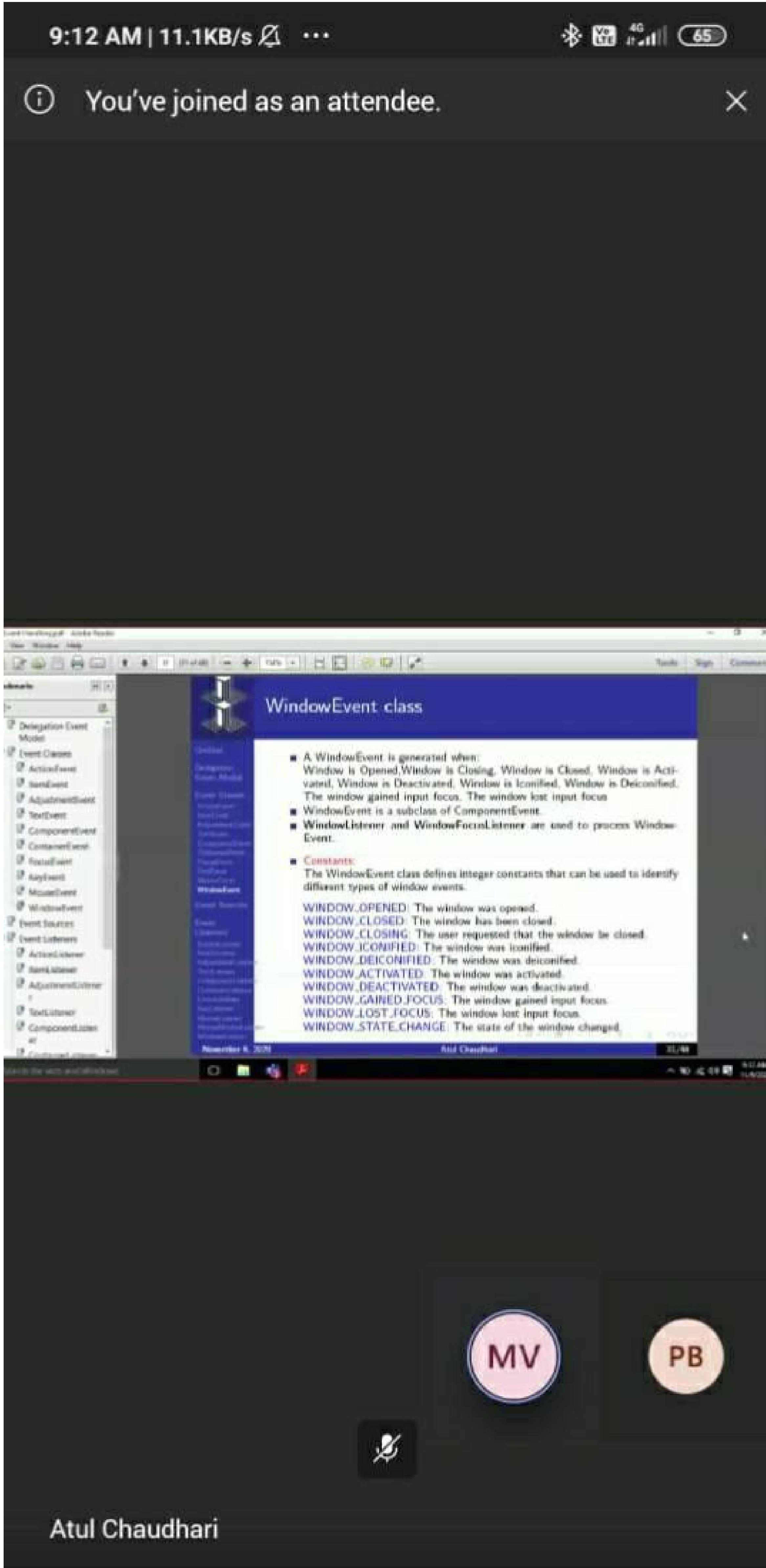
type:- indicates type of the event

other:- specifies the opposite window when a focus or activation event occurs

fromState:-specifies the prior state of the window

toState:- specifies the new state that the window will have when a window state change occurs

Atul Chaudhari



(i) You've joined as an attendee. X

AdjustmentEvent class

- An AdjustmentEvent is generated when scrollbar value is manipulated
- AdjustmentListener is used to process AdjustmentEvent.
- **Constants:**
There are five types of adjustment events.
The AdjustmentEvent class defines integer constants that can be used to identify them.
 - BLOCK_DECREMENT:** The user clicked inside the scrollbar to decrease its value.
 - BLOCK_INCREMENT:** The user clicked inside the scroll bar to increase its value.
 - TRACK:** The slider was dragged
 - UNIT_DECREMENT:** The button at the end of the scrollbar was clicked to decrease its value.
 - UNIT_INCREMENT:** The button at the end of the scrollbar was clicked to increase its value.
- **ADJUSTMENT_VALUE_CHANGED:** indicates that a change has occurred.

