Graphical User Interfaces

Simple GUI components and functionality

COMP2603
Object Oriented Programming 1

Week 6

Outline

- APIs
- Graphical User Interfaces
- Basic GUI Components
- ActionListeners

Reading: Chapter 17 - Introduction to Graphical User Interfaces : P. Mohan 2013
Online resources: https://docs.oracle.com/javase/tutorial/uiswing/

Graphical User Interface (GUI)

A graphical user interface (GUI) is a human-computer interface that is used by humans to interact with computers.

It uses windows, icons and menus to present and collect data and to facilitate interaction with a user.

Java APIs

There are two sets of Java APIs (Application Programming Interfaces) for graphics programming:

- **AWT (Abstract Windowing Toolkit)** - introduced in JDK.1.0.
- **Swing API** - comprehensive set of graphics libraries that enhance the AWT. Incorporated into core Java from JDK 1.1 Part of the Java Foundation Classes (JFC) which are used for building GUIs and adding graphics and functionality.

Using Java APIs

To use the packages, the following statements are necessary at the top of a class:

- AWT: (12 packages of 370 classes)

```
import java.awt.*;
```

- Swing API:(18 packages of 737 classes)

```
import javax.swing.*;
```

Swing API

The Swing API is powerful, flexible — and immense. The Swing API has 18 public packages:

```
javax.accessibility
javax.swing
javax.swing.border
javax.swing.colorchooser
javax.swing.event
javax.swing.filechooser
javax.swing.plaf
javax.swing.plaf.basic
javax.swing.plaf.metal
javax.swing.plaf.multi
javax.swing.plaf.synth
javax.swing.table
javax.swing.text
javax.swing.text.html
javax.swing.text.html.parser
javax.swing.text.rtf
javax.swing.tree
javax.swing.undo
```

An Empty Window

Window: a rectangular portion of the monitor screen that can operate independently of the rest of the screen.

It has a width and height, a title and a border.

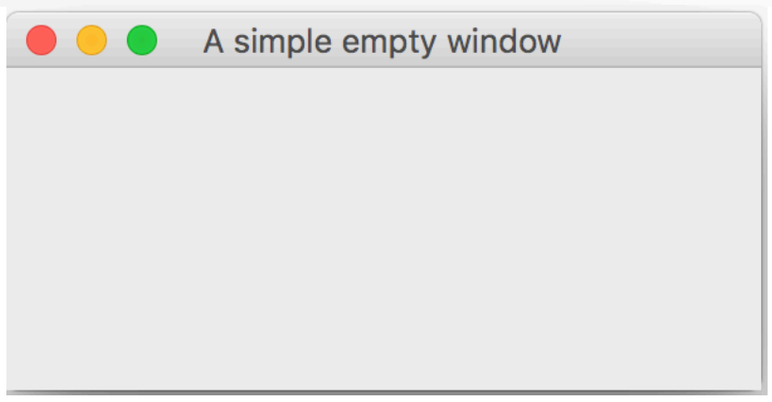
JFrame is a Swing component that is used for creating top-level windows.

Example - Empty Window

SimpleWindow ×

Compile | Undo | Cut | Copy | Paste | Find... | Close | Source Code

```
1  import javax.swing.*;
2  public class SimpleWindow extends JFrame{
3      public SimpleWindow(){
4       setTitle("A simple empty window");
5       setDefaultCloseOperation(EXIT_ON_CLOSE);
6       setSize(300, 150);
7       setVisible(true);
8      }
9      public static void main(String[] args){
10         SimpleWindow w = new SimpleWindow();
11     }
12
13 }
```

