LABSHEET 08 – REPORT

FC110531 – A. K. G. N. DESHAPRIYA

CCS|CSE1513 - Objected Oriented Programming (Practical)

SOURCE CODES :

1.
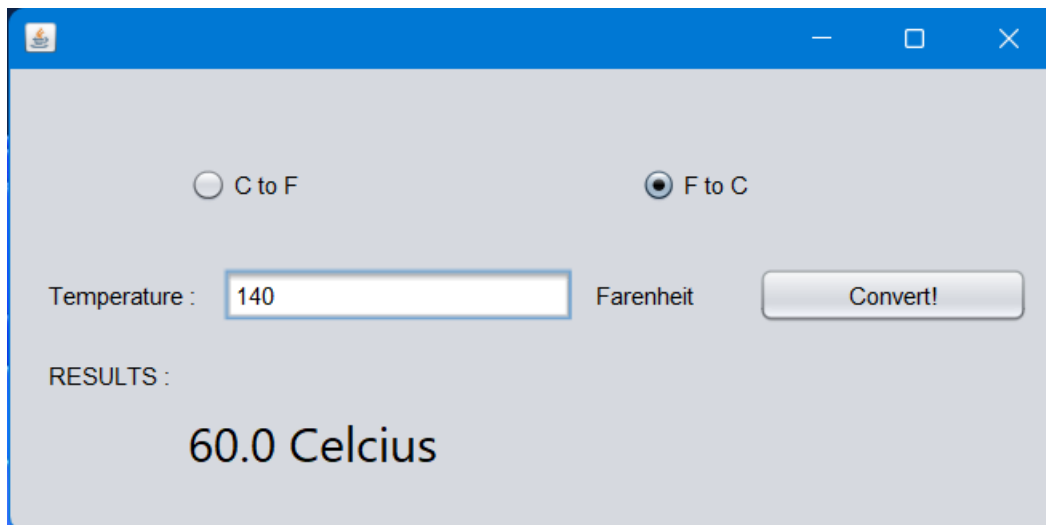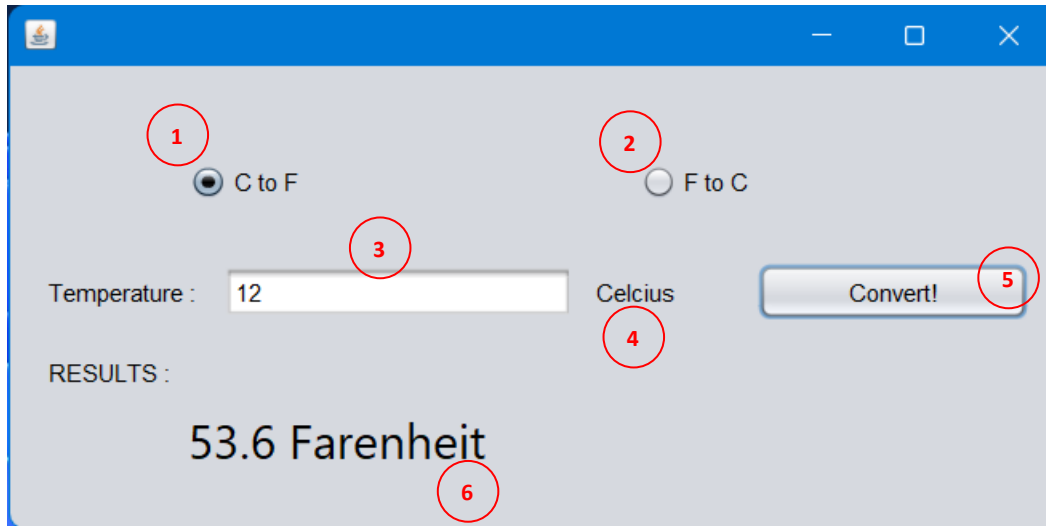
This temperature converter is a simple GUI application developed with Java using the Swing library. The purpose of this application is to convert temperatures between Celsius and Fahrenheit.

Used GUI components

1. ***CtoFradio***: JRadioButton - Radio button for selecting Celsius to Fahrenheit conversion.
2. ***FtoCradio***: JRadioButton - Radio button for selecting Fahrenheit to Celsius conversion.
3. ***tempTxtInput***: JTextField - Text field for entering the temperature value.
4. ***tempstatelabel***: JLabel - Label for displaying the current temperature state (Celsius or Fahrenheit).
5. ***ConvertButton***: JButton - Button for triggering the temperature conversion.
6. ***resultVal***: JLabel - Label for displaying the converted temperature result.

## Special Functions/Enhancements used

- Input Validation:

The ***ConvertButtonActionPerformed*** method includes **a try-catch block to handle exceptions** that may occur during the conversion process.

Input validation ensures that the entered temperature is a valid numerical value. If an exception occurs, an error message is displayed to the user. You may try it out.

- Dynamic State Label:

The ***tempstatelabel*** label dynamically updates to indicate the current temperature state, whether in Celsius or Fahrenheit. This provides a visual cue for the user regarding the selected temperature unit.