November 1, 2020

Atul Chaudhari

18/20

AC

PB



Atul Chaudhari



(i) You've joined as an attendee. X

The screenshot shows a presentation slide titled "TextEvent class". The slide content includes:

- An TextEvent is generated when the value of a text area or text field is changed.
- **TextListener** is used to process TextEvent.
- **Constants:**
TEXT_VALUE_CHANGED
- **Constructor:**
TextEvent(Object src, int type)
Here,
src:- is the object that generates an event
type:- specifies the event
- **Methods:** The TextEvent object does not include the characters currently in the text component that generated the event. Instead, other methods associated with the text component can be used to retrieve that information.

At the bottom of the slide, there are two circular icons labeled "AC" and "PB".

At the very bottom of the screen, the name "Atul Chaudhari" is visible.



(i) You've joined as an attendee. X

Delegation Event Model

- The modern approach to handle event is based on Delegation Event Model.
- It provides standard and consistent mechanism for generating and processing events.
- Concept:** A source generates an event and sends it to one or more listeners. In this scheme listener simply waits until it receives an event. Once received, the listener processes the event and returns.
- There are three main parts of delegation event model:
 - Event
 - Source
 - Listeners

October 26, 2018 Atul Chaudhari 4/28

Atul Chaudhari

i You've joined as an attendee. X

Events, Sources and Listeners

Event (Class)	Source (Class)	Listener (Interface)
ActionEvent	Button, MenuItem, List	ActionListener
ItemEvent	Choice, List, Checkbox, CheckboxMenuItem	ItemListener
AdjustmentEvent	Scrollbar	AdjustmentListener
TextEvent	TextField, TextArea	TextListener
ComponentEvent	Any Component	ComponentListener
ContainerEvent	Any Container (Applet, Frame, Panel)	ContainerListener
FocusEvent	Any Component	FocusListener
WindowEvent	Window (Frame)	WindowListener, WindowFocusListener
KeyEvent	Any Component	KeyListener
MouseEvent	Any Component	MouseListener, MouseMotionListener
MouseWheelEvent	Any Component	MouseWheelListener

October 25, 2020 Atul Chaudhari 8/26

Atul Chaudhari

(i) You've joined as an attendee. X

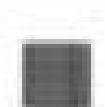
The screenshot shows a presentation slide titled "Event". The slide content includes a bulleted list and a diagram. The list defines an event as an object describing a state change in source, generated by user interaction with GUI elements like buttons, keyboards, mice, or lists. The diagram features a central dark blue square containing two white circles labeled "AC" and "PB". A small microphone icon is located at the bottom left of the slide. The slide has a navigation bar at the bottom with icons for back, forward, and search.

Event

- An event is an object that describes a state change in source.
- It can be generated when user interacts with elements in GUI.
- Some activity that causes event to be generated are:
pressing button,
entering character via k/b,
clicking mouse,
selecting item in list.

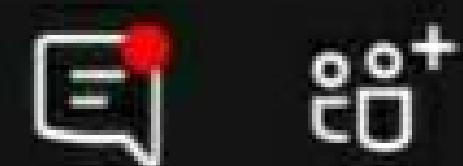
AC PB

Atul Chaudhari



Meeting in 'AJP'

11:24 78 attendees



ⓘ You've joined as an attendee. X

The screenshot shows a presentation slide titled "Delegation Event Model". The slide content includes:

- The modern approach to handle event is based on Delegation Event Model.
- It provides standard and consistent mechanism for generating and processing events.
- Concept:** A source generates an event and sends it to one or more listeners. In this scheme listener simply waits until it receives an event. Once received, the listener processes the event and returns.
- There are three main parts of delegation event model:
 - Event
 - Source
 - Listeners

At the bottom of the slide, there is a footer with the date "October 25, 2020", the name "Atul Chaudhari", and the page number "4/26".