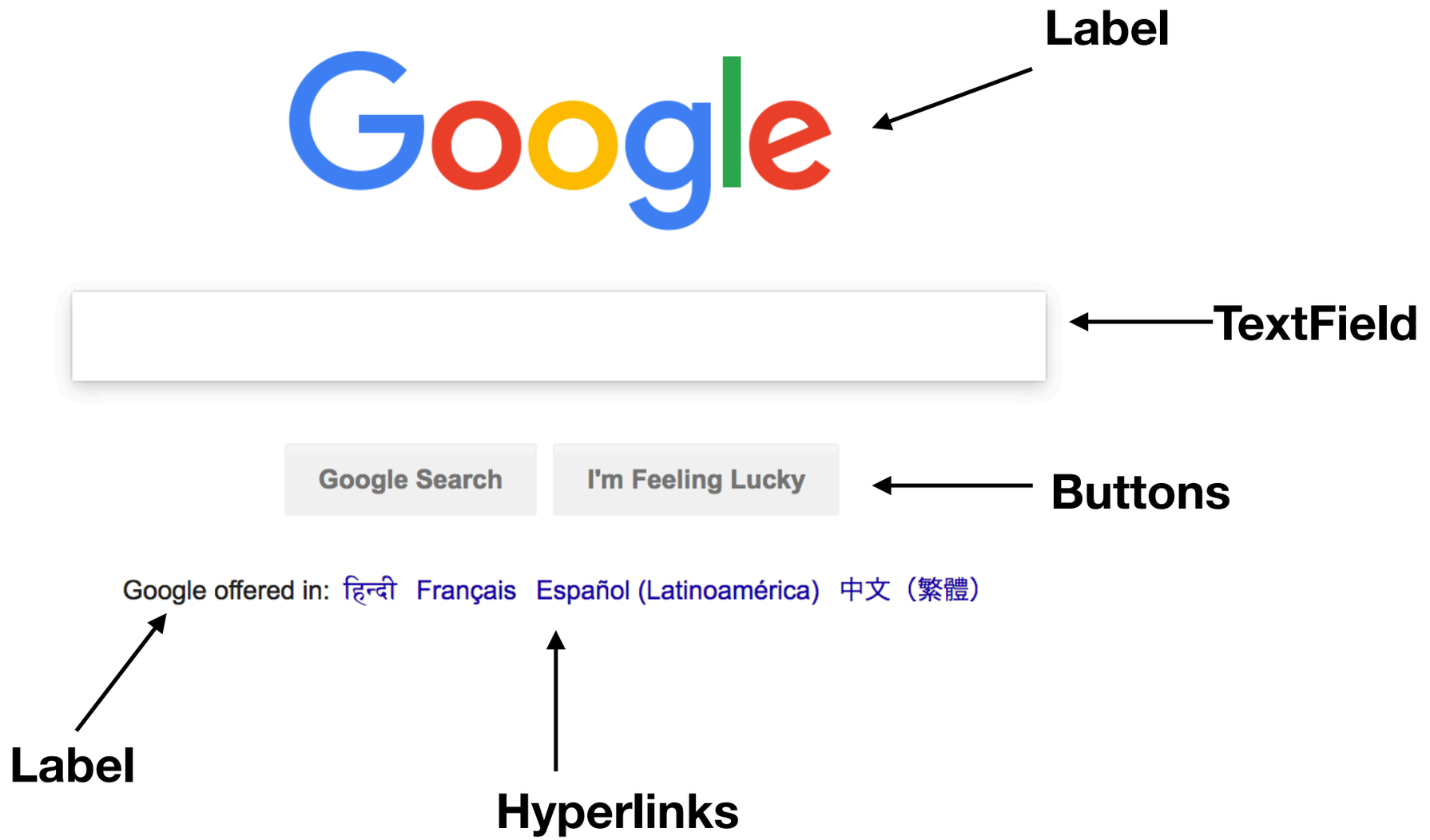


GUI Examples in Use

Google Search I'm Feeling Lucky

Google offered in: हिन्दी Français Español (Latinoamérica) 中文（繁體）

GUI Examples in Use



Simple GUI Components

The following are examples of simple GUI components

- Label
- Text Field
- Text Area
- Command Button

Label

A Label is used for displaying a short string or image. It does not respond to input events such as clicking the mouse or pressing a key.