2.
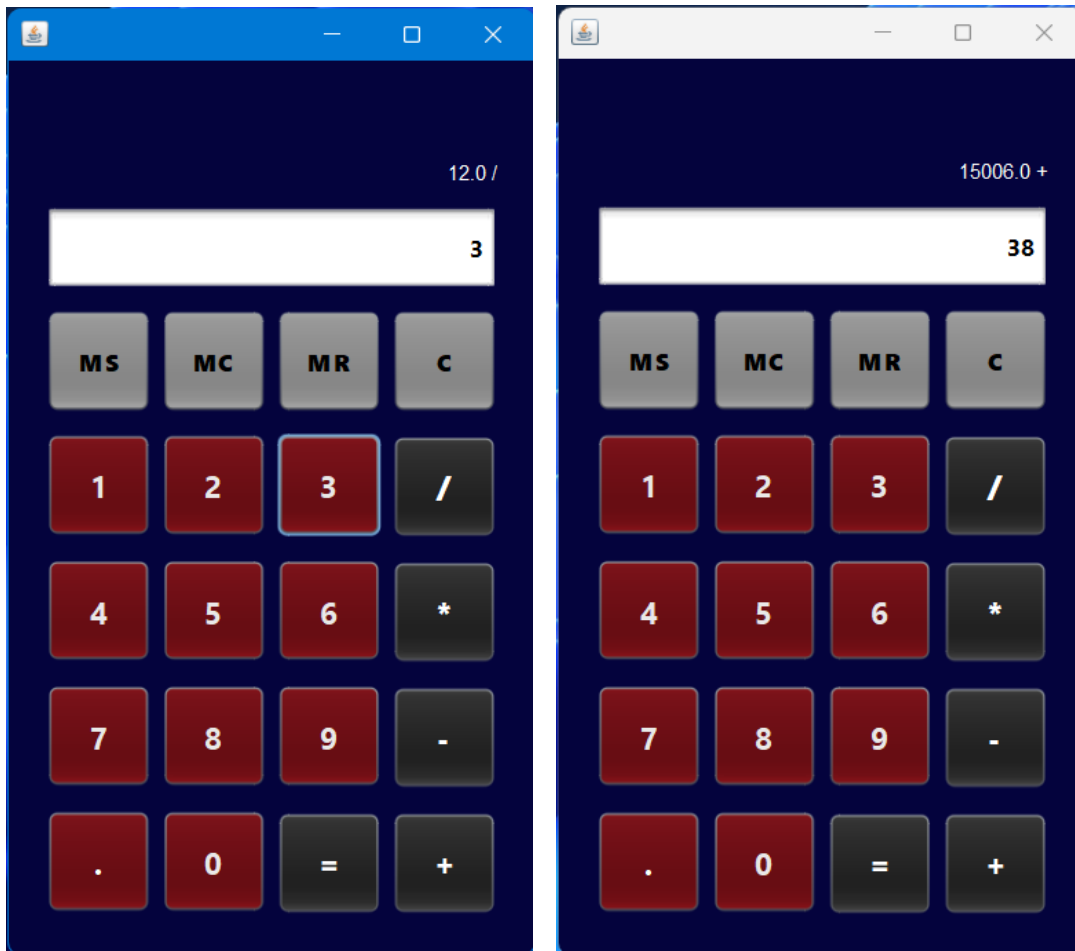
This program represents the implementation of a simple calculator using the Java Swing library for creating graphical user interfaces. The calculator application allows users to perform basic arithmetic operations such as addition, subtraction, multiplication, and division. Additionally, it features memory functionalities, including memory store (MS), memory recall (MR), and memory clear (MC).

Used GUI components

- ***button0*** to ***button9***: Numeric buttons (0-9) used for entering digits in the calculator.
- ***buttonDot***: Decimal point button for entering floating-point numbers.
- ***buttonCE***: Clear Entry button, used to remove the last entered digit or operator.
- ***buttonMS***: Memory Store button, saves the current value displayed in numberField to the memory (memo) variable.
- ***buttonMR***: Memory Recall button, retrieves the value stored in the memory variable and displays it in numberField.
- ***buttonMC***: Memory Clear button, clears the memory variable (memo).
- ***divideButton***: Division operation button, sets the division operator for subsequent calculations.
- ***multiplyButton***: Multiplication operation button, sets the multiplication operator for subsequent calculations.
- ***minusButton***: Subtraction operation button, sets the subtraction operator for subsequent calculations.
- ***plusButton***: Addition operation button, sets the addition operator for subsequent calculations.
- ***equalsButton***: Equals button, calculates the result of the operation based on the selected operator.
- ***numberField***: Text field for displaying input and results of calculations.
- ***lastLabel***: Label for displaying the last operation performed, including the first operand and operator.

Special Functions/Enhancements used

- Memory Operations:

Memory Store (MS): Allows users to store the current displayed number in memory for later recall.

Memory Recall (MR): Retrieves the value stored in memory and displays it in the text field.

Memory Clear (MC): Clears the memory, removing any stored value.

- Error Handling:

In *equalsButtonActionPerformed* The code incorporates basic error handling using **a try-catch block**. attempts to convert user input from the text field to a double and do the calculation, and catching any exceptions that may occur.

- Clear Entry (CE) Button:

The Clear Entry button removes the last character from the displayed number, providing a way for users to correct input mistakes. Works like a backspace.

- Dynamic Display:

The calculator dynamically updates a label *lastLabel* to display the ongoing calculation. For example, when an arithmetic operator button is pressed, it shows the first operand and the pressed operator.

- Decimal Point Handling:

Allows users to input decimal numbers, enhancing precision in calculations.