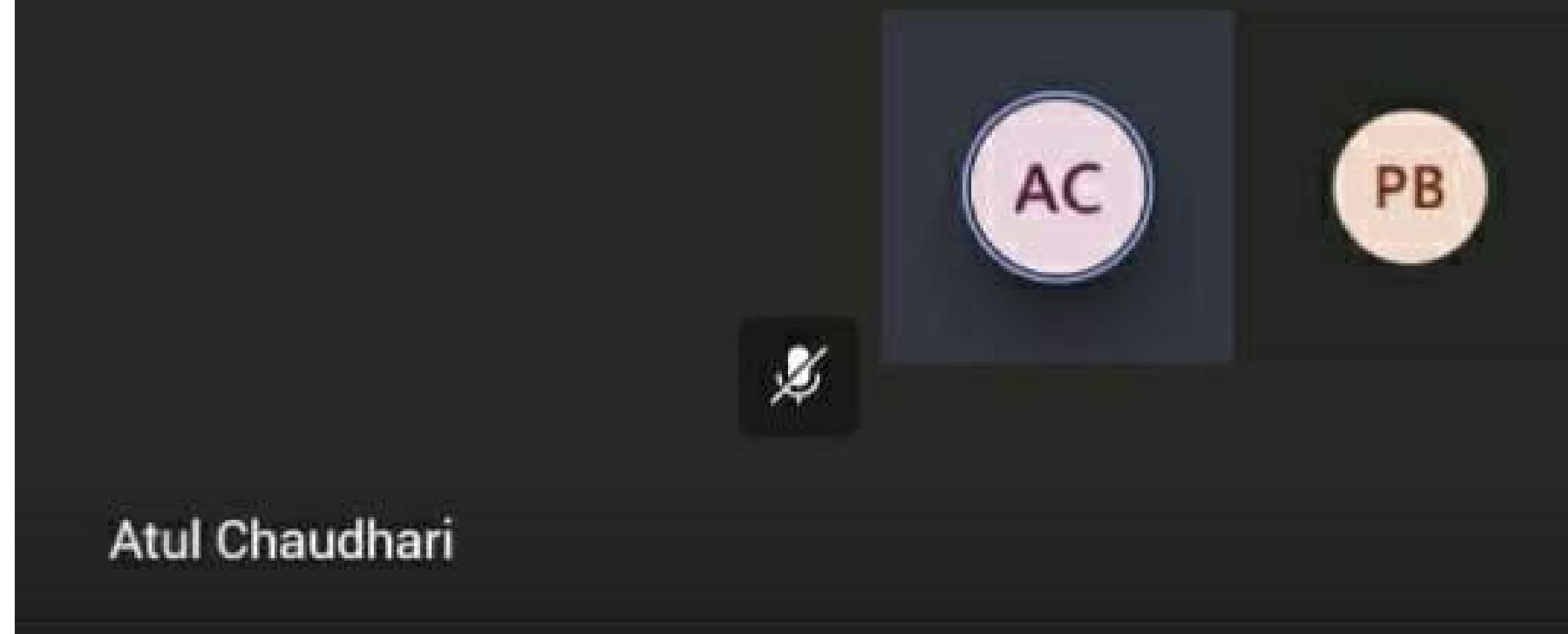
Below the slide, there is a dark interface with several icons: a video camera, a microphone, headphones, three dots, and a red button with a white phone receiver icon. The name "Atul Chaudhari" is also visible at the bottom of this interface.

(i) You've joined as an attendee. X

The screenshot shows a presentation slide titled "Listener". The slide content is as follows:

- A listener is the object that is notified when an event occurs.
- It has two major requirements:
 - ① It must have been registered with one or more event sources to receive notification about specific type of event.
 - ② It must implement methods to receive and process these notifications.

The slide is part of a larger presentation, indicated by the navigation icons at the bottom: back, forward, search, and others. The date "October 25, 2020" and the name "Atul Chaudhari" are visible at the bottom of the slide area.



AJP: Unit-III Event Handling

11:27 106 attendees



ActionEvent class contd...

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 the object that generates an event
- type**- indicates type of the event
- cmd**- contains command string
- when**- specifies when the event occurred
- modifiers**- indicates which modifier keys (ALT, CTRL, META, and/or SHIFT) were pressed when the event was generated

Methods:

- `String getActionCommand()` - Returns the command name for the invoking ActionEvent object.
For example, when a button is pressed, an action event is generated that has a command name equal to the label on that button.
- `long getWhen()` - Returns the time at which the event took place.
- `int getModifiers()` - Returns a value that indicates which modifier keys were pressed when the event was generated.



...



Atul Chaudhari

10:04 AM | 19.3KB/s

4G 92%

Meeting in 'AJP'

