Scientific Calculator

A PROJECT REPORT

Submitted by

DINESH KARTHIK K (2303811710421301)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **" Scientific Calculator"** is the bonafide work of **DINESH KARTHIK K (2303811710421301)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING
Dr.A.DELPHIN CAROLINA RANI, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR

[illegible]

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

CGB1201-JAVA PROGRAMMING
Mrs.K.VALLI PRIYADHARSHINI, M.E.,(Ph.D.)
SUPERVISOR
ASSISTANTP ROFESSOR

[illegible]

SIGNATURE

Mrs.K.Valli Priyadharshini, M.E.,(Ph.D.,),

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on3/12/24

CGB1201-JAVA PROGRAMMING
Mr.MALARMANNAN A, M.E.,
INTERNAL EXAMINER
ASSISTANT PROFESSOR

[illegible]

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Ms. P.ARUNA PRIYA, M.E.,
EXTERNAL EXAMINER
ASSISTANT PROFESSOR
8104-DSEC, PERAMBALUR.

[illegible]

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"Scientific Calculator"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

Signature

R. Dineshkarthik

R.Dinesh karthik

Place: Samayapuram

Date:3.12.24

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs. K. VALLI PRIYADHARSHINI, M.E., (Ph.D.,),** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The **Scientific Calculator** is a Java-based application with a graphical user interface (GUI) implemented using the AWT (Abstract Window Toolkit) framework. It supports basic arithmetic operations like addition, subtraction, multiplication, and division, as well as scientific functions such as sine, cosine, and tangent calculations. The layout consists of a text display at the top for showing input and results, and a grid of buttons for numbers, operators, and functions. The application uses event-driven programming to handle user input via button clicks.

The calculator processes inputs by maintaining two operands and an operator for basic calculations, and it dynamically updates results on button presses. It includes error handling to display "Error" in case of invalid operations. The program also provides a "C" button to clear all inputs and reset the state. The window is designed to close gracefully, ensuring user convenience. This calculator serves as a simple and effective tool for mathematical computations with an intuitive interface

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The **Scientific Calculator** is a Java-based application with a graphical user interface (GUI) implemented using the AWT (Abstract Window Toolkit) framework. It supports basic arithmetic operations like addition, subtraction, multiplication, and division, as well as scientific functions such as sine, cosine, and tangent calculations. The layout consists of a text display at the top for showing input and results, and a grid of buttons for numbers, operators, and functions. The application uses event-driven programming to handle user input via button clicks.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1Objective

The objective of the Scientific Calculator program is to provide a user-friendly graphical interface for performing basic arithmetic operations (addition, subtraction, multiplication, and division) and scientific calculations (sine, cosine, and tangent). It aims to simplify mathematical computations for users through an intuitive layout and efficient functionality while demonstrating the use of Java's AWT framework for GUI design and event handling. Additionally, it seeks to handle errors gracefully, maintain computational accuracy, and offer a versatile tool suitable for educational and practical purposes in solving mathematical problems.

1.2Overview

The **Scientific Calculator** is a Java-based application designed to perform both basic and scientific mathematical operations. Built using the AWT framework, it features an intuitive graphical user interface with a text display for inputs and results, along with a grid of buttons for numbers, arithmetic operators, and trigonometric functions.

The calculator supports operations like addition, subtraction, multiplication, division, and trigonometric functions (sin, cos, tan) using angle values in degrees. It employs event-driven programming, where button clicks trigger specific actions. Additional features include a clear ("C") button for resetting inputs and a display mechanism for error messages in case of invalid inputs.

This application provides a simple yet powerful tool for performing everyday mathematical tasks while also serving as a practical example of AWT-based GUI programming in Java.

1.3 Java Programming Concepts

Object-Oriented Programming Concepts (OOPs)

The project effectively demonstrates key principles of object-oriented programming as follows:

Encapsulation
The Main class encapsulates all the data and methods required for the chat application's functionality. This includes GUI components like TextField and TextArea and networking objects such as Socket, ensuring that the internal implementation details are hidden while providing a simple interface for interaction.

