

Lesson 85: Intro to GUI Development (W30D1-W30D2)

Balboa High School

Michael Ferraro

March 22, 2016

Do Now

- Create a new project called TempConverter
- Download [TempConverterLogic.java](#) and implement the two included methods
- Write a driver, TempConverter.java, which instantiates TempConverterLogic into an object and calls both methods as a test. Test cases:
 - $0^{\circ}\text{C} \approx 32^{\circ}\text{F}$
 - $100^{\circ}\text{C} \approx 212^{\circ}\text{F}$

Students will learn the basics of building a GUI in Java.

GUI Development: Art or Science?

- Trial-and-error is a large part of the process
- There are often multiple ways of accomplishing the same appearance
- UIs may look different on different platforms supporting Java!

Tedium of Manual GUI Development

- Writing code manually for GUIs is burdensome and tedious
- There are quite a few components, each with its own constructors, behaviors, etc.
- Many developers build GUIs, in part, by using tools that write code

- Early versions of Java had AWT, or Abstract Window Toolkit, a set of widgets that could be included in a GUI
- Widget examples: checkboxes, radio buttons, pull-down list selectors, etc.
- Later came Java Swing, which included more widgets
 - We'll use Swing components
 - [Here](#) is an overview of what Swing provides.

Today's Goal

- Today, we aim to build a simple GUI app
- Try it out: [TempConverter.jar](#)¹

¹run via `java -jar TempConverter.jar`

- A free² plug-in for Eclipse for GUI development
- Homepage is at <http://www.cloudgarden.org/jigloo/> — use the installation instructions to set Jigloo up at home
- Already installed on Linux workstations in Rm124 & Rm319

²Free only for non-commercial use!

Add a GUI Class to Our Project

- Right-click the `src` folder
- New → Other → GUI Forms → Swing → JFrame
- Call the class `TempConverterUI`
- By default, a new GUI class is opened with the Jigloo Form Editor; may be done later by right-clicking the class and selecting *open with...*

Tour of Jigloo Interface

- Main window: where GUI is built
- *Outline* tab on right: GUI components & preview button
- *Properties* tab at bottom (near *Console/Problems* tabs)

Properties of a JFrame

- Set title via *Properties* tab: *Temperature Converter*
- Set *layout* for the frame: flow layout
- Layouts describe how components will be arranged within a JFrame (and Containers, which can hold GUI components, including other Containers!)
- Layouts described briefly in Litvin's Ch. 16 (see link at end of slides)

Driver Class Builds GUI

Modify TempConverter's main():

- Comment-out test statements
- Add these statements:

```
TempConverterUI ui = new TempConverterUI();  
ui.setVisible(true); // omit & never see GUI!
```

- Run driver to see the GUI
- To ensure we don't accidentally run the main() in the TemperatureUI class later, let's comment it out

Giving the GUI a Brain

- In practice, the GUI needs some intelligence — like `TempConverterLogic`'s methods
- In this program, the GUI needs the ability to call upon the logic class' methods
- Overload `TempConverterUI`'s constructor: make a version that accepts reference to a `TempConverterLogic` object
- Modify the driver's main to use your new constructor

Adding Elements to the GUI

- Let's create JPanels³
- Each JPanel will hold the following:
 - a JTextField in which to type a temperature
 - a JLabel, which is just some text (non-editable by user)
 - a JButton, which will cause a temp. conversion

³These are subclasses of the Java Swing Container type, so these hold GUI elements.

Adding Elements to the GUI (cont'd)

- Add the JPanel's to the JFrame by dragging-and-dropping
- Let's match certain components' widths for aesthetics: Set the dimensions of multiple elements at once by holding down CTRL while clicking the elements and then using the mouse to resize

- Let's click the preview button in Jigloo to see how our GUI is going to look
- Sometimes, the Form Editor in Jigloo won't give us a view identical to the actual look when we run our program — trial and error!

GUI Elements are Listening!

- The elements you added can *listen* for events — e.g., left and right button clicks, a selection being made with a mouse or tab key, etc.
- Methods called `ActionListeners` enable us to tie buttons, for example, to method calls

ActionListeners for Our Buttons

- Click on a button in the Form Editor
- See right side of *Properties* tab (@ bottom)
- Find event name *ActionListener* & select *anonymous*
- Let's see the generated Java — anonymous inner class/method produced!

Connecting GUI to Logic

- Have our `ActionListeners` send info to our logic class' methods
- Note that `JTextFields` store `Strings`
 - get values via `getText()`
 - set values via `setText()`
 - these methods inherited from `JTextComponent` — so they don't show on the API page for `JTextField`
- Remember how to convert `Strings` to `Doubles`:
`Double.parseDouble(aString)`

Make the GUI More Robust

- Remember, when relying on user entry, we need to program defensively (defend against user mistakes or intentional attacks!)
- If a user types an invalid value — i.e., one that cannot be interpreted as a `Double` — there will be an exception
- Solution: Use `try-catch`, printing caught exceptions to the console

Console Printing in a GUI?!

- Nicer solution: Have a pop-up dialog box that informs the user of their error
- File → New → Other → GUI Forms → Swing → JDialog
InvalidValueJDialog
- Add a JLabel with the error message
- Add an button (I prefer)

Console Printing in a GUI?! (cont'd)

- Tie action to an event on that button: `dismiss dialog dispose()`;
- When exception caught during button pushes in `TempConverterUI`, make a new dialog box object

```
InvalidValueJDialog errorJDialog =  
    new InvalidValueJDialog(new javax.swing.JFrame());
```

- [Jigloo Examples](#) (click *examples* link in left sidebar)
- Oracle's [Java Swing Tutorial](#)
- Litvin's [chapter on GUI development](#) (old edition)

- Read §§16.3–16.5 of Litvin's [Chapter 16 \(old edition\)](#)
- Attempt to recreate what we've built in class from scratch! This is good practice :)