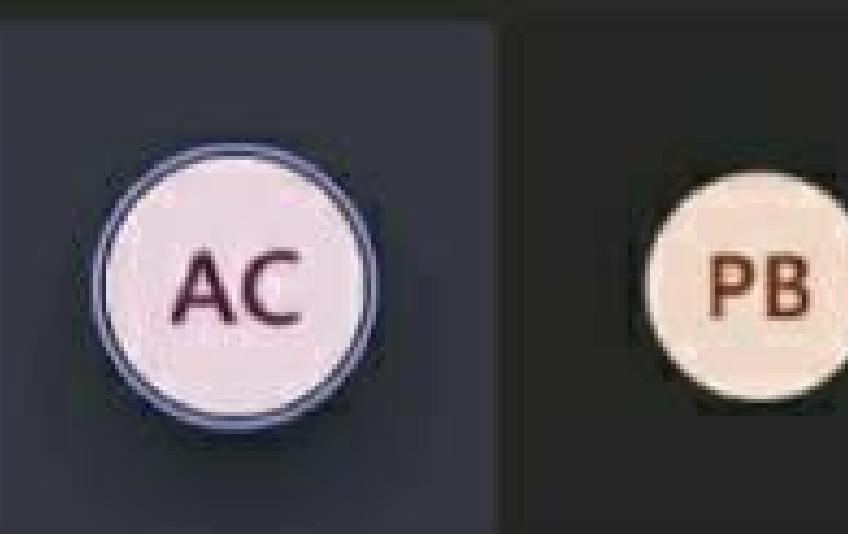
03:23 70 attendees



i You've joined as an attendee. X

The screenshot shows a PDF document titled "Outline of Topics". The table of contents includes:

- 1 Delegation Event Model
- 2 Event Classes
 - ActionEvent
 - BendEvent
 - AdjustmentEvent
 - TextEvent
 - ComponentEvent
 - ContainerEvent
- 3 Event Sources
- 4 Event Listeners
 - ActionListener
- 5 References



Atul Chaudhari

i You've joined as an attendee. X

Events, Sources and Listeners

Event (Class)	Source (Class)	Listener (Interface)
ActionEvent	Button, MenuItem, Link	ActionListener
ItemEvent	Choice, List, Checkbox, CheckbuttonItem	ItemListener
AppointmentEvent	Scrollbar	AppointmentListener
TextEvent	Textfield, TextArea	TextListener
ComponentEvent	Any Component	ComponentListener
ContainerEvent	Any Container (Applet, Frame, Panel)	ContainerListener
FocusedEvent	Any Component	FocusListener
WindowEvent	Window (Frame)	WindowListener, WindowFocusListener
KeyEvent	Any Component	KeyListener
MouseEvent	Any Component	MouseListener, MouseMotionListener
MouseWheelEvent	Any Component	MouseWheelListener

Atul Chaudhari

(i) You've joined as an attendee. X

Delegation Event Model

- The modern approach to handle event is based on Delegation Event Model.
- It provides standard and consistent mechanism for generating and processing events.
- **Concept:** A source generates an event and sends it to one or more listeners. In this scheme listener simply waits until it receives an event. Once received, the listener processes the event and returns.
- There are three main parts of delegation event model:
 - ① Event
 - ② Source
 - ③ Listeners

Atul Chaudhari

AC PB

Atul Chaudhari

i You've joined as an attendee. X

The screenshot shows a presentation slide titled "Source". The slide content is as follows:

- A source is an object that generates an event. This occurs when internal state of object changes some ways.
- Sources may generate more than one type of events.
- A source must register listeners in order for the listener to receive notification about specific type of event.
- When event occurs all the registered listeners are notified and receive copy of event object
- Each type of event has its own registration method. The general form of Method is:

```
public void addTypeListener(TypeListener el)
```

Here,
Type- Name of Event
el- reference to the event listener

Example:
Method that register keyboard event listener is called `addKeyListener()`

At the bottom of the slide, there are two circular icons labeled "AC" and "PB".

At the very bottom of the screen, the name "Atul Chaudhari" is visible.



The screenshot shows a presentation slide titled "Adapter Classes". The slide has a blue header bar with the title. Below the header is a sidebar containing navigation links for "Outline", "Delegation Event Model", "Event Classes" (which is currently selected), and "Event Sources". The main content area contains a bulleted list of points about adapter classes, followed by a table mapping adapter classes to their corresponding listener interfaces. At the bottom of the slide are navigation icons and the date "November 9, 2020".

Adapter Classes

Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent
WindowEvent
Event Sources
Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener
WindowListener

- Adapter classes are used to simplify the process of event handling.
- An adapter class provides an empty implementation of all methods in an event listener interface.
- Adapter classes are useful when you want to receive and process only some of the events that are handled by a particular event listener interface.
- To use adapter class-define a new class to act as an event listener by extending one of the adapter classes and implement only those events in which you are interested.