Inheritance
The Main class extends Frame, inheriting properties and methods from Java's AWT framework. This inheritance facilitates the creation of the graphical user interface by leveraging pre-defined features of the Frame class.

Polymorphism
Method overriding is used to achieve polymorphism, such as in the actionPerformed method, where it is overridden to handle button click events in a customized manner specific to the chat application.

Abstraction
The project abstracts the complexities of socket communication and GUI interactions by encapsulating these details within the Main class. This allows users to focus on essential functionalities, like sending and receiving messages, without worrying about the underlying implementation.

Project-Related Java Concepts

GUI Design: Utilizes AWT components such as TextField, Button, and Panel to create an interactive interface.

Event-Driven Programming: Responds to user interactions through event listeners (ActionListener and WindowAdapter).

Mathematical Functions: Leverages the Math class for trigonometric operations (sin, cos, tan) and angle conversion (Math.toRadians).

Grid Layout: Arranges buttons in a 5x4 grid layout for intuitive user navigation.

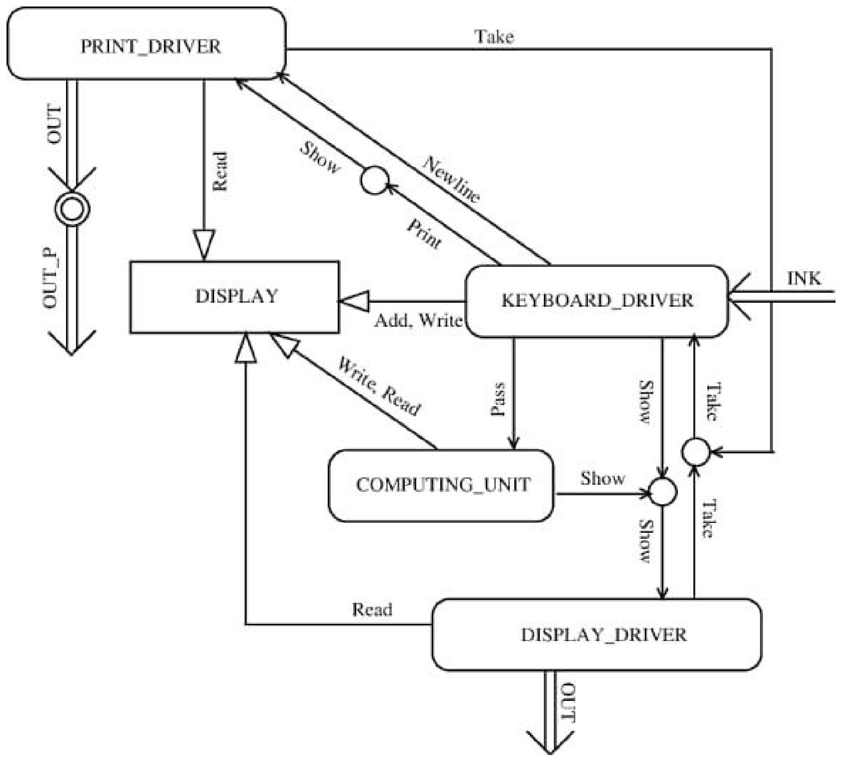
Error Handling: Ensures invalid inputs or operations (like division by zero) do not crash the application but instead display an error message.

CHAPTER 2
PROJECT METHODOLOGY

2.1Proposed Work

The proposed work aims to enhance the **Scientific Calculator** application by adding advanced mathematical functions such as logarithms, square roots, exponents, and hyperbolic trigonometric functions (sinh, cosh, tanh). Memory and history features will be introduced to store, recall, and review previous calculations, making the tool more versatile. The graphical user interface (GUI) will be upgraded with modern layouts or frameworks like JavaFX, along with theming options such as a dark mode for improved aesthetics. Enhanced error handling will provide detailed messages for specific issues like invalid inputs or division by zero, and input validation will prevent user mistakes. The application will support keyboard shortcuts for quicker operations and be packaged as a standalone executable for cross-platform compatibility. Additionally, an educational mode with tooltips and explanations for functions will be incorporated to make the application more user-friendly and informative.

