Java Eclipse Temperature Converter GUI Tutorial

Contents

Time required: 60 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Download and Setup the Eclipse IDE

Eclipse is an open-source IDE for Java and other languages. It includes the capability to drag and drop components for Java GUI program much like Visual Studio does for C#.

NOTE: Please do not download the Eclipse installer. This can cause issues.

1. Go to https://www.eclipse.org/downloads/packages/
2. Go to **Eclipse IDE for Enterprise Java and Web Developers**.
3. Choose **Windows x86_64**. This will download a zip file.
4. Extract the files from the downloaded zip file. You should end up with an eclipse folder with all the eclipse program files inside it.
5. Move that folder to your c: drive. Example: **c:\eclipse**

6. Go into the **c:\eclipse** folder → Right Click **eclipse.exe** → **Send to** → **Desktop (create shortcut)**.

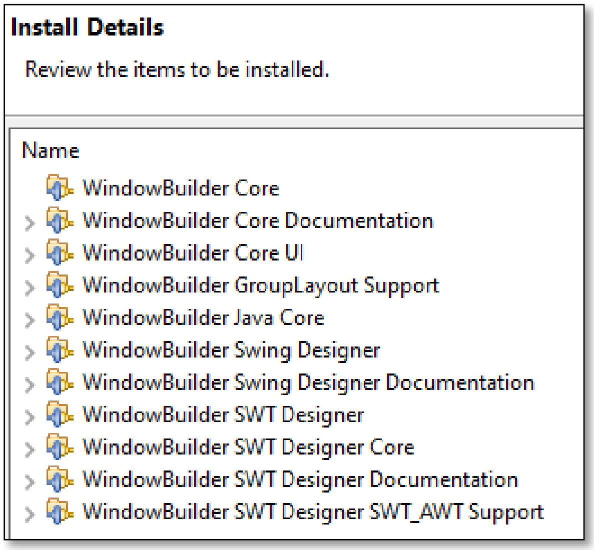
7. Start **Eclipse**.

8. Choose a workspace you can find again. All your Eclipse projects and files are stored here.

Install Window Builder

1. In **Eclipse**, go to **Help**, **Install New Software**. In **Work with**, choose **--All Available Sites--**

2. Go to **General Purpose Tools.** Select the items shown in the screenshot below.



Install Details

Review the items to be installed.

Name

- WindowBuilder Core
- WindowBuilder Core Documentation
- WindowBuilder Core UI
- WindowBuilder GroupLayout Support
- WindowBuilder Java Core
- WindowBuilder Swing Designer
- WindowBuilder Swing Designer Documentation
- WindowBuilder SWT Designer
- WindowBuilder SWT Designer Core
- WindowBuilder SWT Designer Documentation
- WindowBuilder SWT Designer SWT_AWT Support

3. Click **Next**. Accept the license agreement. Click **Finish**.

4. Restart Eclipse when prompted.

Create Swing Application

1. In Eclipse, close the Welcome window.

2. Click **File** → **New** → **Other**. Go to **Java** → **Java Project**. Click **Next**.

3. Name the project **TemperatureConverter**. Click **Finish**.

4. Module Name: **TemperatureConverter**, Click **Create**.

5. Right click the **src** folder → **New** → **Other** → **WindowBuilder** → **Swing Designer** → **JFrame**. Click **Next**.
6. Name: **TemperatureConverter** Click **Finish**. The file opens automatically in **Source** View.
7. In **Package Explorer**: Right Click on **module-info.java** Click **Delete**.
8. Toward the bottom of Eclipse, Click the **Design** tab. You should see the GUI.
9. Click on the **JFrame** edge.
10. Go to **Properties**, Type **Temperature Converter** in the title property. Press Enter.
11. Click the **contentPane** (The main window inside the JFrame).
12. In **Properties**: Choose **Layout: Absolute Layout (absolute)**

