**Week-3 Lab**

**package** lab3;

/\*\*

\* Program for Temperature Converter GUI.

\* **@author** Prashanthi Sudha Kosgi

\* Date : 6/16/2017

\*

\*/

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.\*;

//import java.awt.event.\*;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ItemEvent;

**import** java.awt.event.ItemListener;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.\*;

**import** javax.swing.border.TitledBorder;

**public** **class** LabThree **extends** JFrame {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** JTextField textFieldInputTemp;

**public** JTextField textFieldOutputTemp;

**public** JRadioButton radioButtonFromCel;

**public** JRadioButton radioButtonFromFar;

**public** JRadioButton radioButtonFromKel;

**public** JRadioButton radioButtonToCel;

**public** JRadioButton radioButtonToFar;

**public** JRadioButton radioButtonToKel;

**public** JLabel inputLabel;

**public** JLabel outputLabel;

**public** JPanel panelNorth, panelEast, panelWest, panelSouth;

**public** ButtonGroup buttonGroupFrom, buttonGroupTo;

**public** **static** **void** main(String[] args) {

**new** LabThree();

}

**public** LabThree() {

FlowLayout fl = **new** FlowLayout();

setLayout(fl);

JFrame frame = **new** JFrame("Temperature Converter");

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

inputLabel = **new** JLabel("INPUT");

panelNorth = **new** JPanel(**new** FlowLayout());

panelNorth.add(inputLabel);

textFieldInputTemp = **new** JTextField(10);

panelNorth.add(textFieldInputTemp);

panelWest = **new** JPanel(**new** GridLayout(0, 1, 1, 1));

panelWest.setBorder(**new** TitledBorder("Input Scale"));

radioButtonFromCel = **new** JRadioButton("Celsius");

radioButtonFromFar = **new** JRadioButton("Fahrenheit");

radioButtonFromKel = **new** JRadioButton("Kelvin");

buttonGroupFrom = **new** ButtonGroup();

buttonGroupFrom.add(radioButtonFromCel);

buttonGroupFrom.add(radioButtonFromFar);

buttonGroupFrom.add(radioButtonFromKel);

panelWest.add(radioButtonFromCel);

panelWest.add(radioButtonFromFar);

panelWest.add(radioButtonFromKel);

panelEast = **new** JPanel(**new** GridLayout(0, 1, 1, 1));

panelEast.setBorder(**new** TitledBorder("Output Scale"));

radioButtonToCel = **new** JRadioButton("Celsius");

radioButtonToFar = **new** JRadioButton("Fahrenheit");

radioButtonToKel = **new** JRadioButton("Kelvin");

buttonGroupTo = **new** ButtonGroup();

buttonGroupTo.add(radioButtonToCel);

buttonGroupTo.add(radioButtonToFar);

buttonGroupTo.add(radioButtonToKel);

panelEast.add(radioButtonToCel);

panelEast.add(radioButtonToFar);

panelEast.add(radioButtonToKel);

outputLabel = **new** JLabel("OUTPUT");

textFieldOutputTemp = **new** JTextField(10);

panelSouth = **new** JPanel(**new** FlowLayout());

panelSouth.add(outputLabel);

panelSouth.add(textFieldOutputTemp);

textFieldOutputTemp.setEditable(**false**);

Container contentPane = frame.getContentPane();

contentPane.add(panelNorth, BorderLayout.***NORTH***);

contentPane.add(panelWest, BorderLayout.***WEST***);

contentPane.add(panelEast, BorderLayout.***EAST***);

contentPane.add(panelSouth, BorderLayout.***SOUTH***);

frame.setSize(400, 400);

frame.setVisible(**true**);

TextFieldHandler textFieldHandler = **new** TextFieldHandler();

radioButtonFromCel.addActionListener(textFieldHandler);

radioButtonFromFar.addActionListener(textFieldHandler);

radioButtonFromKel.addActionListener(textFieldHandler);

radioButtonToCel.addActionListener(textFieldHandler);

radioButtonToFar.addActionListener(textFieldHandler);

radioButtonToKel.addActionListener(textFieldHandler);

textFieldInputTemp.addActionListener(textFieldHandler);