It is simply used for presenting information.

```
JLabel info = new JLabel(“Enter your name”);
```

Two methods for retrieving or modifying the label:

```
getText(): String

setText(String label)
```

Text Field

A Text Field is used for editing a single line of text.

It is used for data entry and data modification.

```
JTextField nameTF = new JTextField(15);
```

The argument specifies the number of columns that the text field should have (determines the width of the text field and the number of characters that are visible)

Two methods for retrieving or modifying the data in the text field:

```
getText(): String
```

```
setText(String data)
```

Text Area

A Text Area is used for editing a several lines of text.

It is used for data entry and data modification.

```
JTextArea notesTF = new JTextArea(5, 30);
```

The arguments specify the number of rows and columns that the text area will occupy on the screen.

Two methods for retrieving or modifying the data in the text field:

```
getText(): String

setText(String data)
```

Command Button

A Command Button is used for performing particular actions when the user presses the button through a mouse click or key press.

```
JButton search = new JButton(“Search”);
```

The string “Search” becomes the label displayed on the button.

Responding to the mouse click requires some additional code to be written.

Layout Manager

GUI components must be placed at some location on a window.

A layout manager is responsible for positioning GUI components, and there are several layout managers in Swing.

Simple examples:

FlowLayout: positions components in rows. The width of the row is approximate to the window/container width

GridLayout: uses a rectangular grid to position GUI components like in a table. It consists of rows and columns.

Panel

A panel is a container that can contain other GUI components.

A panel can use its own layout manager.

```
JPanel topPanel = new JPanel();
GridLayout grid = new GridLayout(6,2); //6r,2c
topPanel.setLayout(grid);
```

