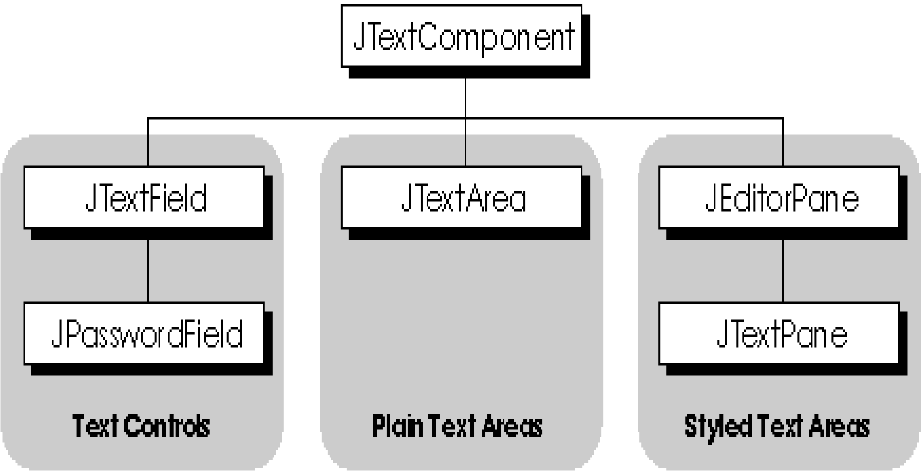
Text-entry Components:

Text components display text and optionally allow the user to edit the text. Programs need text components for tasks ranging from the straightforward (enter a word and press Return).

Swing provides five text components, along with supporting classes and interfaces, that meet even the most complex text requirements. In spite of their different uses and capabilities, all of Swing's text components inherit from the same superclass, JTextComponent API, which provides a highly-configurable and powerful foundation for text manipulation.

The following figure shows the JTextComponent hierarchy and places each text component class into one of three groups:



The following paragraphs describe the three groups of text components.

Text Controls:

Text Controls known simply as text fields, text controls can display and edit only one line of text and are action-based, like buttons. Use them to get a small amount of textual information from the user and take some action after the text entry is complete.

JTextfield:

A Textfield is a component used for displaying, inputting and editing a single line of plain text. We can create text field by creating an instance of JTextField class. The JTextComponent is a superclass of JTextField that provides common set of methods used by JTextfield.

Commonly used Constructors:

1. **JTextField():** This constructor is used to construct an empty Text field.
   Example

```
JTextField txt=new JTextField();
```

2. **JTextField(int column):** This constructor is used to construct an empty Text field with given number of columns.

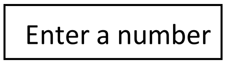
   Example

   ```
   JTextField txt=new JTextField(20);
   ```

3. **JTextField(String text):** This constructor is used to construct a Text field initialized with specific text.

   Example

   ```
   JTextField txt=new JTextField("Enter number");
   ```

   Enter a number

4. **JTextField(String text, int column):** This constructor is used to construct a Text field initialized with specific text and columns.

   Example

   ```
   JTextField txt=new JTextField("Enter a name",20);
   ```

   Enter a name

Methods of the JTextField class:

1. **setText(String text):** This method is used to sets a String message on the JTextField.

   Syntax:

   ```
   JTextFieldObject.setText(String text);
   ```

   Example:

   ```
   JTextField txt=new JTextField();
   txt.setText("Enter Number");
   ```

2. **getText():** This method is used to gets a String message of JTextField.

   Example:

   ```
   JTextField t1=new JTextField();
   t1.setText("Welcome to Swing");

   String str = t1.getText();
   JTextField t2=new JTextField();
   t2.setText(str);
   ```

3. **setEditable(boolean b)**: This method is used to sets a JTextfield to editable or uneditable.

   Example:

   ```
   txt.setEditable(false);
   ```

4. **setFont(Font f):** This method is used to sets a font type to the JTextField

   Example:

   ```
   txt.setFont(new Font("Serif", Font.BOLD, 10));
   ```

5. **setForeground(Color c):** This method is used to sets a foreground color, i.e. color of text in JTextField.

   Example:

   ```
   txt.setForeground(Color.RED);
   ```

6. **setColumns(int n)**: This method is used to set the number of columns of the text field.

   Example:

   ```
   txt.setColumns(15)
   ```

7. **int getColumns()**: This method is used to get the number of columns in the textfield.

   Example:

   ```
   int  n= textFieldObject.getColumns();
   ```

8. **addActionListener(ActionListener l)**: set an ActionListener to the text field.

Example

```
import javax.awt.*;
import javax.swing.*;
class TextEx
{
    public static void main(String args[])
    {
        JFrame f= new JFrame("Text Field Example");
        JTextField txt1,txt2;

        txt1=new JTextField();
        txt1.setText(“Enter 1st Value”);
        txt1.setBounds(50,50, 100,30);

        txt2=new JTextField();
        txt2.setText(“Enter 2nd Value”);
        txt2.setBounds(50,50, 100,30);
```

```
        f.add(txt1);
        f.add(txt2);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```