**GUI16 Personal Info**

**Directions**

The next graphical component we will look at is the textarea. A textarea is a multi-line text entry field similar to a textfield. It can be used for both entering text and displaying text. In our programs we will use it to display text. In Java, a textarea is created using the class **JTextArea**. Look at the following program that displays text in a textarea when a button is pressed.

Copy the following code to your source file. Compile and execute program.

import java.util.\*;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class GUI16 extends JFrame implements ActionListener

{

// instance variables

private JButton button;

private

JTextArea textArea

;

// constructor

public GUI16()

{

// create button

button = new JButton("run");

// set button attributes

button.setLocation(220,20);

button.setSize(80, 25);

// add button to frame

getContentPane().add(button);

// register listener with button

button.addActionListener(this);

// create textarea

textArea = new JTextArea();

// set textarea attributes

textArea.setLocation(50,50);

textArea.setSize(400, 400);

textArea.setFont(new Font("Arial", Font.PLAIN, 16));

textArea.setBorder(BorderFactory.createLineBorder (Color.blue, 2));

textArea.setEditable(false);

// cannot type text into textArea

// add textarea to frame

getContentPane().add(textArea);

// set frame attributes

setLayout(null);

setSize(500, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

}

public void actionPerformed(ActionEvent event)

{

if(event.getSource() == button) // identify source

{

String text = "";

for(int i = 0; i < 20; i++)

{

text += "Programming GUI is fun!!!

\n

";

}

textArea.setText("");

// clear text area

textArea.append(text);

// append text to text area

}

}

// main method

public static void main(String[] args)

{

GUI16 app = new GUI16(); // run program

}

}

Once again, a **JTextArea** is added to a frame using the same process we used to add the other graphical components. I am only going to highlight the code that is different.

textArea.setBorder(BorderFactory.createLineBorder (Color.blue, 2));

textArea.setEditable(false); // cannot type text into textArea

The first line of code above places a blue border around the textarea with a line thickness of 2. The **BorderFactory**class has other methods for making different kinds of borders. I am using the method createLineBorder to create a simply line border.

The second line of code is used to lock the textarea so that the user cannot type text into textarea window. Since we are only using the textarea to display text it is a good idea to lock the textarea for editing.

for(int i = 0; i < 20; i++)

{

text += "Programming GUI is fun!!! \n";

}

The code above builds a string that includes 20 lines that say "Programming GUI is fun!!!". Each "Programming GUI is fun!!!" is on a separate line. To accomplish this I added a \n to the end of the line. \n is called the new line escape sequence and it will cause an eoln (end-of-line) marker to be place at that location in the string. When the textarea encounters the eoln marker it will go to the beginning of the next line. This has the same effect as using a System.out.println command in console programming.

An escape sequence can be embedded any where within a string. Look at the following example:

String s = "\n\nMy dog's name is Maddie.\nMy cat's name is Cubbie.\n");

It can also be append to the end of a string. Look at the following example:

String s1 = "My dog's name is Maddie.";

String s2 = s1 + "\n";

The two lines below are used to add text to the textarea. The method setText will replace the text in the textarea with the text inside the quotes. I am using setText to clear the textarea window.

The method append appends the text in the quotes to any text that is already in the textarea.

textArea.setText(""); // clear text area

textArea.append(text); // append text to text area

**setText** - replaces the current text in the textarea with the text in the parentheses.  
**append** - adds the text in the parentheses to the end of the text that is currently in the textarea.

**Modifications**

1. Modify the **actionPerformed** method so that when the button is clicked it will display your first name, last name, address, city, state, and zip code on separate lines within the textarea window.
2. Include a **for loop** so that your information will be displayed three times with a blank line between each section.

**Source File**

GUI16.java

**Sample Run**