Example - Simple GUI

SimpleWindow ×

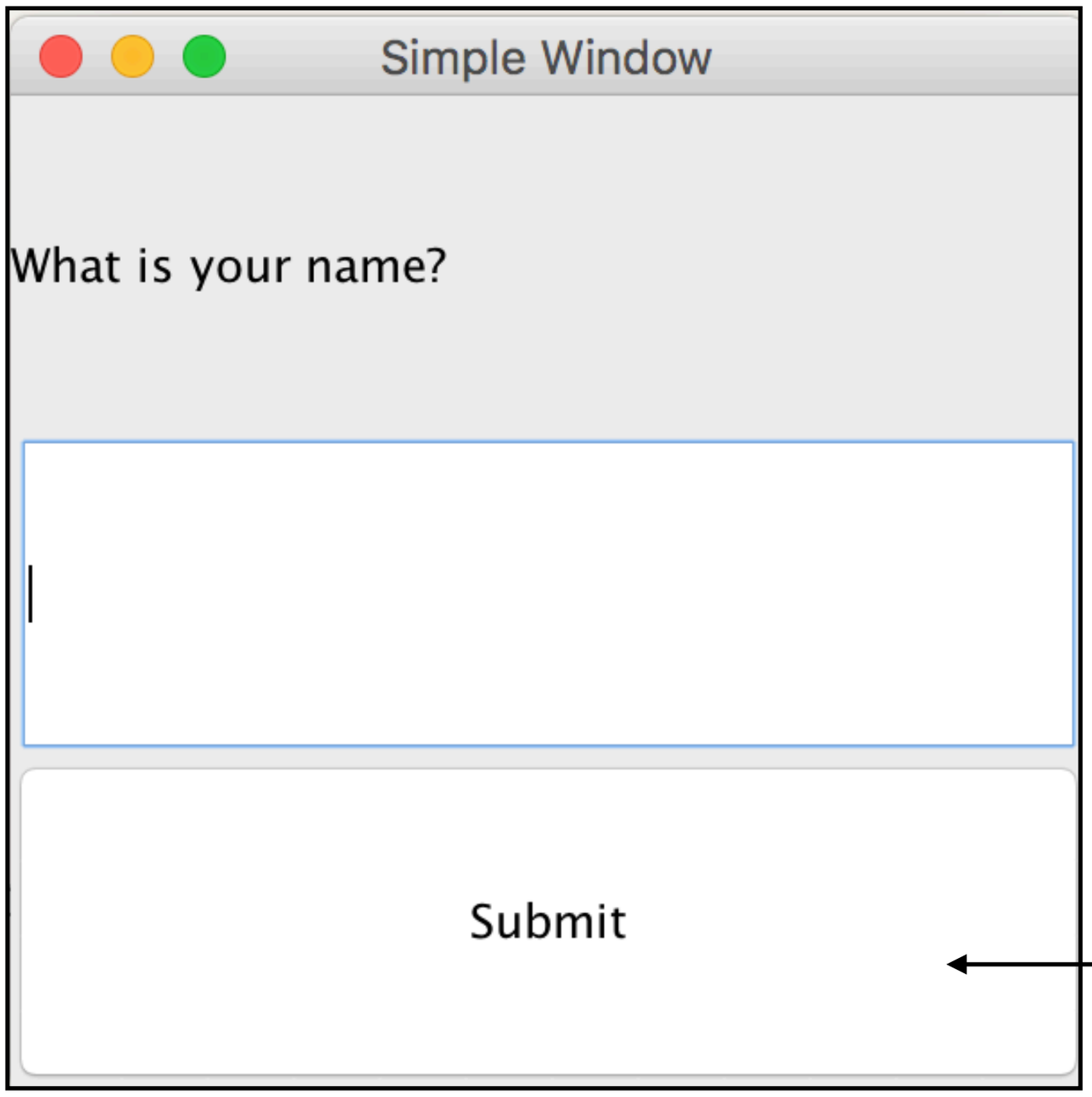
Compile | Undo | Cut | Copy | Paste | Find... | Close | Source Code

```
1  import javax.swing.*;    // for GUI components
2  import java.awt.*;       // for Layout
3
4  public class SimpleWindow extends JFrame {
5
6      private JLabel questionLabel;
7      private JTextField nameField;
8      private JButton submitButton;
9
```

Example - Simple GUI

```
10      public SimpleWindow( ){
11          GridLayout gridLayout = new GridLayout(3,2);
12          setTitle("Simple Window");
13
14          JPanel panel = new JPanel();
15          panel.setLayout(gridLayout);
16
17          questionLabel = new JLabel("What is your name?");
18          nameField = new JTextField(10);
19          submitButton = new JButton("Submit");
20
21          panel.add(questionLabel);
22          panel.add(nameField);
23          panel.add(submitButton);
24          add(panel);
25
26          setDefaultCloseOperation(EXIT_ON_CLOSE);
27          setSize(300,300);
28          setVisible(true);
29
30      }
31
32      public static void main(String[] args){
33          JFrame window = new SimpleWindow();
34      }
35  }
```

Example - Simple GUI



Responding to Window Events

Event-driven programming allows us to respond to window events such as mouse clicks and key presses.

This involves writing application-specific code to take action when some pre-determined event occurs.

This code is referred to as an event handler.

Event Handling

The problem of how to connect application-specific code to a built-in GUI component (that has no prior knowledge of the application-specific code) is dealt with using interfaces.

Common Events

Event	Interface
Clicking on a command button or pressing the Enter key on a command button	ActionListener
Pressing a key on the keyboard	KeyListener
Clicking the mouse on the window surface	MouseListener

Event Handler

The event handler for a particular event must be written in a class that implements the interface.

For example, to specify what to do when a user clicks on a command button, a class must be written that implements the ActionListener interface.

Event Handling Methods

Interface	Interface Methods	Event Class
ActionListener	void actionPerformed(ActionEvent e)	ActionEvent
KeyListener	void keyPressed(KeyEvent e) void keyReleased(KeyEvent e) void void keyTyped(KeyEvent e)	KeyEvent
MouseListener		MouseEvent

Command Button Click

In order to respond to a command button click, the **ActionListener** interface must be implemented.

