K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS), TRICHY.

CALCULATOR

PRESENTED
Dinesh Karthik R
2303811710421301

SUPERVISOR
Mrs.Valli priyadharshini
M.E,(Phd),
AP/CSE

PRESENTATION OVERVIEW

1. **Objective**
2. **Project Introduction**
3. **Problem Statement**
4. **Methodologies (Programming concepts relevant to problem statement)**
5. **Architecture of the proposed system**
6. **List of Modules**
7. **Merits**
8. **Results and Discussion**
9. **Queries**

OBJECTIVE

The objective of this project is to develop a **Calculator Application** that can perform basic arithmetic operations like addition, subtraction, multiplication, and division. Additionally, the application can be extended to include advanced operations like square roots, exponents, and other mathematical functions. The goal is to provide a user-friendly interface and to implement the functionality efficiently using relevant programming techniques.

PROJECT INTRODUCTION

This project aims to create a software tool capable of performing both basic and complex mathematical calculations. It serves as a virtual calculator where users can input numbers and perform operations using buttons or a keyboard interface. The application can be designed for use on both desktop and mobile platforms and can support both graphical (GUI) and command-line interfaces (CLI).

PROBLEM STATEMENT

- While handheld calculators and software applications exist, they may lack flexibility or advanced features. Moreover, some users may want a customizable and lightweight calculator that provides enhanced functionalities such as history tracking, scientific calculations, or user-specific features.
- This project seeks to address these problems by designing a calculator that:
- Supports both basic and advanced operations.
- Provides a user-friendly and efficient interface.
- Allows future customization and extensibility.
- Can be used across multiple platforms (desktop or mobile).

METHODOLOGIES

•**Object-Oriented Programming (OOP)**: To design the application, OOP concepts like classes and objects will be utilized to represent different operations (addition, subtraction, etc.), along with a main Calculator class to coordinate these operations.

•**Event Handling**: For GUI-based applications, event-driven programming will be used to handle user input from buttons or other UI elements. Each button press will trigger an event (such as an operation or number being added to the current input).

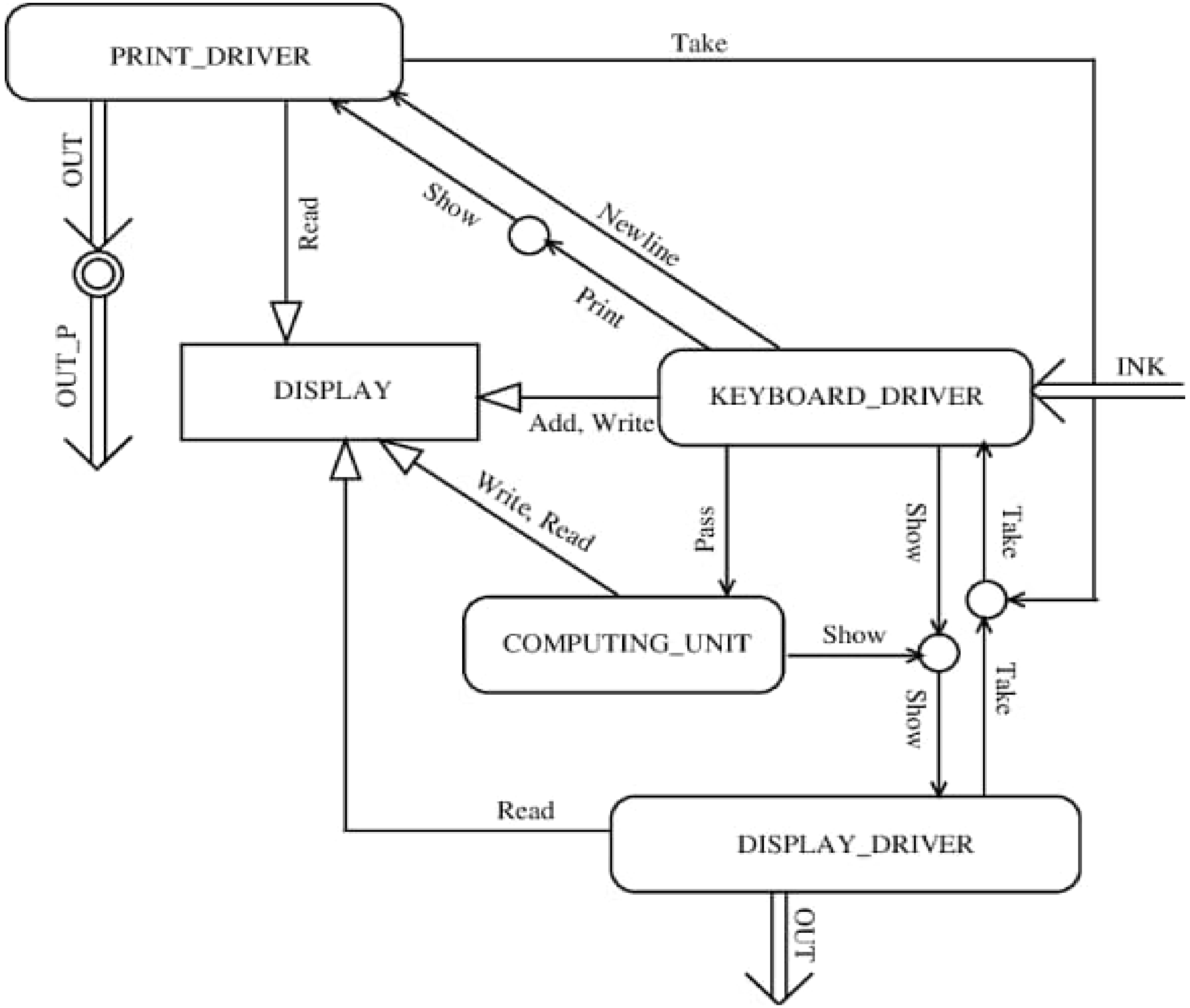
•**Recursion**: This can be used in advanced operations like factorials or solving complex mathematical problems.