Adapter Class	Listener Interface
ComponentAdapter	ComponentListener
ContainerAdapter	ContainerListener
FocusAdapter	FocusListener
KeyAdapter	KeyListener
MouseAdapter	MouseListener
MouseMotionAdapter	MouseMotionListener
WindowAdapter	WindowListener

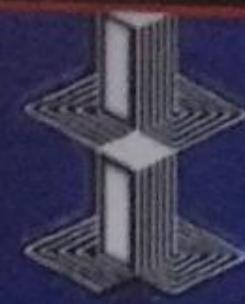
November 9, 2020 Atul Chaudhari 47/51

AC

AP

Atul Chaudhari

Request control



WindowFocusListener interface

Outline

Delegation Event Model

Event Classes

ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent
WindowEvent

Event Sources

Event Listeners

ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ConcurrentListener
FocusListener
KeyListener
MouseListener
MouseMotionListener
WindowListener

November 9, 2020

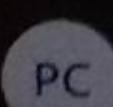
Atul Chaudhari

46/48

- This interface defines two methods.
- Syntax:

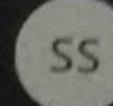
```
public void windowGainedFocus(WindowEvent we)  
public void windowLostFocus(WindowEvent we)
```

Atul Chaudhari

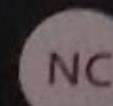


MAYURI VILAS VISPUTE

Atul Chaudhan



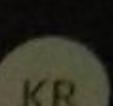
SAHIL GIRISH SALUNK...



NANDINI MILIND CH...



DIPESH NAresh MAN...



KHAIRNAR KALPESH ...

HD 1080

MouseEvent class contd...

■ Methods:

`int getX()`- return the X coordinates of the mouse within the component when the event occurred

`int getY()`- return the Y coordinates of the mouse within the component when the event occurred

`Point getPoint()`- used to obtain the coordinates of the mouse.

`int getClickCount()`- used to obtain the number of mouse clicks for this event.

`int getButton()`- It returns a value that represents the mouse button that caused the event.



Outline

Delegation Event Model

Event Classes

ActionEvent

ItemEvent

AdjustmentEvent

TextEvent

ComponentEvent

ContainerEvent

FocusEvent

KeyEvent

MouseEvent

WindowEvent

Event Sources

Event

Listeners

ActionListener

ItemListener

AdjustmentListener

TextListener

ComponentListener

ContainerListener

FocusListener

KeyListener

MouseListener

MouseMotionListener

WindowListener

November 8, 2020

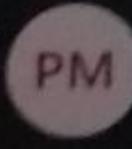
Atul Chaudhari

30 / 48

NM



RUGVED PRAMOD SUR...



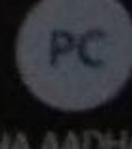
PANDYA PARESH MA...



MADHAVI DEVIDAS D...



MAYURI VILAS VISPUTE



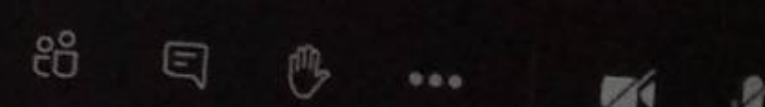
PRATIKSHA AADHAR ...



Atul Chaudhari

Meeting in "AJP"

Request control



Leave

MouseEvent class contd...

■ Constructor:

`MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)`

Here,

`src`:- reference to the component that generated this event

`type`:- indicates type of the event

`when`:- indicate system time at which the mouse event was occurred

`modifiers`:-indicates which modifiers were pressed when this mouse event occurred

`x`:- contains x coordinate of the mouse

`y`:-contains y coordinate of the mouse

`clicks`:-contains click count

`triggersPopup`:-flag indicates if this event causes a pop-up menu to appear on this platform

Atul Chaudhari

29/48

NM

RUGVED PRAMOD SUR...

PM



PANDYA PARESH MA...

MV

MADHAVI DEVIDAS D...

PC

MAYURI VILAS VISPUTE

AC

PRATIKSHA AADHAR ...

Atul Chaudhar...

AP



MouseEvent class

- A MouseEvent is generated when:
Mouse button is pressed, Mouse button is released, Mouse is clicked, Mouse entered the component, Mouse exited the component, Mouse is moved, Mouse is dragged
- MouseEvent is a subclass of InputEvent.
- MouseListener and MouseMotionListener are used to process MouseEvent.

■ Constants:

There are eight types of mouse events, which are identified by these integer constants:

MOUSE_PRESSED: The mouse was pressed.

MOUSE_RELEASED: The mouse was released

MOUSE_CLICKED: The user clicked the mouse.

MOUSE_ENTERED: The mouse entered a component.

MOUSE_EXITED: The mouse exited from a component.

MOUSE_MOVED: The mouse moved.

MOUSE_DRAGGED: The user dragged the mouse.

MOUSE_WHEEL: The mouse wheel was moved.

