

Text Fields and Text Areas

A text field is an object of the class **JTextField** and is displayed as a field that allows the user to enter a single line of text.

Syntax:

```
private JTextField name;  
...  
name = new JTextField(NUMBER_OF_CHAR);
```

Here,

NUMBER_OF_CHAR : It is least NUMBER_OF_CHAR characters to be visible in textfield.

The classes **JTextField** and **JTextArea** both contain methods called **getText** and **setText**.

- The method **getText** can be used to retrieve the text written in the text field or text area.
- The method **setText** can be used to change the text written in the text field or text area.

Syntax

Name_of_Text_Component.getText() returns the text currently displayed in the text field or text area.

Name_of_Text_Component.setText(New_String_To_Display);

Examples

```
String inputString = ioComponent.getText();  
ioComponent.setText("Hello out there!");
```

ioComponent may be an instance of either of the classes **JTextField** or **JTextArea**.

Constructors Constructor and Description

- [JTextField\(\)](#)
Constructs a new TextField.

- [JTextField](#)(Document doc, String text, int columns)
Constructs a new JTextField that uses the given text storage model and the given number of columns.
- [JTextField](#)(int columns)
Constructs a new empty TextField with the specified number of columns.
- [JTextField](#)([String](#) text)
Constructs a new TextField initialized with the specified text.
- [JTextField](#)([String](#) text, int columns)
Constructs a new TextField initialized with the specified text and columns.

Some Methods in the Class JTextComponent

All these methods are inherited by the classes **JTextField** and **JTextArea**.

The abstract class **JTextComponent** is in the package **javax.swing.text**. The classes **JTextField** and **JTextArea** are in the package **javax.swing**.

- **public String getText()**

Returns the text that is displayed by this text component.

- **public boolean isEditable()**

Returns true if the user can write in this text component. Returns false if the user is not allowed to write in this text component.

- **public void setBackground(Color theColor)**

Sets the background color of this text component.

- **public void setEditable(boolean argument)**

If argument is true, then the user is allowed to write in the text component. If argument is false, then the user is not allowed to write in the text component.

- **public void setText(String text)**

Sets the text that is displayed by this text component to be the specified text.

//Sample Code 1

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class TextFieldDemo extends JFrame implements ActionListener
{
    public static final int WIDTH=600;
    public static final int HEIGHT=500;
    public static final int MAX=30;

    private JTextField colorText;

    private JPanel BluePanel;
    private JPanel WhitePanel;
    private JPanel GrayPanel;

    public static void main(String[] args)
    {
        TextFieldDemo pc=new TextFieldDemo();
        pc.setVisible(true);
    }

    public TextFieldDemo()
    {
        super("Text Fields");
        setSize(WIDTH,HEIGHT);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        JPanel BigPanel=new JPanel();
        BigPanel.setLayout(new GridLayout(1,3));

        BluePanel=new JPanel();
```

```
BluePanel.setBackground(Color.LIGHT_GRAY);
BigPanel.add(BluePanel);
```

```
WhitePanel=new JPanel();
WhitePanel.setBackground(Color.LIGHT_GRAY);
BigPanel.add(WhitePanel);
```

```
GrayPanel=new JPanel();
GrayPanel.setBackground(Color.LIGHT_GRAY);
BigPanel.add(GrayPanel);
```

```
add(BigPanel,BorderLayout.CENTER);
```

```
JPanel ButtonPanel=new JPanel();
ButtonPanel.setLayout(new FlowLayout());
```

```
JButton changeButton=new JButton("Change");
changeButton.addActionListener(this);
ButtonPanel.add(changeButton);
```

```
JButton resetButton=new JButton("Reset");
resetButton.addActionListener(this);
ButtonPanel.add(resetButton);
```

```
JButton closeButton=new JButton("Close");
closeButton.addActionListener(this);
ButtonPanel.add(closeButton);
ButtonPanel.setBackground(Color.LIGHT_GRAY);
add(ButtonPanel,BorderLayout.SOUTH);
```

```
JPanel TextPanel=new JPanel();
TextPanel.setLayout(new FlowLayout());
```

```
colorText=new JTextField(MAX);
JLabel colorLabel=new JLabel("Enter Color-Blue/Gray/White");
TextPanel.setBackground(Color.LIGHT_GRAY);
TextPanel.add(colorLabel);
TextPanel.add(colorText);
```

```

        add(TextPanel,BorderLayout.NORTH);