The **ActionListener** interface has only 1 method:

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

The implementation of this method must specify what to do when the user clicks on a command button.

Command Button Click

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

The ActionEvent object can be queried to find out more about the event that occurred.

Example: `String command = e.getActionCommand( )`

This method returns the label of the button that the user clicked on.

Example - Functional GUI

SimpleWindow ×

Compile | Undo | Cut | Copy | Paste | Find... | Close | Source Code

```
1  import javax.swing.*;   // for GUI components
2  import java.awt.*;      // for Layout
3  import java.awt.event.*; // for ActionListener
4
5  public class SimpleWindow extends JFrame implements ActionListener{
6
7      private JLabel questionLabel;
8      private JTextField nameField;
9      private JButton submitButton;
10
```

Our simple GUI now must implement the ActionListener interface in order to have functionality

Example - Functional GUI

```
11      public SimpleWindow( ){
12          GridLayout gridLayout = new GridLayout(3,2);
13          setTitle("Simple Window");
14
15          JPanel panel = new JPanel();
16          panel.setLayout(gridLayout);
17
18          questionLabel = new JLabel("What is your name?");
19          nameField = new JTextField(10);
20          submitButton = new JButton("Submit");
21
22          submitButton.addActionListener(this);
23
24          panel.add(questionLabel);
25          panel.add(nameField);
26          panel.add(submitButton);
27          add(panel);
28
29          setDefaultCloseOperation(EXIT_ON_CLOSE);
30          setSize(300,300);
31          setVisible(true);
32
33      }
34
```