2.2 Block Diagram



CHAPTER 3
MODULE DESCRIPTION

3.1 User Interface Module:

This module focuses on the design and layout of the calculator's graphical user interface (GUI). It includes components like the display screen, buttons for numbers and operations, and the panel layout (e.g., GridLayout). The module ensures a user-friendly, visually appealing, and responsive interface, potentially integrating modern frameworks like JavaFX for advanced styling and theming options.

3.2 Arithmetic and Scientific Operations Module:

This module focuses on the design and layout of the calculator's graphical user interface (GUI). It includes components like the display screen, buttons for numbers and operations, and the panel layout (e.g., GridLayout). The module ensures a user-friendly, visually appealing, and responsive interface, potentially integrating modern frameworks like JavaFX for advanced styling and theming options.

3.3 Event Handling Module:

Responsible for managing user interactions, this module listens for button clicks or keyboard inputs and triggers the appropriate actions. It employs Java's ActionListener and WindowAdapter interfaces to process commands like number entry, operation selection, and result display. It ensures smooth and responsive user experiences through event-driven programming.

3.4Error Handling and Validation Module:

This module ensures the application handles invalid inputs gracefully. It validates user entries, preventing errors such as entering multiple decimal points or performing operations on empty inputs. It also displays meaningful error messages for invalid operations like division by zero or unsupported input formats, ensuring a robust and error-free application.

3.5Memory and History Module:

This module allows users to store, recall, and review previous calculations. Memory functions such as M+, M-, MR, and MC provide quick access to stored values, while a calculation history feature displays past results. This module enhances usability by helping users track and reuse their computational data efficiently.

CHAPTER 4
CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

In conclusion, the **Scientific Calculator** application demonstrates the integration of Java programming concepts, object-oriented principles, and GUI design to create a robust and user-friendly tool for mathematical computations. By supporting both basic arithmetic and advanced scientific operations, the application serves a wide range of user needs. The proposed enhancements, including additional functions, improved interface, memory and history features, and better error handling, aim to further elevate its functionality and user experience. This project not only simplifies complex calculations but also serves as a practical example of efficient application development in Java, showcasing the potential for future scalability and adaptability.