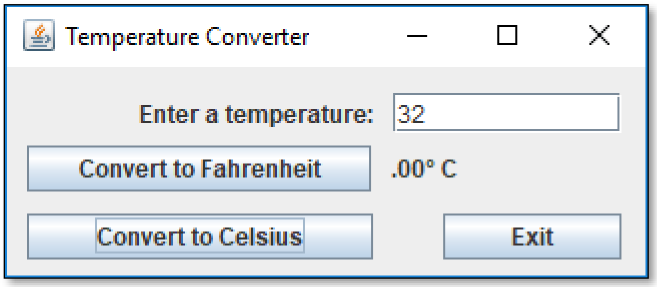
Add a JLabel

1. From the **Palette** → **Components:** Click **JLabel.** Click in the window where you want the **JLabel.**
2. In **Properties**:
    a. text: **Enter a temperature:**
    b. horizontalAlignment: **RIGHT**

More Components

1. Add a **JLabel** to display the converted temperature.
    a. Variable: **lblDisplay**
2. Add a **TextField**. Change the **Variable** to **txtTemperature**. The Variable name is the same as the name of the control.
3. Add a **JButton.**
    a. text: **Convert to Fahrenheit**
    b. Variable: **btnConvertToFahrenheit**
4. Add a **JButton.**
    a. text: **Convert to Celsius**
    b. Variable: **btnConvertToCelsius**

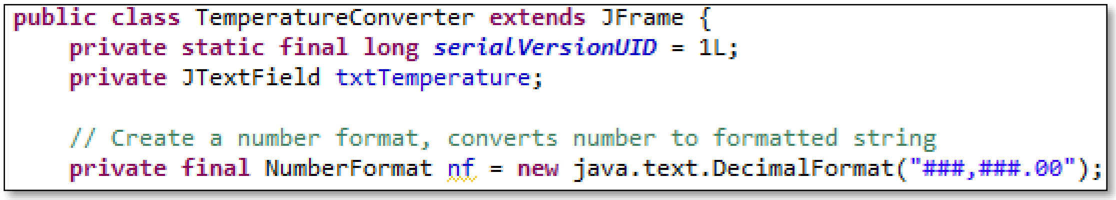
5. Add a **JButton**.
    a. text: **Exit**
    b. Variable: **btnExit**
6. Arrange the controls as shown in the example. Notice that the controls snap and align automatically to guidelines.
7. After you have resized the **lblDisplay** label, clear the text property.



Run the program.

