# **MILITARY COLLEGE OF SIGNALS, NUST**

A red circle with white text and a book and lightning bolt

Description automatically generatedA blue and white logo

Description automatically generated

**Object Orientated Programming**

**Assignment**

**SUBMITTED TO: Dr Tauqeer**

**Ahmed Talal Sajid**

**Capt Shaarif**

**Capt Samiullah**

**Capt Haider**

**COURSE: BESE-29 A**

**Program Overview**

**User Interface (UI):**

The calculator UI is built using Swing components (JFrame, JPanel, JButton, JTextArea). It has a black background with white text for the main display (screen) and uses a custom font (DejaVuSansMono-Bold.ttf) for consistent look and feel.

**Button Layout:**

Buttons are arranged in a grid layout (5 rows x 8 columns) within btnPanel. Each button corresponds to a symbol or function (symbols array), including digits, operators, mathematical functions (sin, cos, etc.), and constants (π, e).

1. **User Interface (UI)**:
   * The calculator UI is built using Swing components (JFrame, JPanel, JButton, JTextArea).
   * It has a black background with white text for the main display (screen) and uses a custom font (DejaVuSansMono-Bold.ttf) for consistent look and feel.
2. **Button Layout**:
   * Buttons are arranged in a grid layout (5 rows x 8 columns) within btnPanel.
   * Each button corresponds to a symbol or function (symbols array), including digits, operators, mathematical functions (sin, cos, etc.), and constants (π, e).
3. **Functionality**:
   * **Numeric Input**: Users can input numbers (0-9) and decimal points (.).
   * **Basic Operations**: Addition (+), Subtraction (-), Multiplication (x), Division (÷), and Modulus (%).
   * **Advanced Operations**: Trigonometric functions (sin, cos, tan), logarithms (log, ln), power functions (x^2, x^3, √x, ∛x), factorial (x!), and more.
   * **Memory and Clearing**: Memory operations (AC for all clear, ⌫ for backspace), sign change (+/-), and reciprocal (1/x).
   * **Constants**: Mathematical constants (π, e).
   * **Evaluation**: Pressing = evaluates the expression entered in the screen.
4. **Event Handling**:
   * Buttons register actions using ActionListener.
   * Different actions are handled in actionPerformed(ActionEvent e) method, where the appropriate operation or function is executed based on the button pressed.
5. **Error Handling**:
   * Basic error handling is implemented to prevent invalid operations, such as division by zero or factorial of negative numbers.
   * Exceptions like font loading failures (FontFormatException, IOException) are caught and managed with fallback options.

### Improvements and Recommendations

1. **UI Enhancement**:
   * Consider refining the layout to enhance user experience, such as grouping related functions and improving spacing.
   * Implement tooltips or labels for buttons to provide clarity on their functions.
2. **Function Expansion**:
   * Expand functionality to include more advanced mathematical operations or scientific calculations.
   * Improve precision handling or format output dynamically based on user preferences.
3. **Code Optimization**:
   * Refactor repetitive code where possible to improve maintainability and reduce redundancy.
   * Utilize constants or enums for button symbols to enhance readability and ease of updates.
4. **User Feedback**:
   * Implement visual feedback for operations (e.g., highlighting the current operation being performed).
   * Provide error messages or notifications for invalid inputs or operations.
5. **Localization**:
   * Support multiple languages or locales by externalizing string literals and considering cultural conventions for numeric formatting and symbols.

### Conclusion

The Calculator program provides a comprehensive scientific calculator with a user-friendly interface and robust functionality. It leverages Java's Swing library for GUI development and incorporates mathematical operations and functions essential for scientific calculations. By refining UI design, expanding functionality, optimizing code, and enhancing user feedback, the calculator can be further improved to meet diverse user needs and preferences/