COMP2232 - Lab Practical #5

READ THIS ENTIRE DOCUMENT BEFORE STARTING

Concept: Working with GUI using Swing Components.

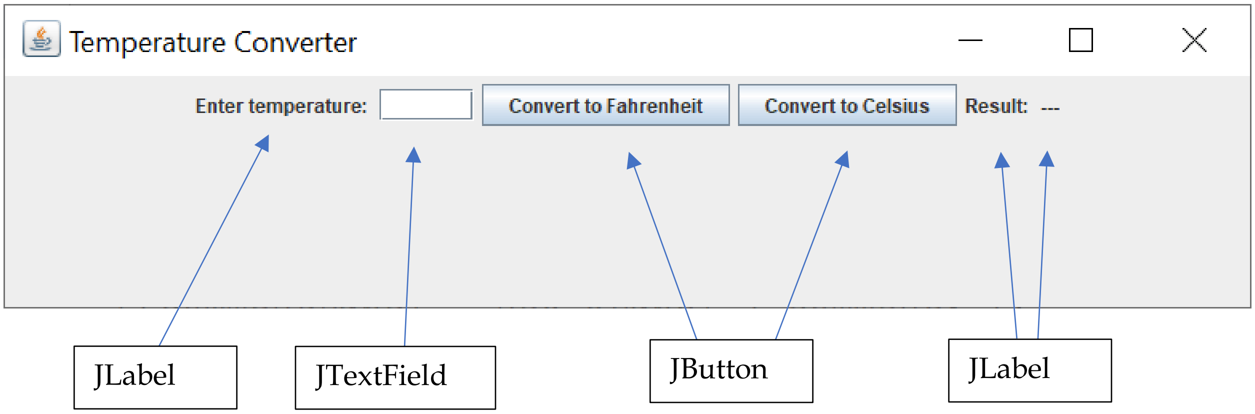
A – Introduction to Working with Graphical User Interfaces

In preparation for this week's lab, watch the following 6 short videos in the **GUI - Java Swing Tutorials** section on the course page.

- Video 1: Creating a Window (6m 18s)
- Video 2: Centering the Window (7m 20s)
- Video 3: Maximizing a Window (2m 52s)
- Video 4: Using a Button (3m 45s)
- Video 5: Using BorderLayout (4m 48s)
- Video 6: A Small Application (7m 55s)

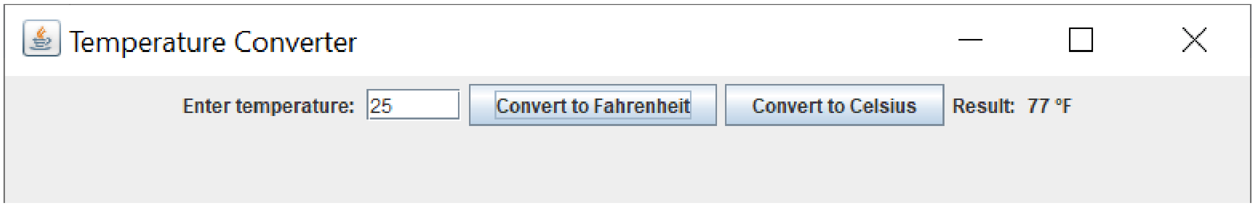
B – Working with Graphical User Interfaces

- You will be re-using the **<u>Temperature Converter</u>** class you created in Lab #1 and creating a GUI interface like the one below.



Create a class called **ConvertTemp** which will be used to create the GUI above. You may use a FlowLayout to place components on your frame.

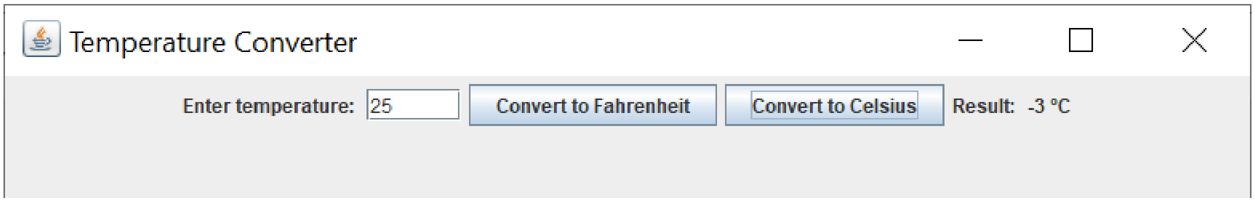
The user should be allowed to enter a temperature into the text field (e.g. 25). Clicking on the "Convert to Fahrenheit" button will display the converted value in the result area (e.g. 77) with the correct unit (F). (*See image below.*)



Enter temperature: 25

Result: 77 °F

Clicking on the "Convert to Celsius" button will display the converted value in the result area (e.g. -3) with the correct unit (C). (*See image below.*)



Enter temperature: 25

Result: -3 °C

Text fields hold values as text values (i.e. characters vs. numeric). However, numeric values are required for your calculations. Therefore, when getting the value from the text field, you will have to convert it to Integer (i.e. a number) to allow for calculations.

You can use: `Integer.parseInt(text)`

*This function will take the value in the variable "text" and **returns** it as a numeric value.*

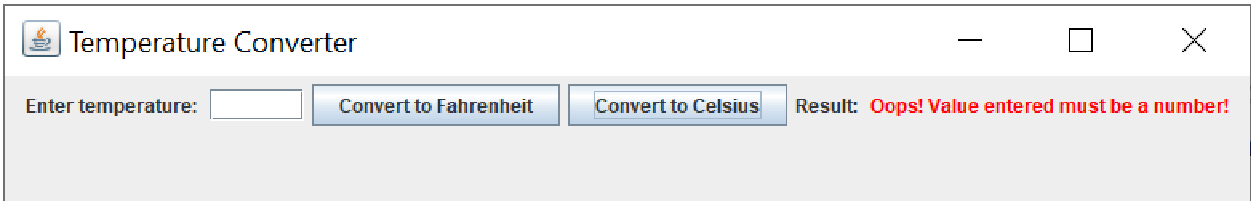
When attempting to display the converted temperature, you will have to convert the numeric value back to text (String) before placing into the label to be displayed.

You can use: `Integer.toString ((int)convertedTemp)`

*This will take the "convertedTemp" numeric value and **returns** a String which can be accepted by the label.*

Exception Handling

In the event that the user did not enter a temperature but still goes ahead and clicks on one of the convert buttons, an exception (type of error) will be thrown. It will throw a **NumberFormatException**. You will need to use an Exception handler to display the message "**Oops! Value entered must be a number!**" to alert the user.



Enter temperature:

Result: Oops! Value entered must be a number!

You can use the following handler:

```
try
{
   //get temperature and convert here
   //display converted temperature here
}
catch(NumberFormatException e)
{
   //display appropriate error message here
}
```

<End of Lab>