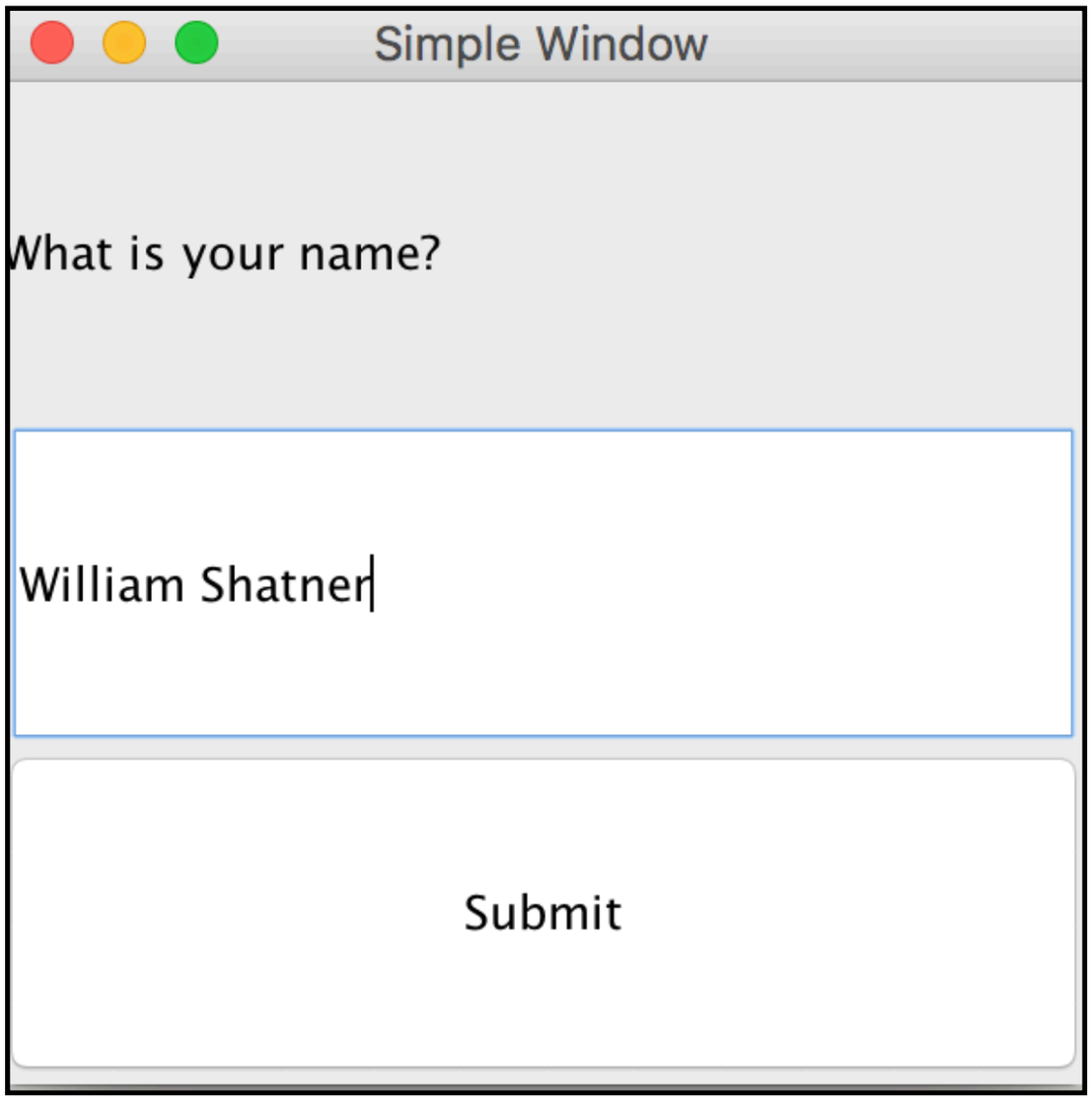
The button requires functionality

Example - Functional GUI

```
35      public void actionPerformed(ActionEvent e){
36          String command = e.getActionCommand();
37          String userName = nameField.getText();
38          questionLabel.setText("Hello " + userName);
39          nameField.setText("");
40      }
41
42      public static void main(String[] args){
43          JFrame window = new SimpleWindow();
44      }
45  }
46
```