Other constants defined by MouseEvent class:

NOBUTTON Button1 Button2 Button3

Outline
Delegation
Event Model
Event Classes
KeyboardEvent
InputEvent
AdjustmentEvent
TextEvent
ComponentEvent
ColorSelectionEvent
FocusEvent
ImageEvent
MouseEvent
WindowEvent

Event Sources
Events
Listeners
ActionListener
BasicListener
MouseListener
MouseMotionListener
TextListener
ComponentAdapter
CompoundListener
FocusableListener
FontListener
DocumentListener
MouseEventListener

November 8, 2020

Atul Chaudhari

28/48



RUGVED PRAKASH SURANA



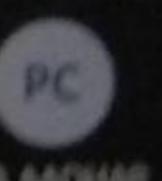
PANCHA PARESH MAHESHWARI



MADHUM DEVIDAS DABHADKAR



MAYURI VILAS VISPUTE



PRATIKSHA AADHAR



Atul Chaudhari



AP

11:48:38

HD 4G 31%

Unit-III Event Handling.pdf - Adobe Acrobat Reader DC

File Edit View Sign Window Help

Home Tools Unit-III Event Hand... Lab Manual_AJP_2...

28 (28 of 43) 145% Sign In

MouseEvent class

Outline Delegation Event Model

Event Classes ActionEvent ItemEvent AdjustmentEvent TextEvent ComponentEvent ContainerEvent FocusEvent KeyEvent MouseEvent

Event Sources Event Listeners ActionListener ItemListener AdjustmentListener TextListener ComponentListener ContainerListener FocusListener KeyListener MouseListener MouseMotionListener References

■ A MouseEvent is generated when:
Mouse button is pressed, Mouse button is released, Mouse is clicked, Mouse entered the component, Mouse exited the component, Mouse is moved, Mouse is dragged

■ MouseEvent is a subclass of InputEvent.

■ MouseListener and MouseMotionListener are used to process MouseEvent.

■ Constants:
There are eight types of mouse events, which are identified by these integer constants:

MOUSE_PRESSED: The mouse was pressed.
KEY_RELEASED: The mouse was released
KEY_CLICKED: The user clicked the mouse.
KEY_ENTERED: The mouse entered a component.
KEY_EXITED: The mouse exited from a component.
KEY_MOVED: The mouse moved.
KEY_DRAGGED: The user dragged the mouse.
KEY_WHEEL: The mouse wheel was moved.

Other constants defined by MouseEvent class:
NOBUTTON Button1 Button2 Button3

Atul Chaudhari November 3, 2020 Atul Chaudhari 28/43

Type here to search

Scanned by TapScanner



MouseEvent class contd...

Outline

Delegation Event Model

Event Classes

ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent

Event Sources

Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener

Atul Chaudhari

November 3, 2020

Atul Chaudhari

AC

AP

■ **Constructor:**

`MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)`

Here,

src:- reference to the component that generated this event

type:- indicates type of the event

when:- indicate system time at which the mouse event was occurred

modifiers:-indicates which modifiers were pressed when this mouse event occurred

x:- contains x coordinate of the mouse

y:-contains y coordinate of the mouse

clicks:-contains ~~contains~~ count

triggersPopup:-flag indicates if this event causes a pop-up menu to appear on this platform



Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent
Event Sources
Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener

MouseEvent class contd...

■ Methods:

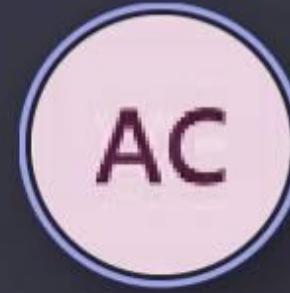
`int getX()`- return the X coordinates of the mouse within the component when the event occurred

`int getY()`- return the Y coordinates of the mouse within the component when the event occurred

`Point getPoint()`- used to obtain the coordinates of the mouse.

`int getClickCount()`- used to obtain the number of mouse clicks for this event.

`int getButton()`- It returns a value that represents the mouse button that caused the event.





Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
FocusEvent
KeyEvent
MouseEvent
Event Sources
Event
Listeners
ActionListener
ItemListener
AdjustmentListener
TextListener
ComponentListener
ContainerListener
FocusListener
KeyListener
MouseListener
MouseMotionListener

MouseEvent class contd...

Constructor:

MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)

Here,

src:- reference to the component that generated this event

type:- indicates type of the event

when:- indicate system time at which the mouse event was occurred

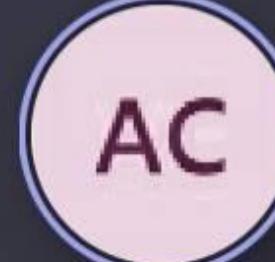
modifiers:-indicates which modifiers were pressed when this mouse event occurred

x:- contains x coordinate of the mouse

y:-contains y coordinate of the mouse

clicks:-contains count Contains clickcount

triggersPopup:-flag indicates if this event causes a pop-up menu to appear on this platform



10:53 AM

You've joined as an attendee

ActionEvent class

Outline Delegation Event Model Event Classes ActionEvent ItemEvent AdjustmentEvent TextEvent ComponentEvent ContainerEvent Event Sources Event Listeners ActionListener References

■ An ActionEvent is generated when:

- Button is pressed
- List item is double clicked
- MenuItem is selected

■ ActionListener is used to process ActionEvent.