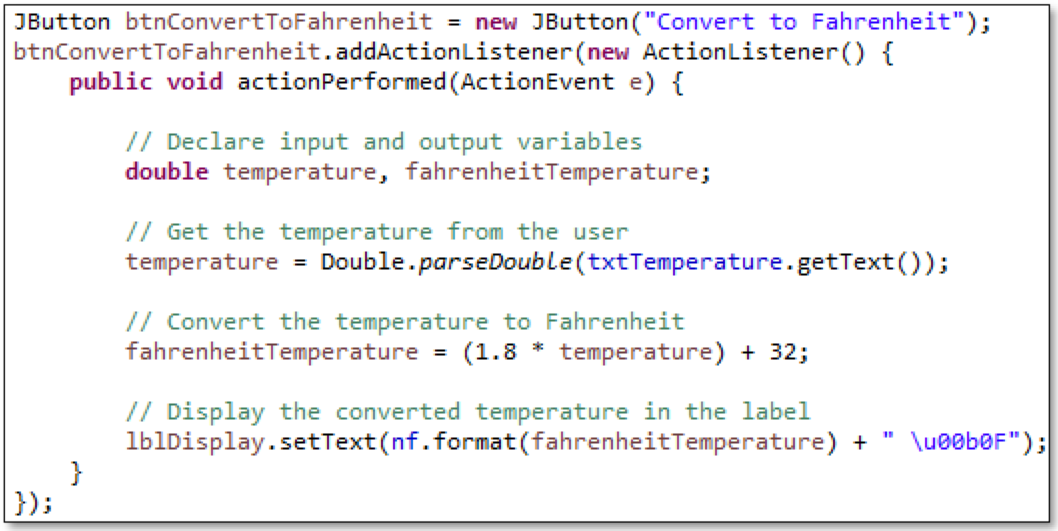
Code

1. Click the **Source** tab. Open the import block and add **import java.text.NumberFormat;**
2. Add the line shown below: private final NumberFormat etc. to the beginning of the **TemperatureConverter** class.



```
public class TemperatureConverter extends JFrame {
    private static final long serialVersionUID = 1L;
    private JTextField txtTemperature;

    // Create a number format, converts number to formatted string
    private final NumberFormat nf = new java.text.DecimalFormat("###,###.00");
```

3. Click the **Design** tab. Double Click the **Convert to Fahrenheit** button.
4. Add the following code to the **btnConvertFahrenheit actionPerformed event**.

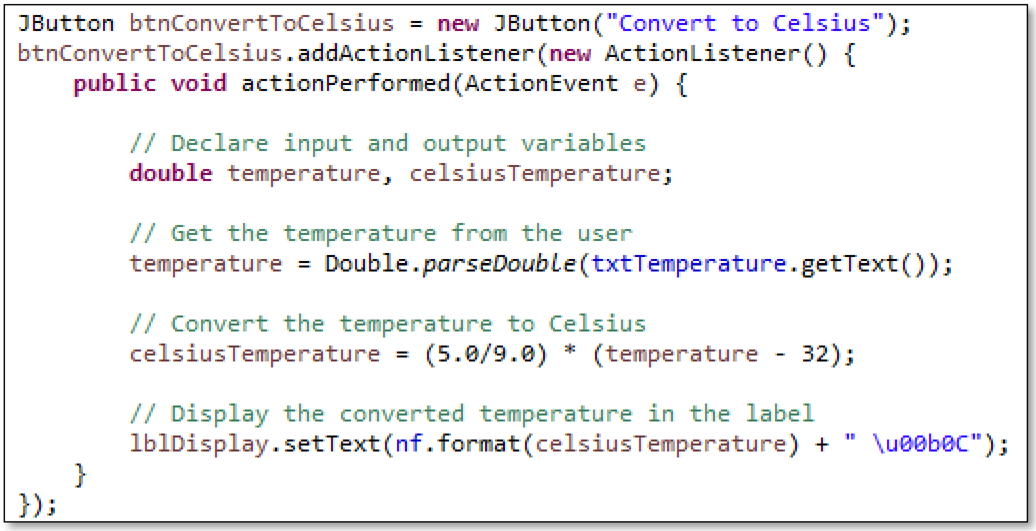


```
JButton btnConvertToFahrenheit = new JButton("Convert to Fahrenheit");
btnConvertToFahrenheit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        // Declare input and output variables
        double temperature, fahrenheitTemperature;

        // Get the temperature from the user
        temperature = Double.parseDouble(txtTemperature.getText());

        // Convert the temperature to Fahrenheit
        fahrenheitTemperature = (1.8 * temperature) + 32;

        // Display the converted temperature in the label
        lblDisplay.setText(nf.format(fahrenheitTemperature) + " \u00b0F");
    }
});
```

NOTE: " \u00b0F" adds the degree symbol ° and F to the display label. \u00b0 is the Unicode symbol for degree °.