ButtonHandler buttonHandler = **new** ButtonHandler();

// register the buttons

radioButtonFromCel.addItemListener(buttonHandler);

radioButtonFromFar.addItemListener(buttonHandler);

radioButtonFromKel.addItemListener(buttonHandler);

radioButtonToCel.addItemListener(buttonHandler);

radioButtonToFar.addItemListener(buttonHandler);

radioButtonToKel.addItemListener(buttonHandler);

}

// Event handler for input field enter key events

**private** **class** TextFieldHandler **implements** ActionListener {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**if** (textFieldInputTemp.getText().compareTo("") == 0) {

// put information in text box "setText"

textFieldOutputTemp.setText("No Input");

} **else** {

**boolean** radioButtonFromCelIsSelected = radioButtonFromCel.isSelected();

**boolean** radioButtonFromFarIsSelected = radioButtonFromFar.isSelected();

**boolean** radioButtonFromKelIsSelected = radioButtonFromKel.isSelected();

**boolean** radioButtonToCelIsSelected = radioButtonToCel.isSelected();

**boolean** radioButtonToFarIsSelected = radioButtonToFar.isSelected();

**boolean** radioButtonToKelIsSelected = radioButtonToKel.isSelected();

**if** ((radioButtonFromCelIsSelected == **false** && radioButtonFromFarIsSelected == **false**

&& radioButtonFromKelIsSelected == **false**)) {

JOptionPane.*showMessageDialog*(**null**, "Please select input scale");

**return**;

} **else** **if** ((!"Celsius".equals(e.getActionCommand()) && !"Fahrenheit".equals(e.getActionCommand())

&& !"Kelvin".equals(e.getActionCommand())) && radioButtonToCelIsSelected == **false**

&& radioButtonToFarIsSelected == **false** && radioButtonToKelIsSelected == **false**) {

JOptionPane.*showMessageDialog*(**null**, "Please select output scale");

**return**;

} **else** {

updateDisplay();

}

}

}

}

**private** **class** ButtonHandler **implements** ItemListener, ActionListener {

@Override

**public** **void** itemStateChanged(ItemEvent e) {

**if** (textFieldInputTemp.getText().compareTo("") == 0) {

textFieldOutputTemp.setText("No Input");

}

}

@Override

**public** **void** actionPerformed(ActionEvent ev) {

}

}// end ButtonHandler

**private** **void** updateDisplay() {

**double** input = 0;

**double** dOutput = 0;

**try** {

input = Double.*parseDouble*(textFieldInputTemp.getText());

} **catch** (Exception e) {

JOptionPane.*showMessageDialog*(**null**, "Please enter valid input");

**return**;

}

**if** (radioButtonFromCel.isSelected() && radioButtonToFar.isSelected()) {

dOutput = TemperatureConversionFormulae.*CelsiustoFahrenheit*(input);

textFieldOutputTemp.setText(String.*format*("%.2f %c F", dOutput, 176));

}

**if** (radioButtonFromCel.isSelected() && radioButtonToKel.isSelected()) {

dOutput = TemperatureConversionFormulae.*CelsiustoKelvin*(input);

textFieldOutputTemp.setText(String.*format*("%.2f K", dOutput));

}

**if** (radioButtonFromCel.isSelected() && radioButtonToCel.isSelected()) {

textFieldOutputTemp.setText(String.*format*("%.2f %c C", input, 176));

}

**if** (radioButtonFromFar.isSelected() && radioButtonToCel.isSelected()) {

dOutput = TemperatureConversionFormulae.*FahrenheittoCelsius*(input);

textFieldOutputTemp.setText(String.*format*("%.2f %c C", dOutput, 176));

}

**if** (radioButtonFromFar.isSelected() && radioButtonToKel.isSelected()) {

dOutput = TemperatureConversionFormulae.*FahrenheittoKelvin*(input);

textFieldOutputTemp.setText(String.*format*("%.2f K", dOutput));

}

**if** (radioButtonFromFar.isSelected() && radioButtonToFar.isSelected()) {

textFieldOutputTemp.setText(String.*format*("%.2f %c F", input, 176));