    }

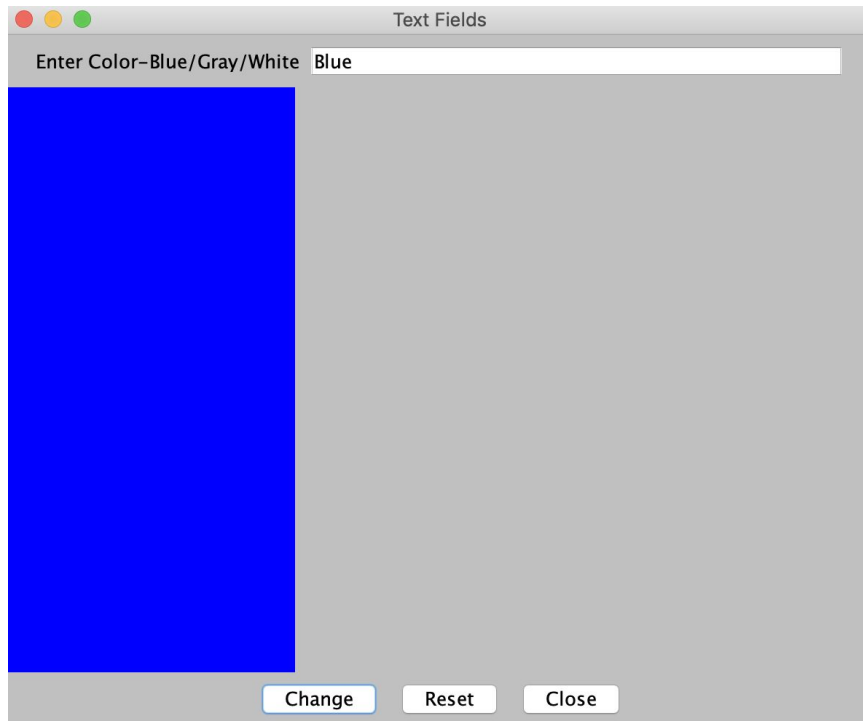
    public void actionPerformed(ActionEvent e)
    {
        String buttonString=e.getActionCommand();

        if(buttonString.equals("Change"))
        {
            String c=colorText.getText();
            if(c.equals("Blue"))
                BluePanel.setBackground(Color.BLUE);
            else if(c.equals("White"))
                WhitePanel.setBackground(Color.WHITE);
            else if(c.equals("Gray"))
                GrayPanel.setBackground(Color.GRAY);
            else
                System.out.println("Error");
        }
        else if(buttonString.equals("Close"))
            System.exit(0);
        else if (buttonString.equals("Reset"))
        {
            colorText.setText("");
            BluePanel.setBackground(Color.LIGHT_GRAY);
            WhitePanel.setBackground(Color.LIGHT_GRAY);
            GrayPanel.setBackground(Color.LIGHT_GRAY);
        }
    }

}

```

//Sample Output



//Calculator

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Calculator extends JFrame implements ActionListener
{
    public static final int WIDTH=600;
    public static final int HEIGHT=500;
    public static final int MAX=30;

    private JTextField Text1;
    private JTextField Text2;
    private JTextField TextResult;

    public static void main(String[] args)
    {
        Calculator pc=new Calculator();
        pc.setVisible(true);
    }
}
```

```
}
```

```
public Calculator()
```

```
{
```

```
    super("Calculator");  
    setSize(WIDTH,HEIGHT);  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setLayout(new BorderLayout());
```

```
    JPanel ButtonPanel=new JPanel();  
    ButtonPanel.setLayout(new GridLayout(3,2));
```

```
    JButton addButton=new JButton("+");  
    addButton.addActionListener(this);  
    ButtonPanel.add(addButton);
```

```
    JButton subButton=new JButton("-");  
    subButton.addActionListener(this);  
    ButtonPanel.add(subButton);
```

```
    JButton prodButton=new JButton("*");  
    prodButton.addActionListener(this);  
    ButtonPanel.add(prodButton);
```

```
    JButton divButton=new JButton("/");  
    divButton.addActionListener(this);  
    ButtonPanel.add(divButton);
```

```
    JButton resetButton=new JButton("Reset");  
    resetButton.addActionListener(this);  
    ButtonPanel.add(resetButton);
```

```
    JButton closeButton=new JButton("Close");  
    closeButton.addActionListener(this);  
    ButtonPanel.add(closeButton);
```

```
    ButtonPanel.setBackground(Color.LIGHT_GRAY);  
    add(ButtonPanel,BorderLayout.CENTER);
```

```
JPanel TextPanel=new JPanel();
TextPanel.setLayout(new GridLayout(3,2));
```

```
Text1=new JTextField(MAX);
Text2=new JTextField(MAX);
JLabel text1Label=new JLabel("Enter first number");
JLabel text2Label=new JLabel("Enter second number");
TextResult=new JTextField(MAX);
TextResult.setEditable(false);//Cannot be edited
JLabel resultLabel=new JLabel("Result");
TextPanel.setBackground(Color.LIGHT_GRAY);
TextPanel.add(text1Label);
TextPanel.add(Text1);
TextPanel.add(text2Label);
TextPanel.add(Text2);
TextPanel.add(resultLabel);
TextPanel.add(TextResult);
```

```
add(TextPanel,BorderLayout.NORTH);
```

```
}
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
    try
```

```
    {
```

```
        assumingCorrectNumberFormats(e);
```

```
    }
```

```
    catch(NumberFormatException e2)
```

```
    {
```

```
        Text1.setText("Error - Re enter numbers");
```

```
        Text2.setText("Error - Re enter numbers");
```

```
    }
```

```
}
```

```
//Throws Numnber Format Exception
```

```
public void assumingCorrectNumberFormats(ActionEvent e)
```

```
{
```