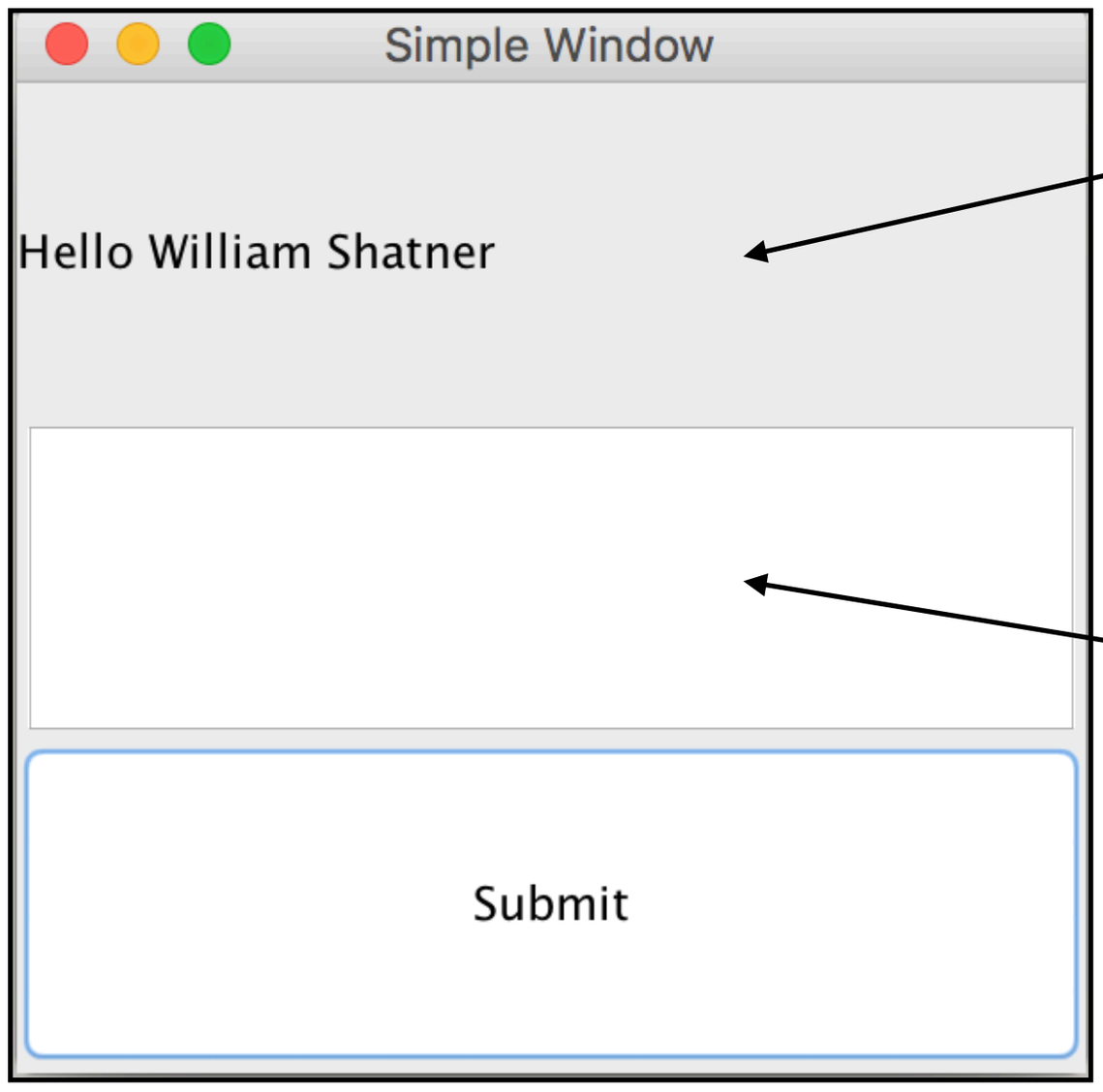
This is what to do when the button is clicked

Example - Functional GUI



Entering text into the text field

Example - Functional GUI



Label was changed with a message containing the data captured from the text field