}

**if** (radioButtonFromKel.isSelected() && radioButtonToCel.isSelected()) {

dOutput = TemperatureConversionFormulae.*kelvinToCelsius*(input);

textFieldOutputTemp.setText(String.*format*("%.2f %c C", dOutput, 176));

}

**if** (radioButtonFromKel.isSelected() && radioButtonToFar.isSelected()) {

dOutput = TemperatureConversionFormulae.*KelvintoFahrenheit*(input);

textFieldOutputTemp.setText(String.*format*("%.2f %c F", dOutput, 176));

}

**if** (radioButtonFromKel.isSelected() && radioButtonToKel.isSelected()) {

textFieldOutputTemp.setText(String.*format*("%.2f K", input));

}

}

}

**package** lab3;

**class** TemperatureConversionFormulae

{

**public** **static** **double** CelsiustoFahrenheit(**double** inputTemp)

{

**double** outputtemp= ((9.0/5.0) \* inputTemp) + 32;//Celsius to Fahrenheit equation;

**return** outputtemp;

}

**public** **static** **double** CelsiustoKelvin(**double** inputTemp)

{

**double** outputtemp= inputTemp + 273.15;//Celsius to Kelvin equation;

**return** outputtemp;

}

**public** **static** **double** FahrenheittoCelsius(**double** inputTemp)

{

**double** outputtemp= (inputTemp-32)\*(5.0/9.0);//Fahrenheit to Celsius equation;

**return** outputtemp;

}

**public** **static** **double** FahrenheittoKelvin(**double** inputTemp)

{

**double** outputtemp= (inputTemp +459.67)\*(5.0/9.0);//Fahrenheit to Celsius equation;;

**return** outputtemp;

}

**public** **static** **double** kelvinToCelsius(**double** inputTemp)

{

**double** outputtemp= inputTemp -273.15;//Kelvin to Celsius equation;

**return** outputtemp;

}

**public** **static** **double** KelvintoFahrenheit(**double** inputTemp)

{

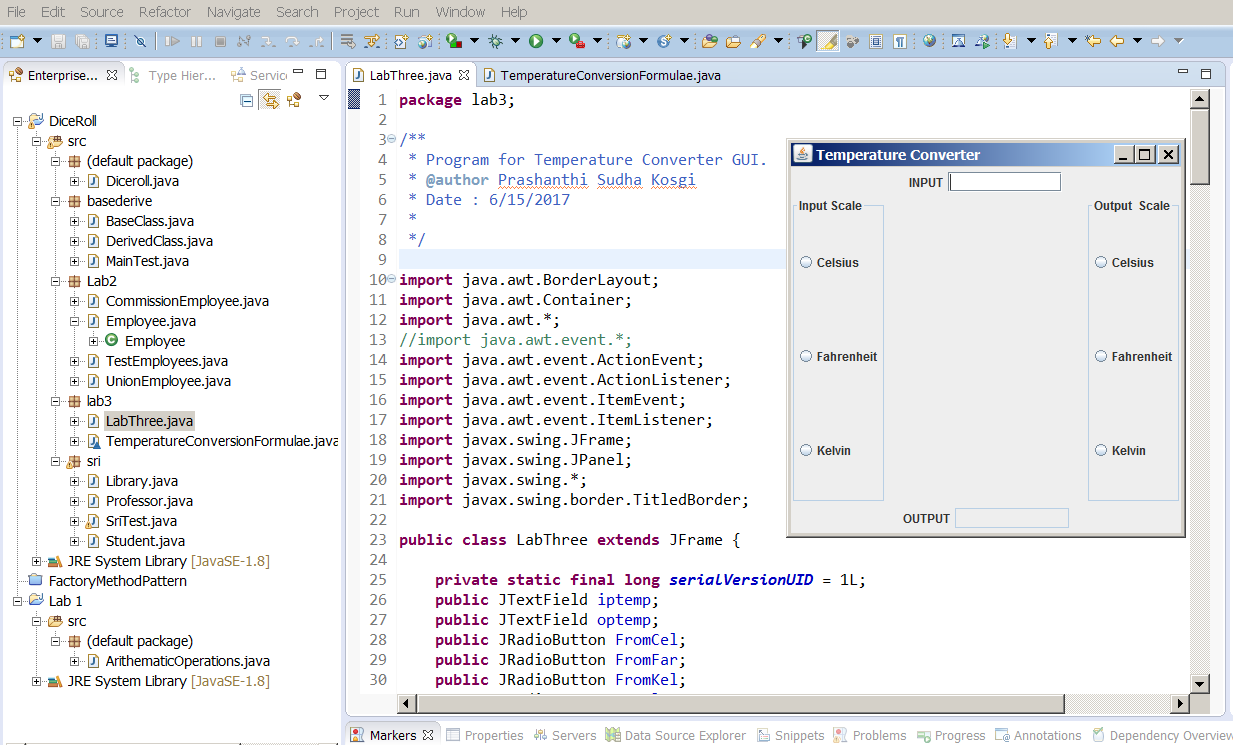
**double** outputtemp= (inputTemp\*(9.0/5.0))-459.67;//Kelvin to Fahrenheit;

