# Intro to GUIs (Graphical User Interfaces)

Section 2.5 Intro. to GUIs: a GUI Greeter

Section 3.7 Graphical/Internet Java: Einstein's Equation

## Alan Kay

Alto

- 1980
- The first *GUI* interface
- Developed at Xerox PARC by a group which included Kay.
- The key ideas are based on child psychology

from www.parc.xerox.com

"Only people born before a technology is invented think of it as a technology"

## Steven Jobs (1955- )

MacOS

- 1984
- The first commercially successful GUI OS
- Developed at Apple by a group that included Jobs.
- Based on the Xerox PARC Alto


from www.apple.com

# Example: Building a Temperature-Conversion GUI

(similar to Einstein Calculator in §3.7)

**Problem Scenario**:

Write a program to read a temperature in Celsius, compute and display the equivalent Fahrenheit temperature. However, instead of the text-based solution from before, use a graphical user interface.

4

## Models of Programming

- Limited interaction with the user
- Activity is initiated by the program:
  1. input data
  2. process the data
  3. output the results

- Continuous interaction with the user
- Activity is initiated by user events:
  1. user initiates events
  2. system responds

## Kinds of User Interfaces

- Driven by:
  - Text input prompts
  - Command-line interfaces

- Driven by user-initiated graphical events, e.g.,
  - pressing mouse button
  - releasing mouse button
  - dragging the mouse
  - pressing a keyboard key

## Behavior of our Temperature-Converter GUI

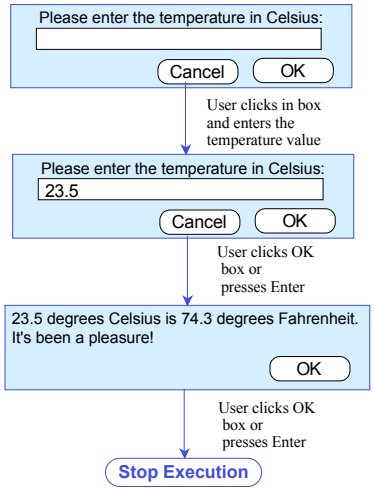
The program will have a graphical user interface that will:

- Open a window containing:
  - a prompt for a Celsius temperature
  - a box to enter the temperature
  - OK and Cancel buttons
- Let the user enter a Celsius temperature and then click OK (or press Enter key)
- Compute the equivalent Fahrenheit temperature
- Open a second window that displays:
  - the Fahrenheit temperature
  - a button for terminating the program

7

**Transition Diagrams:**

Graphical behavior is frequently modeled with_____ _____that show the various states of execution and what causes transitions from one state to another.




8

**Objects (_____):**

In addition to the temperature variables and constants from our earlier text-based version, we obviously need new _____ objects, usually called _____, to build the GUI — windows, buttons, menus, etc.

_____

- Original package provided in early versions of Java (1.0 & 1.1)
- A collection of classes — e.g., **`Component`** and **`Applet`** — for building widgets.

9
9

_____

- Extension to Java in 1997; now a standard part of Java 2. Java 2 still supports AWT, but . . .
- JFC has a collection of _____ for enhanced GUIS that should be used whenever possible:
  - more powerful
  - easier to use
  - large collection, but most useful are those in _____ package
  - Names all begin with _____

10

Some of the more commonly-used Swing classes:

- `JOptionPane` To implement pop-up input or output dialog boxes
- `JLabel` To hold a short one-line message or an image
- `JTextArea` To hold multi-line messages
- `JFrame` To create a main frame (window) for a GUI
- `JApplet` Like `JFrame`, but for applets
- `JPanel` To create panes (or panels) to put on a frame
- `JButton` To create buttons
- `JMenu` To create menus
- `JFileChooser` A simple way for a user to choose a file

See the "How to Make Dialogs" section of Java's Swing Tutorial:
**http://java.sun.com/docs/books/tutorial/uiswing/TOC.html#components**

11

```
/** GUITemperature.java converts Celsius temperatures to
 * Fahrenheit.  It uses a graphical user interface to
 * interact with the user.
 * Author: L. Nyhoff
 * Date:   Nov. 29, 2002
 */
```

Building our GUI Temperature Converter

_____

```
class GUITemperature extends Object  {

  public static void main(String [] args)   {

   //--- Get Celsius temperature

    double fahrenheit = ((9.0/5.0)*celsius) + 32;

   //--- Output Fahrenheit temperature

  }
}
```

12

Since we need a widget for creating pop-up input and output windows, we will use **`JOptionPane`**. Some of its most useful methods are the following; all are class (static) methods:

- **`showInputDialog()`** Prompt for input
- **`showMessageDialog()`** Display message to user
- **`showConfirmDialog()`** Ask user to confirm something, usually with Yes, No, or Cancel
- **`showOptionDialog()`** "a grand unification of the above three" (from Java's API doc.)

