AWT Control: Scrollbar

- **Scrollbar control** represents a scroll bar component in order to enable *user to select from range of values*.

- Used to *select continuous values between a specified minimum and maximum*.

- Scroll bars may be oriented ***horizontally*** *or* ***vertically***.

- Each end has an arrow that you can click to move the current value of the scroll bar one unit in the direction of the arrow.

- **The current value of the scroll bar** relative to its minimum and maximum values is indicated by the ***slider box (or thumb)*** *for the scroll bar.*

AWT Control: Scrollbar

- **Constructors:**
  - Scrollbar( ) : //construct new vertical scrollbar
  - Scrollbar(int *style*) *//construct new scrollbar with style orientation*
  - Scrollbar(int *style*, *int initialValue*, *int thumbSize*, *int min*, *int max)*

Where style :

- ***Scrollbar.VERTICAL* or**
- ***Scrollbar.HORIZONTAL***

- ***For set Values:***
  - void setValues(int *initialValue*, *int thumbSize*, *int min*, *int max)*

- ***For get and set current value:***
  - int getValue( )
  - void setValue(int *newValue)*

AWT Control: Scrollbar

- ***For get Min and Max value:***
  - int getMinimum( )
  - int getMaximum( )

- **By default unit increment/decrement is 1** and Block page-up and page-down increment/decrement is **10.**

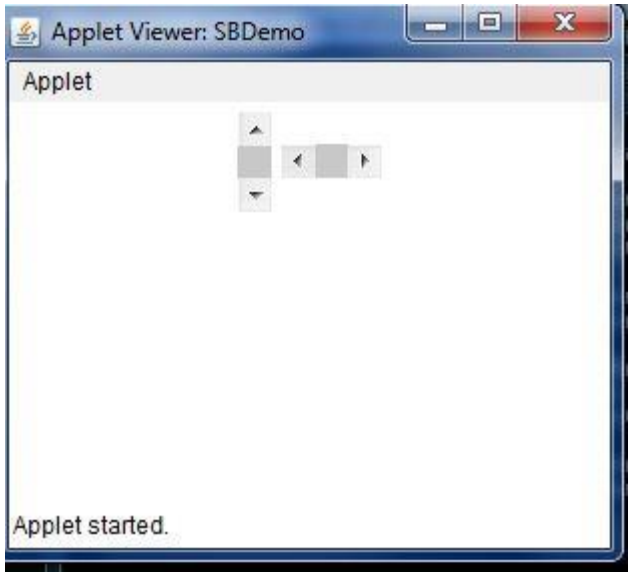
- ***For change increment and decrement:***
  - void setUnitIncrement(int *newIncr)*
  - void setBlockIncrement(int *newIncr)*

AWT Control: Scrollbar (Example)

```java
//Demonstrate scroll bars.
import java.awt.*;
import java.applet.*;
/*
<applet code="SBDemo" width=300 height=200>
</applet>
*/
public class SBDemo extends Applet
{
    Scrollbar vertSB, horzSB;
    public void init()
    {
        vertSB=new Scrollbar(Scrollbar.VERTICAL, 0, 1, 0, 10);

        horzSB=new Scrollbar(Scrollbar.HORIZONTAL,0, 1, 0, 10);

        add(vertSB);
        add(horzSB);
    }
}
```



AWT Control: Handling Scroll bars

- **AdjustmentEvent is generated.**
- Implement the **AdjustmentListener interface.**
- adjustmentValueChanged() method we have to override
- **getAdjustmentType( )** method can be used to determine the type of the adjustment.

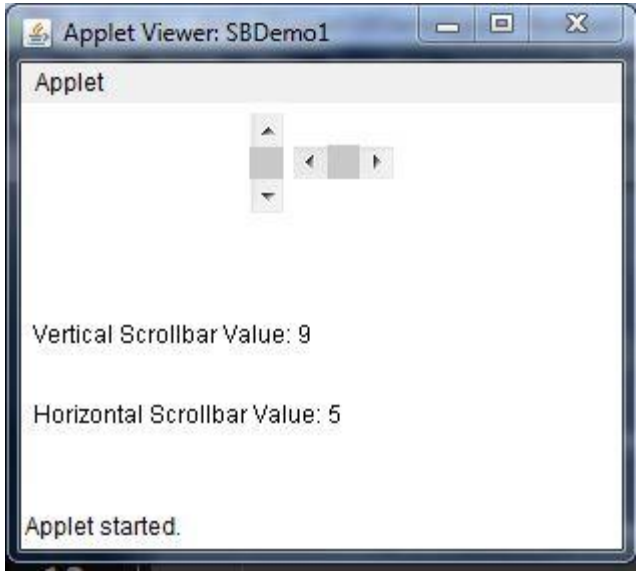
BLOCK_DECREMENT	A page-down event has been generated.
BLOCK_INCREMENT	A page-up event has been generated.
TRACK	An absolute tracking event has been generated.
UNIT_DECREMENT	The line-down button in a scroll bar has been pressed.
UNIT_INCREMENT	The line-up button in a scroll bar has been pressed.