**return** outputtemp;

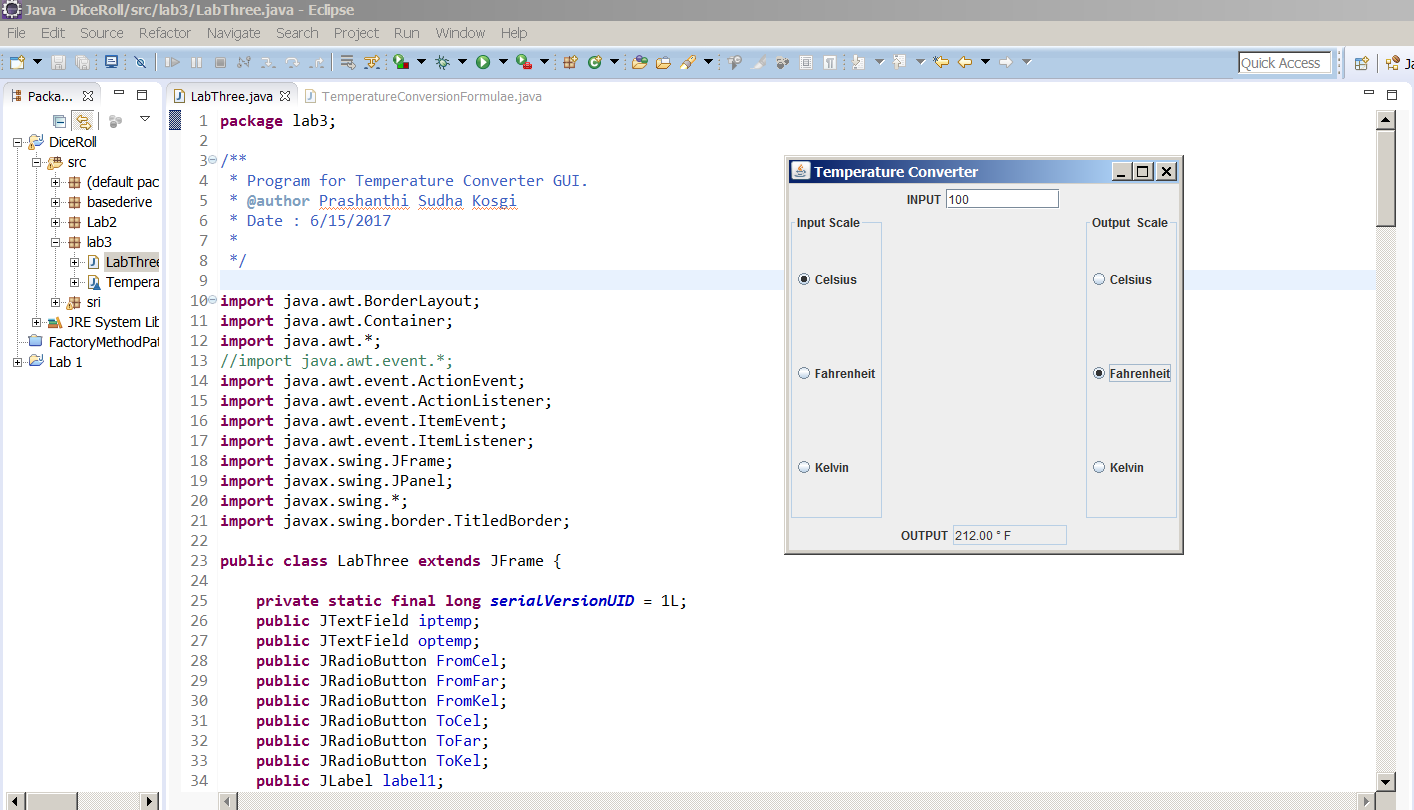
}

}