■ Constants:
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
SHIFT_MASK

ACTION_PERFORMED: used to identify action events

EventObject
AWTEvent
ActionEvent

```
graph TD; EventObject --> AWTEvent; AWTEvent --> ActionEvent;
```

Atul Chaudhari

October 25, 2020

Atul Chaudhari

AC CS

11:00 AM 4G 84% 11:00 AM

Unit 10 Event Handling.pdf - Adobe Reader

You've joined as an attendee.

ActionListener interface

This interface defines the actionPerformed() method that is invoked when an action event occurs.

Syntax:

```
public void actionPerformed(ActionEvent ae)
```

Atul Chaudhari October 25, 2020 Atul Chaudhari 24/26

AC CS

A screenshot of an Adobe Acrobat Reader window displaying a PDF page about the ActionListener interface. The page title is "ActionListener interface". It contains text explaining that the interface defines the actionPerformed() method and provides its syntax. The PDF has a dark theme with a blue header bar. The status bar at the bottom shows the date (October 25, 2020), the author (Atul Chaudhari), and the page count (24/26). There are also two circular icons with initials (AC and CS) in the bottom right corner.

11:00 AM

4G 85

(i) You've joined as an attendee.

ActionEvent class contd...

Outline
Delegation
Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
Event Sources
Event
Listeners
ActionEvents
References

Atul Chaudhari

October 25, 2020

■ 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 the object that generates an event

`type`:- indicates type of the event

`cmd`:- contains command string

`when`:- specifies when the event occurred

`modifiers`:- indicates which modifier keys (ALT, CTRL, META, and/or SHIFT) were pressed when the event was generated

■ Methods:

`String getActionCommand()`- Returns the command name for the invoking ActionEvent object.

For example, when a button is pressed, an action event is generated that has a command name equal to the label on that button.

`long getWhen()`- Returns the time at which the event took place

`int getModifiers()`- Returns a value that indicates which modifier keys were pressed when the event was generated.



11:10

⌚ ⌂ ⌂ HD 4G 77%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

25 (25 of 27) 154% Tools Sign Comment

Bookmarks

- Delegation Event Model
- Event Classes
 - ActionEvent
 - ItemEvent
 - AdjustmentEvent
 - TextEvent
 - ComponentEvent
 - ContainerEvent
- Event Sources
- Event Listeners
 - ActionListener
 - ItemListener
- References

ItemListener interface

Outline

Delegation Event Model

Event Classes

ActionEvent

ItemEvent

AdjustmentEvent

TextEvent

ComponentEvent

ContainerEvent

Event Sources

ActionListener

ItemListener

References

October 28, 2020 Atul Chaudhari

This interface defines the itemStateChanged() method that is invoked when an item event occurs.

Syntax:

```
public void itemStateChanged(ItemEvent ie)
```

Atul Chaudhari

Search the web and Windows

AC AP

11:06

⌚ ⌂ ⌂ HD 4G 78%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

24 (24 of 26) 154% Tools Sign Comment

Bookmarks

- Delegation Event Model
- Event Classes
 - ActionEvent
 - ItemEvent
 - AdjustmentEvent
 - TextEvent
 - ComponentEvent
 - ContainerEvent
- Event Sources
- Event Listeners
 - ActionListener
- References

ActionListener interface

Outline

Delegation Event Model

Event Classes

ActionEvent

ItemEvent

AdjustmentEvent

TextEvent

ComponentEvent

ContainerEvent

Event Sources

Event Listeners

ActionListener

References

October 25, 2020 Atul Chaudhari

This interface defines the actionPerformed() method that is invoked when an action event occurs.

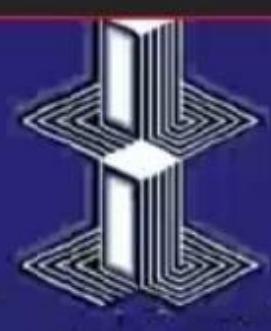
Syntax:

```
public void actionPerformed(ActionEvent ae)
```

Atul Chaudhari

Search the web and Windows

AC AP



Outline
Delegation Event Model
Event Classes
ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent
Event Sources
Event Listeners
ActionListener
References

ItemEvent class

- An ItemEvent is generated when:
 - Checkbox is clicked
 - List item is clicked
 - Choice selection is made
 - Checkable menu item is selected or deselected
- **ItemListener** is used to process ItemEvent.

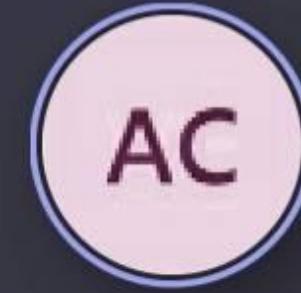
- **Constants:**

There are two types of item events, which are identified by the following integer constants:

SELECTED: The user selected an item.