AWT Control: Handling Scroll bars

```java
//Demonstrate scroll bars WITH EVENT HANDLING.
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="SBDemo1" width=300 height=200></applet>*/
public class SBDemo1 extends Applet implements AdjustmentListener
{
    Scrollbar vertSB, horzSB;
    String msg="";
    public void init()
    {
        vertSB=new Scrollbar(Scrollbar.VERTICAL, 0, 1, 0, 10);
        horzSB=new Scrollbar(Scrollbar.HORIZONTAL,0, 1, 0, 10);

        add(vertSB);
        add(horzSB);

        vertSB.addAdjustmentListener(this);
        horzSB.addAdjustmentListener(this);
    }
    public void adjustmentValueChanged(AdjustmentEvent ae)
    {
        repaint();
    }
    public void paint(Graphics g)
    {
        msg="Current Scrollbar values are: ";
        g.drawString("Vertical Scrollbar Value: "+vertSB.getValue(),6,120);
        g.drawString("Horizontal Scrollbar Value: "+horzSB.getValue(),6,160);
    }
}
```



AWT Control: TextField

- **TextField** is *subclass* of **TextComponent.**
  TextComponent is *subclass* of **Component.**

- **TextField** class implements a single-line text- entry area, usually called an ***edit control.***

- Text fields allow the user to enter strings and to edit the text using the arrow keys, cut and paste keys, and mouse selections.

- **Constructors:**
  - TextField( )
  - TextField(**int *numChars***)
  - TextField(**String *str***)
  - TextField(**String *str, int numChars***)

AWT Control: TextField

- **Setter and Getter Method of TextField and TextComponent:**
  - String getText( )
  - void setText(String *str)*

- **Particular Text selection:**
  - String getSelectedText( )
  - void select(int *startIndex*, *int endIndex)*

- **About Modification of Text:**
  - boolean isEditable( )
  - void setEditable(boolean *canEdit)*

AWT Control: TextField

- **Setting echo character to text field and related methods:**
  - void setEchoChar(char *ch)*
  - boolean echoCharIsSet( )
  - char getEchoChar( )

- Button can be used to Handling Event: **ActionEvent** generates.

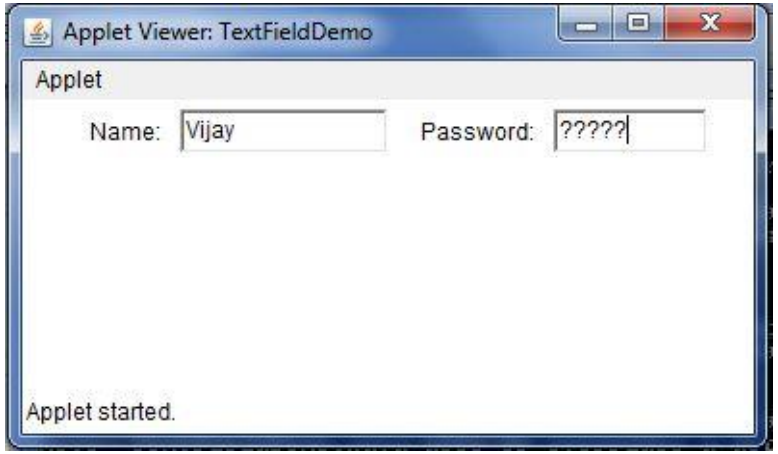
- implements **ActionListener** Class

AWT Control: TextField (Example)

```java
//Demonstrate TextField
import java.awt.*;
import java.applet.*;
/*
<applet code="TextFieldDemo" width=380 height=150>
</applet>
*/
public class TextFieldDemo extends Applet
{
    TextField name, pass;
    public void init()
    {
        Label namep=new Label("Name: ",Label.RIGHT);
        Label passp=new Label("Password: ",Label.RIGHT);
        name=new TextField(12);
        pass=new TextField(8);

        pass.setEchoChar('?');

        add(namep);
        add(name);
        add(passp);
        add(pass);
    }
}
```