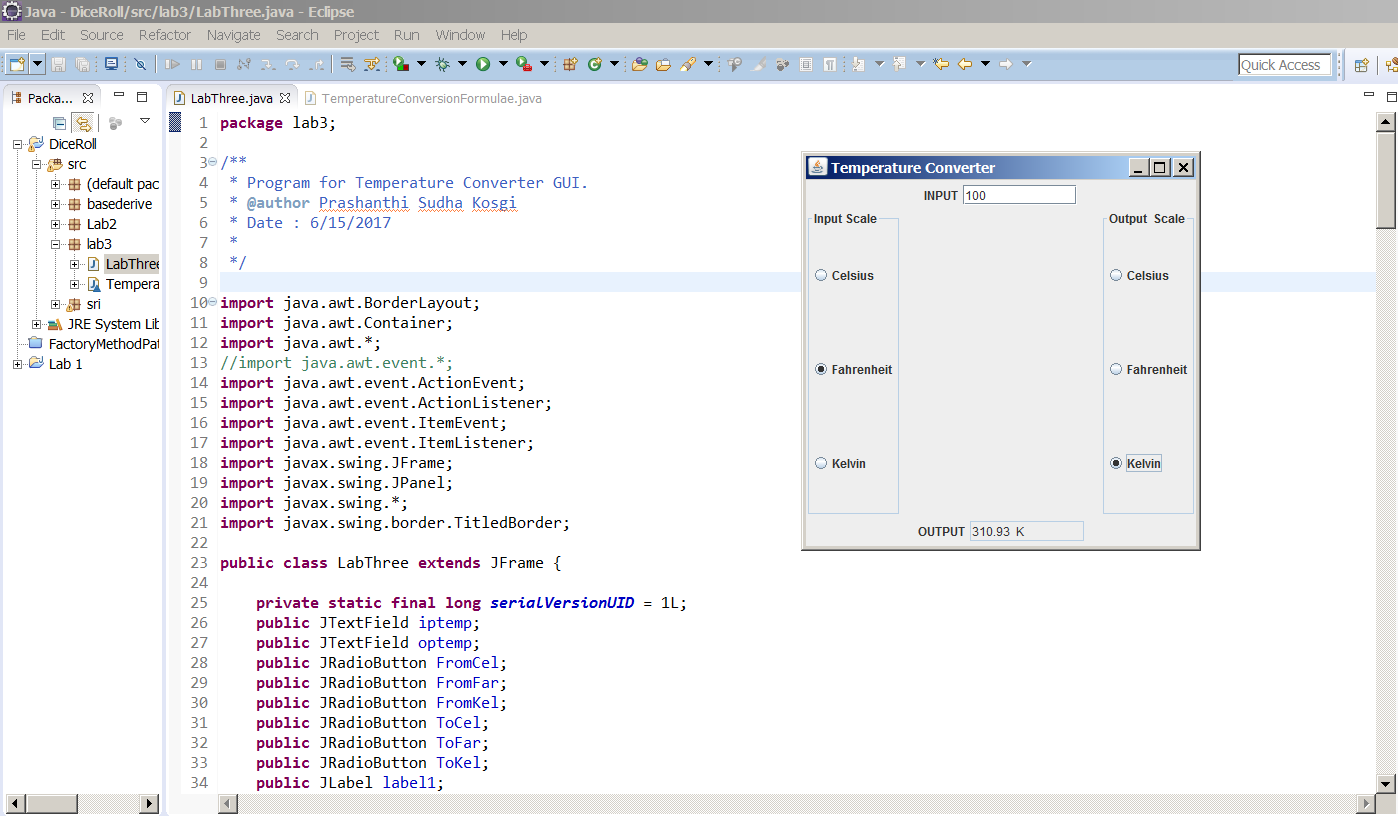
**Screen shot with no input.**