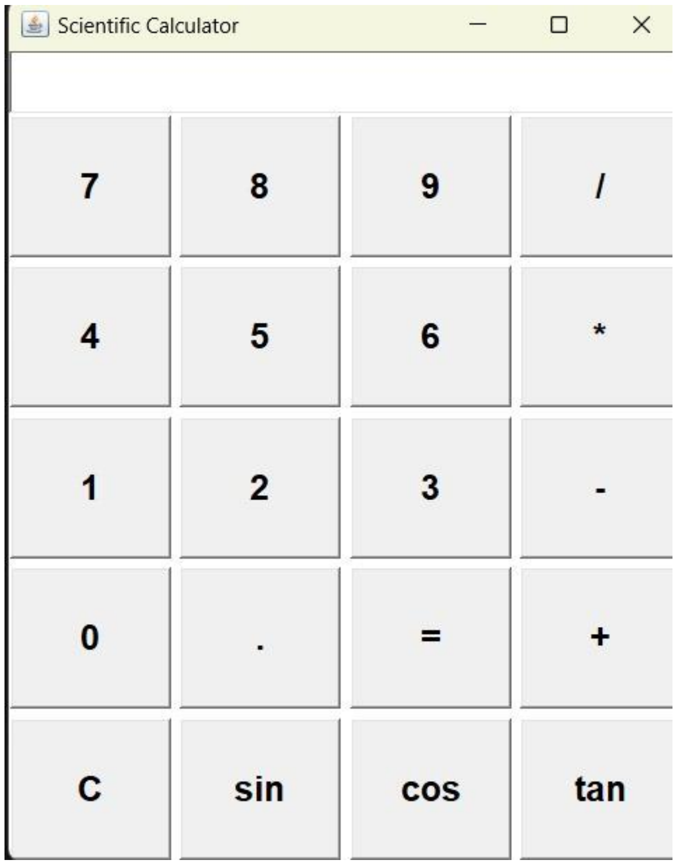
4.2 FUTURE SCOPE

The future scope of the **Scientific Calculator** application offers exciting opportunities for growth and enhancement. One potential development is the adaptation of the calculator for mobile platforms, allowing users to access its functionality on smartphones and tablets. Integrating voice commands for hands-free operation could increase accessibility, especially for users with disabilities. Additionally, adding graphing capabilities would transform the calculator into a powerful tool for visualizing mathematical functions, benefiting students and professionals in fields like engineering and data science. Cloud integration could enable users to synchronize their data and access it across devices, providing a seamless experience. Incorporating AI to suggest calculations based on user behavior would make the application more intelligent and user-friendly. Expanding its features to support advanced mathematics, such as matrix operations, calculus, and complex numbers, would make it even more valuable for higher-level users. Finally, allowing users to customize the interface, from button arrangements to themes, would further personalize the tool.

APPENDIX A
(SOURCE CODE)

```
import java.awt.*;
import java.awt.event.*;

public class ScientificCalculator extends Frame implements ActionListener {
    TextField display;
    double num1 = 0, num2 = 0, result = 0;
    char operator;

    public ScientificCalculator() {
        // Frame setup
        setTitle("Scientific Calculator");
        setSize(400, 500);
        setLayout(new BorderLayout());

        // Display
        display = new TextField();
        display.setFont(new Font("Arial", Font.BOLD, 24));
        add(display, BorderLayout.NORTH);

        // Buttons Panel
        Panel panel = new Panel();
        panel.setLayout(new GridLayout(5, 4, 5, 5));

        // Buttons
        String[] buttons = {
                "7", "8", "9", "/",
                "4", "5", "6", "*",
            "1", "2", "3", "-",
            "0", ".", "=", "+",
            "C", "sin", "cos", "tan"
        };

        for (String text : buttons) {
            Button button = new Button(text);
            button.setFont(new Font("Arial", Font.BOLD, 20));
            button.addActionListener(this);
            panel.add(button);
        }
        add(panel, BorderLayout.CENTER);

        // Window closing
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });

        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();

        try {
            if (command.matches("[0-9]") || command.equals(".")) {
                display.setText(display.getText() + command);
            } else if (command.equals("C")) {
                display.setText("");
                num1 = num2 = result = 0;
                operator = '\0';
            } else if (command.equals("=")) {
                num2 = Double.parseDouble(display.getText());
                switch (operator) {
                    case '+': result = num1 + num2; break;
                    case '-': result = num1 - num2; break;
                    case '*': result = num1 * num2; break;
                    case '/': result = num1 / num2; break;
                }
                display.setText(String.valueOf(result));
            } else if (command.equals("sin") || command.equals("cos") ||
command.equals("tan")) {
                double value = Math.toRadians(Double.parseDouble(display.getText()));
                switch (command) {
                    case "sin": result = Math.sin(value); break;
                    case "cos": result = Math.cos(value); break;
                    case "tan": result = Math.tan(value); break;
                }
                display.setText(String.valueOf(result));
            } else {
                num1 = Double.parseDouble(display.getText());
                operator = command.charAt(0);
                display.setText("");
            }
        } catch (Exception ex) {
            display.setText("Error");
        }
    }

    public static void main(String[] args) {
        new ScientificCalculator();
    }
}
```

APPENDIX B
(SCREENSHOTS)



REFERENCES

- **Books**
  - *"Head First Java"* by Kathy Sierra and Bert Bates: A fun, visual guide for beginners to learn Java programming concepts.
  - *"Java: The Complete Reference"* by Herbert Schildt: Covers both basic and advanced topics, making it a go-to resource for Java learners.

· **Websites**

- GeeksforGeeks: Offers Java tutorials, coding examples, and interview preparation materials.
- Oracle Official Java Documentation: The authoritative source for Java features, APIs, and updates.

· **YouTube Channels**

- *Programming with Mosh*: Provides beginner-friendly Java tutorials with clear explanations and examples.
- *Telusko*: Covers Java topics in depth, including advanced concepts and frameworks.

· **Online Platforms**

- Codecademy: Interactive lessons to practice Java coding in real time.
- Udemy: Offers affordable courses on Java with practical projects and lifetime access.