Text field data was captured
Text field was then cleared

After clicking the Submit Button

GUI Examples in Use

Google

Advanced Search

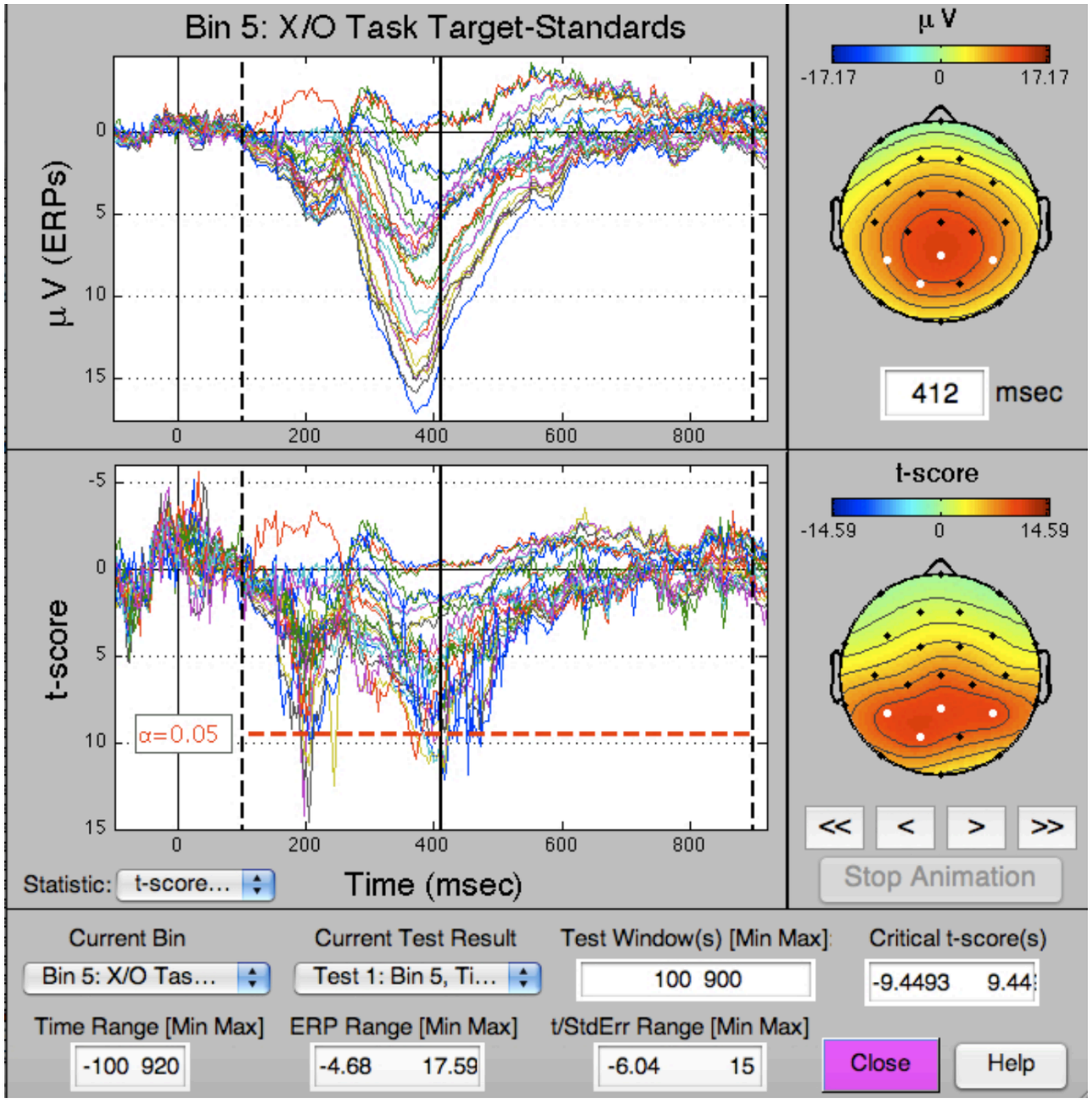
Find pages with...		To do this in the search box.
all these words:		Type the important words: `tri-colour rat terrier`
this exact word or phrase:		Put exact words in quotes: `"rat terrier"`
any of these words:		Type `OR` between all the words you want: `miniature OR standard`
none of these words:		Put a minus sign just before words that you don't want: `-rodent, -"Jack Russell"`
numbers ranging from:	to	Put two full stops between the numbers and add a unit of measurement: `10..35 kg, £300..£500, 2010..2011`

Then narrow your results by...

language:	**any language**	Find pages in the language that you select.
region:	**any region**	Find pages published in a particular region.
last update:	**anytime**	Find pages updated within the time that you specify.
site or domain:		Search one site (like `wikipedia.org`) or limit your results to a domain like `.edu`, `.org` or `.gov`
terms appearing:	**anywhere in the page**	Search for terms in the whole page, page title or web address, or links to the page you're looking for.
SafeSearch:	**Show most relevant results**	Tell SafeSearch whether to filter sexually explicit content.
file type:	**any format**	Find pages in the format that you prefer.
usage rights:	**not filtered by licence**	Find pages that you are free to use yourself.

Advanced Search

GUI Examples in Use



Netbeans IDE

Putting together the layout of a GUI from scratch is very time consuming and error prone.

There are tools that programmers can use to simplify this process such as the Netbeans IDE (Integrated Development Environment).

It is important however to be able to understand the code that is generated by the IDE.