`http://java.sun.com/j2se/1.4.1/docs/api/`

13

## Input Dialogs:

- Used to get text input from user
- Simple form
  **`showInputDialog(`*`prompt`*`)`**
- Returns a **`String`**
- the ***`prompt`*** can be a **`String`**, a graphic image, or another Java **`Object`**.

So we can implement the first state in our program:

`http://java.sun.com/j2se/1.4.1/docs/api/`

14

```
/** GUITemperature.java
      . . .
 */

import javax.swing.*;

class GUITemperature extends Object  {

  public static void main(String [] args)   {

    //--- Get Celsius temperature




    double fahrenheit = ((9.0/5.0)*celsius) + 32;

   //--- Output Fahrenheit temperature

  }
}
```

15

## Message Dialogs:

- Used to display information to the user
- Simple form
  **`showMessageDialog(null, `*`message`*`)`**
- A void method
- The ***`message`*** is a **`String`** or other Java **`Object`**
- The first parameter refers to the frame in which the message is to be displayed; **`null`** causes a default frame to be used.

So we can implement the last state in our program:

`http://java.sun.com/j2se/1.4.1/docs/api/`

16

```
/** GUITemperature.java
      . . .
 */
import javax.swing.*;
class GUITemperature extends Object  {
  public static void main(String [] args)   {
    //--- Get Celsius temperature
    String celsiusString = JOptionPane.showInputDialog(
                "Please enter the temperature in Celsius: ");
    double celsius = Double.parseDouble(celsiusString);
    double fahrenheit = ((9.0/5.0)*celsius) + 32;
    //--- Output Fahrenheit temperature
  }
}
```

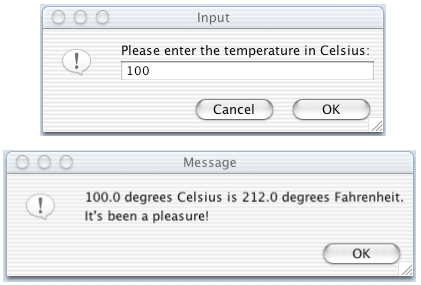


We added the line

```
System.exit(1);
```

to ensure that the application stops running when the OK button is clicked.

## Execution:





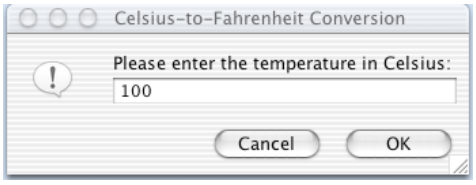

## Some Improvements:

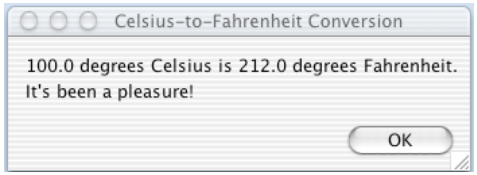
- Give each dialog box a meaningful title and appropriate symbol for type of dialog:

```
String TITLE = "Celsius-to-Fahrenheit Conversion";

String celsiusString = JOptionPane.showInputDialog(
            null,
            "Please enter the temperature in Celsius: ",
            TITLE,
            JOptionPane.QUESTION_MESSAGE);

JOptionPane.showMessageDialog(null,
       celsius + " degrees Celsius is " + fahrenheit
     + " degrees Fahrenheit.\nIt's been a pleasure!\n",
       TITLE,
       JOptionPane.PLAIN_MESSAGE);
```

http://java.sun.com/j2se/1.4.1/docs/api/





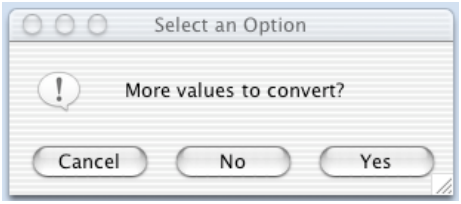

http://java.sun.com/j2se/1.4.1/docs/api/

- Allow repeated conversions:

```
do {
  String celsiusString = JOptionPane.showInputDialog(
         null,
         "Please enter the temperature in Celsius: ",
         TITLE,
         JOptionPane.QUESTION_MESSAGE);

     double celsius = Double.parseDouble(celsiusString);

     double fahrenheit = ((9.0/5.0)*celsius) + 32;

     JOptionPane.showMessageDialog(null,
          celsius + " degrees Celsius is " +  fahrenheit
        + " degrees Fahrenheit.\nIt's been a pleasure!\n",
          TITLE,
            JOptionPane.PLAIN_MESSAGE);
}
```

http://java.sun.com/j2se/1.4.1/docs/api/






http://java.sun.com/j2se/1.4.1/docs/api/