DESELECTED: The user deselected an item.

ITEM_STATE_CHANGED: Signifies a change of state





AJP: Unit-II Event Handling

24:26 105 attendees



The slide title is "ItemEvent class contd...". It shows a UML class diagram with `ItemSelectable` as the superclass, and `ItemEvent` as a subclass. `ItemEvent` has methods `getItemSelectable()`, `getItem()`, `getState()`, and `getStateChange()`. A callout points to the constructor `ItemEvent(ItemSelectable src, int type, Object entry, int state)`.

Constructor:

```
ItemEvent(ItemSelectable src, int type, Object entry, int state)
```

Here,
src:- is the object that generates an event
type:- indicates type of the event
entry:- contains the specific item that generated the item event
state:- represents current state of item that generated the event

Methods:

- `ItemSelectable getItemSelectable()` - used to obtain a reference to the ItemSelectable object that generated an event.
- `Object getItem()` - used to obtain a reference to the item that generated an event

For example, if checkbox is selected or deselected it returns label associated with checkbox that generated event.

`int getStateChange()` - method returns the state change (that is, SELECTED or DESELECTED) for the event.

October 25, 2020 Atul Chaudhari 15/26



Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

23 (23 of 26) 154% Tools Sign Comment

Event Listeners

Outline Delegation Event Model

Event Classes ActionEvent ItemEvent AdjustmentEvent TextEvent ComponentEvent ContainerEvent

Event Sources

Event Listeners ActionListener

References

October 25, 2020 Atul Chaudhari 23/26

■ Listeners are created by implementing one or more of the interfaces defined by the java.awt.event package.

■ When an event occurs, the event source invokes the appropriate method defined by the listener and provides an event object as its argument.

Windows Search the web and Windows 10:57 AM 10/27/2020



ActionEvent class contd...

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 the object that generates an event
type:- indicates type of the event
cmd:- contains command string
when:- specifies when the event occurred
modifiers:- indicates which modifier keys (ALT, CTRL, META, and/or SHIFT) were pressed when the event was generated

Methods:

- String getActionCommand() - Returns the command name for the invoking ActionEvent object.
For example, when a button is pressed, an action event is generated that has a command name equal to the label on that button.
- long getWhen() - Returns the time at which the event took place
- int getModifiers() - Returns a value that indicates which modifier keys were pressed when the event was generated.

October 25, 2020 Atul Chaudhari 13/26

AC

AP

Atul Chaudhari



Outline

Delegation
Event Model

Event Classes

ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent

Event Sources

Event
Listeners
ActionListener

References

AWTEvent class

- It is subclass of EventObject defined in java.awt package.
- It is superclass (directly or indirectly) of all AWT-based events used by delegation event model.

- Constructors:

AWTEvent(Object src, int type)

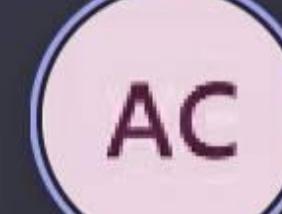
Here,

src:- is the object that generates an event

type:- indicates type of the event

- Methods:

int getID()-It is used to determine the type of event





Outline

Delegation
Event Model**Event Classes**ActionEvent
ItemEvent
AdjustmentEvent
TextEvent
ComponentEvent
ContainerEvent**Event Sources**Event
Listeners
ActionListener
References

EventObject class

- It is root of the Java event class hierarchy.
- It is present in java.util package
- It is superclass for all event classes

- **Constructors:**

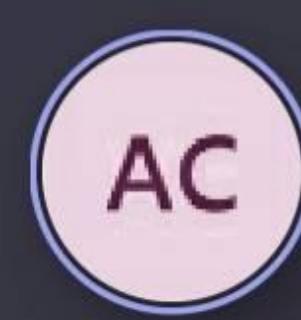
EventObject(Object src)

Here,

src:- is the object that generates an event

- **Methods:**

Object getSource()-This method returns source of the event.



11:12

HD 4G 66%

Unit-III Event Handling.pdf - Adobe Reader

File Edit View Window Help

Tools Sign Comment

Listener

Outline

Delegation Event Model

Event Classes

- ActionEvent
- ItemEvent
- AdjustmentEvent
- TextEvent
- ComponentEvent
- ContainerEvent

Event Sources

Event

Listeners

ActionListener

References

October 25, 2020 Atul Chaudhari

■ A listener is the object that is notified when an event occurs.

■ It has two major requirements:

- ① It must have been registered with one or more event sources to receive notification about specific type of event.
- ② It must implement methods to receive and process these notifications.

Atul Chaudhari

Search the web and Windows

AC AP