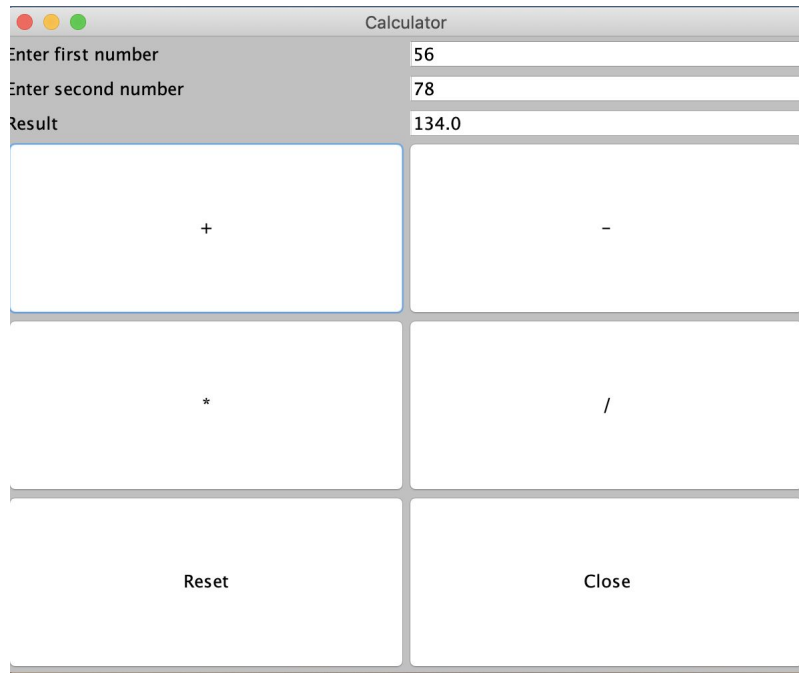


```

String buttonString=e.getActionCommand();
double num1=Double.parseDouble(Text1.getText());
double num2=Double.parseDouble(Text2.getText());
double result;
if(buttonString.equals("+"))
{
    result=num1+num2;
    TextResult.setText(Double.toString(result));
}
else if(buttonString.equals("-"))
{
    result=num1-num2;
    TextResult.setText(Double.toString(result));
}
else if(buttonString.equals("*"))
{
    result=num1*num2;
    TextResult.setText(Double.toString(result));
}
else if(buttonString.equals("/"))
{
    result=num1/num2;
    TextResult.setText(Double.toString(result));
}
else if(buttonString.equals("Reset"))
{
    TextResult.setText(" ");
    Text1.setText(" ");
    Text2.setText(" ");
}
else if(buttonString.equals("Close"))
    System.exit(0);
}
}

```

//Sample Output



In-Class Exercise:

Create a GUI with two text boxes for input and one textbox for result. Add Buttons to perform the following operations:

- 1. Concat two strings**
- 2. Replace the occurrence of String2 in String1 with any other text of your choice.**
- 3. Reset Button**
- 4. Close Button**