•**Error Handling**: Proper error handling will be implemented to manage edge cases, such as division by zero or invalid input.

•**Data Structures**: Arrays or lists can be used to store the history of operations or to implement an operation stack.

•**Mathematical Libraries**: External libraries (e.g., math in Python) can be used for advanced functions like square root, logarithms, and trigonometry.

ARCHITECTURE OF THE PROPOSED SYSTEM



LIST OF MODULES

MODULE DESCRIPTON

- **User Interface (UI) Module**
- Design the interface for button interactions.
- Displays current inputs and output results.

- **Calculation Module**
- Performs basic and advanced mathematical operations.
- Handles mathematical computations using algorithms or libraries.

- **Error Handling Module**
- Manages invalid operations (e.g., divide by zero).
- Provides feedback to the user on errors.

- **History/Memory Module**
- Stores a log of previous calculations.
- Allows users to retrieve or clear history.

- **Testing Module**
- Contains test cases to verify the accuracy of each function.
- Ensures error cases are managed correctly.

MERITS

- **Ease of Use**: The calculator will have an intuitive interface, making it accessible to both beginners and advanced users.

- **Customizability**: Users can modify or extend the calculator to include new features or operations.

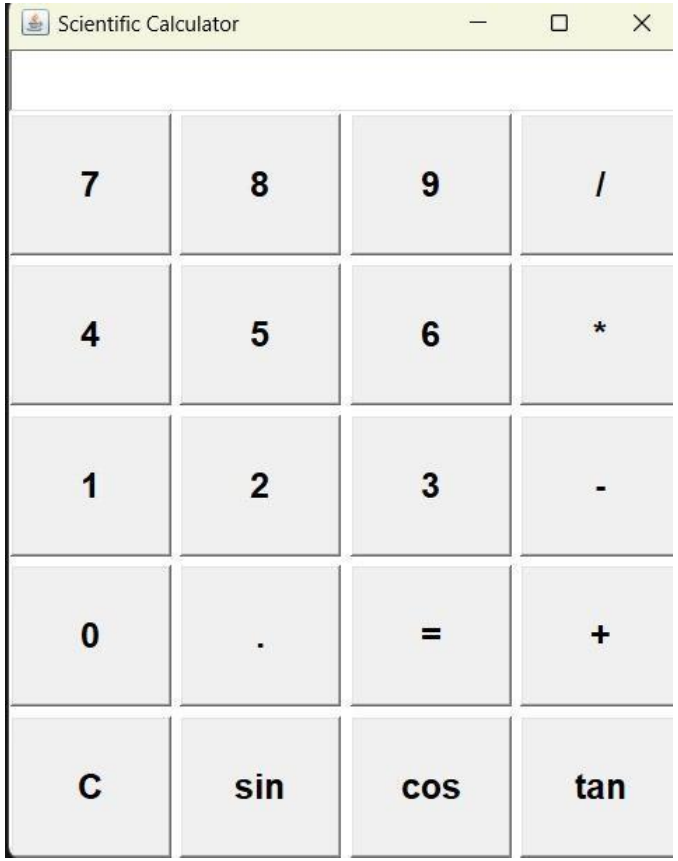
- **Platform Independence**: The application can be designed to work on different platforms (desktop, mobile, web).

- **Advanced Operations**: Beyond basic arithmetic, the calculator can be extended to scientific, financial, and statistical functions.

- **Error Handling**: Comprehensive error handling to prevent crashes or unexpected behavior.

- **History Functionality**: Users can track their past calculations for reference.

RESULTS AND DISCUSSION



QUERIES ?