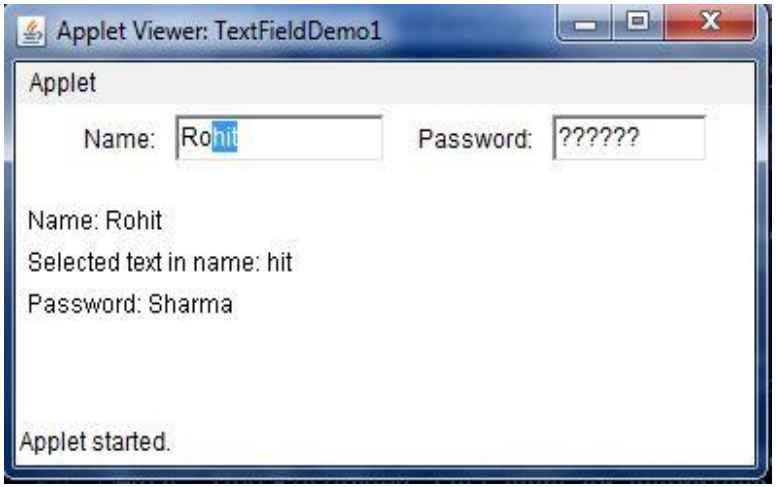
AWT Control: TextField Event Handling (Example)

```java
//Demonstrate TextField with event handling
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
/*<applet code="TextFieldDemo1" width=380 height=150>
</applet>
*/
public class TextFieldDemo1 extends Applet implements ActionListener
{
    TextField name, pass;
    public void init()
    {
        Label namep=new Label("Name: ",Label.RIGHT);
        Label passp=new Label("Password: ",Label.RIGHT);
        name=new TextField(12);
        pass=new TextField(8);

        pass.setEchoChar('?');

        add(namep);
        add(name);
        add(passp);
        add(pass);

        //register to receive action events
        name.addActionListener(this);
        pass.addActionListener(this);
    }
    //user presses enter
    public void actionPerformed(ActionEvent ae)
    {
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString("Name: "+name.getText(),6,60);
        g.drawString("Selected text in name: "+name.getSelectedText(), 6, 80);
        g.drawString("Password: "+pass.getText(),6,100);
    }
}
```



AWT Control: TextArea

- Need ? Sometimes a single line of text input is not enough for a given task.

- subclass of ***TextComponent.***

- **Constructors:**
  - **TextArea( )**
  - **TextArea(int *numLines, int numChars*)**

    -Here, ***numLines*** specifies the **height, in lines, of the text area**, and ***numChars*** specifies **its width, in characters**. Initial text can be specified by *str*.
  - **TextArea(String *str*)**
  - **TextArea(String *str, int numLines, int numChars*)**
  - **TextArea(String *str, int numLines, int numChars, int sBars*)**

AWT Control: TextArea

- **The values of sbar:**
  - ***SCROLLBARS_BOTH***
  - ***SCROLLBARS_NONE***
  - ***SCROLLBARS_HORIZONTAL_ONLY***
  - ***SCROLLBARS_VERTICAL_ONLY***

- It supports: **getText( ), setText( ), getSelectedText( ), select( ), isEditable( ), and setEditable( )**

- **Other some methods:**
  - void **append**(String *str)*
  - void **insert**(String *str, int index)*
  - void **replaceRange**(String *str, int startIndex, int endIndex)*

AWT Control: TextArea (Example)

```java
//Demonstrate TextArea.
import java.awt.*;
import java.applet.*;
/*
<applet code="TextAreaDemo" width=300 height=250>
</applet>
*/
public class TextAreaDemo extends Applet
{
    public void init()
    {
        String val=
        "Java 8 is the latest version of the most\n" +
        "widely-used computer language for Internet programming.\n" +
        "Building on a rich heritage, Java has advanced both\n" +
        "the art and science of computer language design.\n\n" +
        "One of the reasons for Java's ongoing success is its\n" +
        "constant, steady rate of evolution. Java has never stood\n" +
        "still. Instead, Java has consistently adapted to the\n" +
        "rapidly changing landscape of the networked world.\n" +
        "Moreover, Java has often led the way, charting the\n" +
        "course for others to follow.";

        TextArea ta=new TextArea(val,10,30);
        add(ta);
    }
}
```