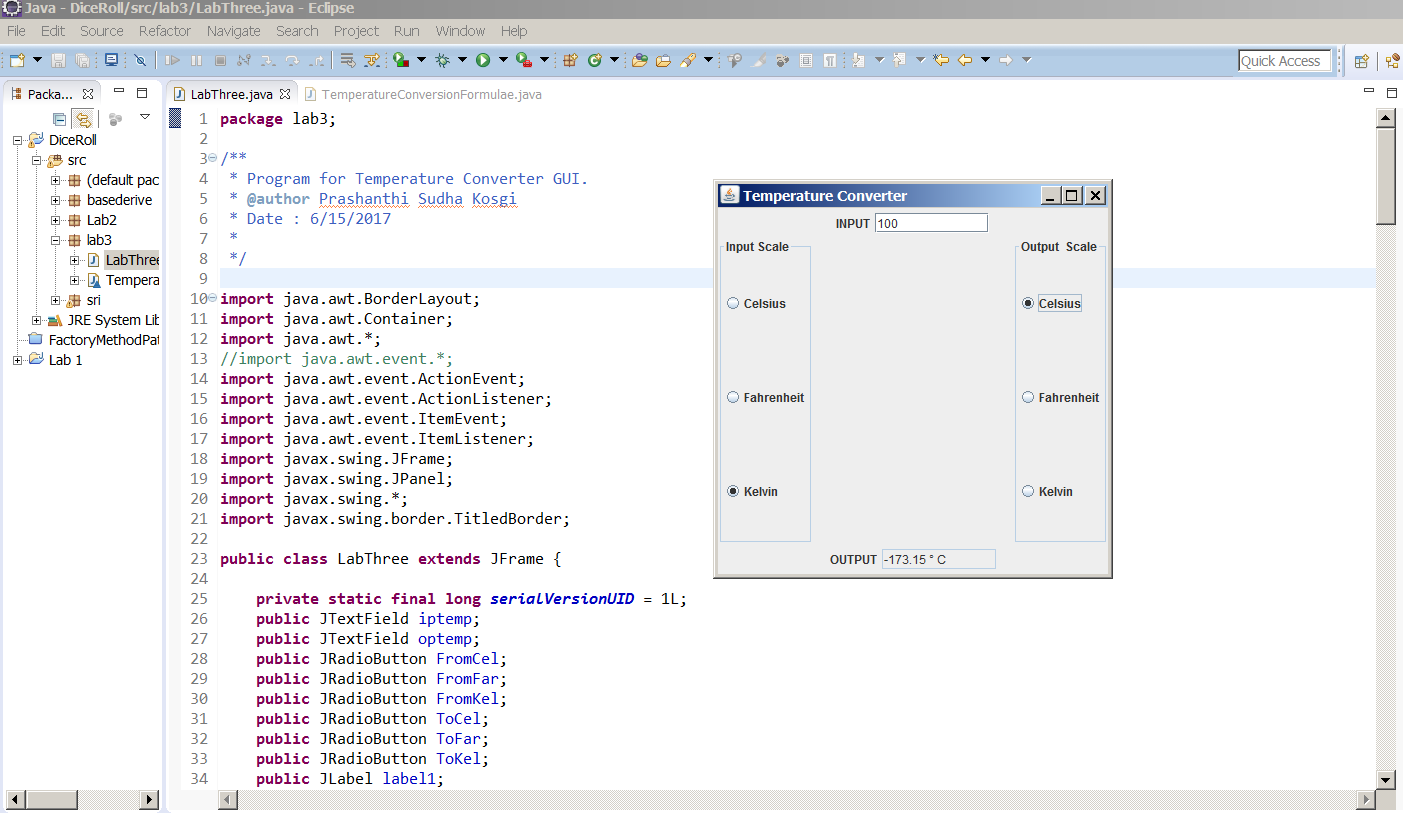
**Screen shot for Celsius to Fahrenheit.**