5. Click the **Design** tab. Double Click the **Convert to Celsius** button.

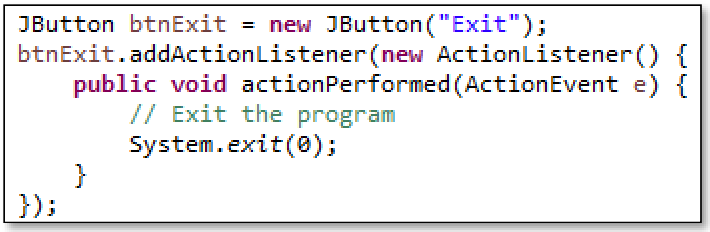
6. Add the following code to the **btnConvertToCelsius actionPerformed event.**



```
JButton btnConvertToCelsius = new JButton("Convert to Celsius");
btnConvertToCelsius.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        // Declare input and output variables
        double temperature, celsiusTemperature;

        // Get the temperature from the user
        temperature = Double.parseDouble(txtTemperature.getText());

        // Convert the temperature to Celsius
        celsiusTemperature = (5.0/9.0) * (temperature - 32);

        // Display the converted temperature in the label
        lblDisplay.setText(nf.format(celsiusTemperature) + " \u00b0C");
    }
});
```

7. Click the **Design** Tab. Double Click the **Exit** button.

8. Add the following code to the **btnExit.addActionListener**.



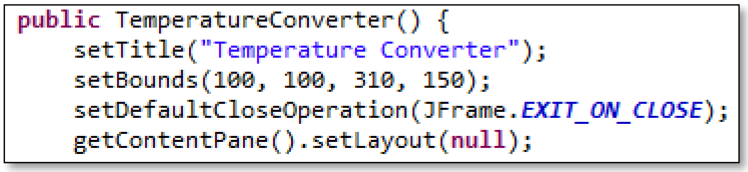
```
JButton btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Exit the program
        System.exit(0);
    }
});
```

9. Add a try catch or tryparse block around each convert button's code to catch any bad entries.

10. Click the Run button ▶ to compile and execute the program.

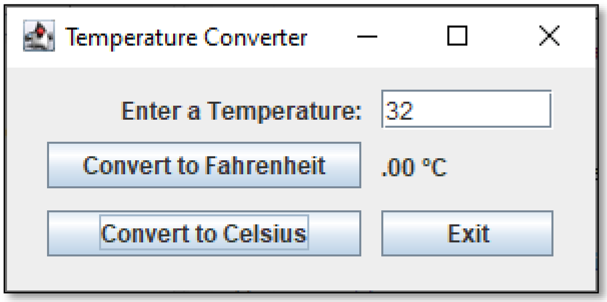
11. You may have to adjust the size of the program and the controls.

12. To control where the program opens, change the first two numbers of **setBounds**. You can use 100, or whatever x, y values you wish.



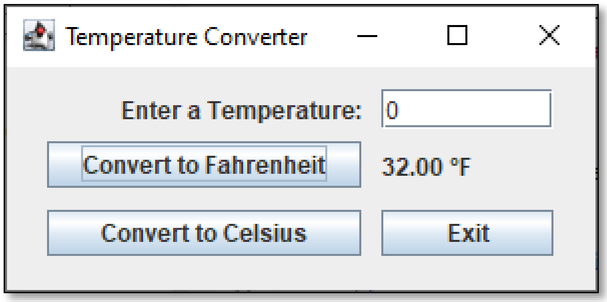
```
public TemperatureConverter() {
    setTitle("Temperature Converter");
    setBounds(100, 100, 310, 150);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);
```

Example runs:



Enter a Temperature: 32

.00 °C



Enter a Temperature: 0

32.00 °F

Export to a JAR File

1. Go to **File → Export → Java → Runnable JAR file**. Click **Next**.
2. Launch Configuration: **TemperatureConverter – TemperatureConverter.**
3. Browse and save the JAR file somewhere you can find it as **TemperatureConverter**.
4. Double click the resulting JAR file to ensure it works properly.

Assignment Submission

1. Attach the pseudocode.
2. Go to the workspaces folder and zip up the folder called **TemperatureConverter**.
3. Zip up the Eclipse project folder and the Jar.
4. Attach screenshots showing the successful operation of the program.
5. Submit in Blackboard.