**Screen shot for Fahrenheit to Kelvin.**

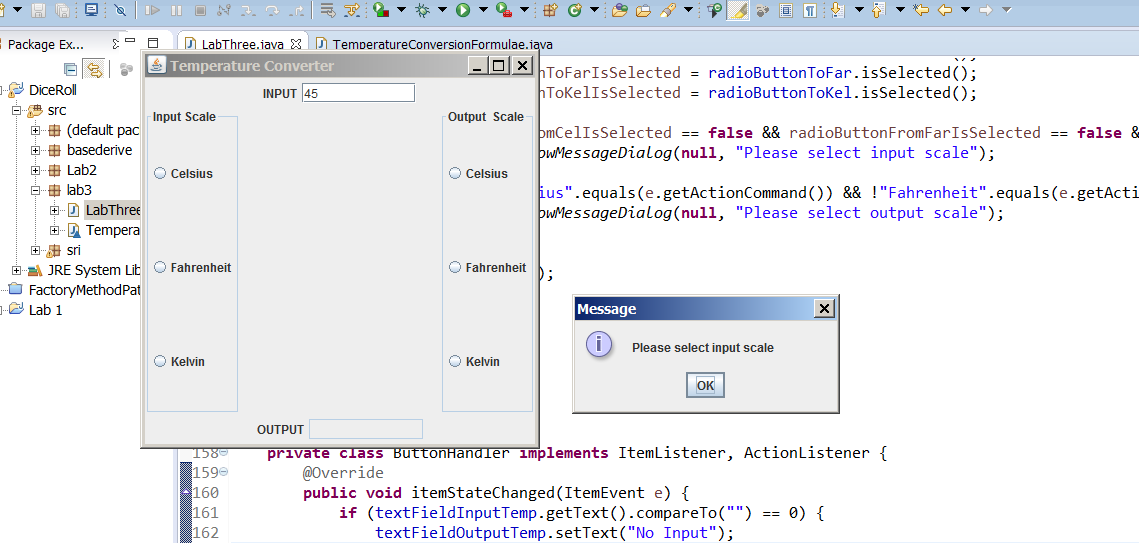


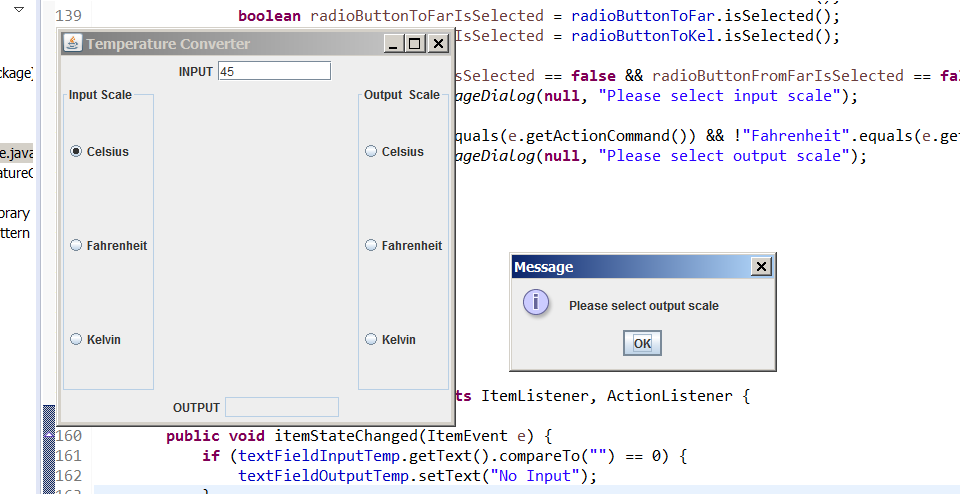
**Screen shot for Kelvin to Celsius.**



**Error case handing:**

**1.**



**2.** 

**3.**

