INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Project - 1 **MA517M-Basic Programming Laboratory** Last Date: 09 November 2025
Name **Roll No.: MA25M008**

Square Matrices using C++ Classes and Operator Overloading

Objective: To design a C++ program that represents square matrices using classes, computes determinant, inverse (if exists), and supports addition and multiplication using operator overloading.

> A square matrix is a matrix with the same number of rows and columns, $n \times n$. Operations such as determinant and inverse are defined only for square matrices.

Problem Description

Design a class `SquareMatrix` to represent an n $n \times n$ matrix. The class should enable the computation of the determinant, inverse (if it exists), and support arithmetic operations such as addition and multiplication using operator overloading.

Class Specification

- **Class Name:** `SquareMatrix`
- **Private Data Members:**
  - `int n;`
  - `vector<vector<double>> values;`
  - `double determinant() const;`
  - `SquareMatrix inverse() const;`
- **Public Member Functions:**
  - `SquareMatrix(int size, vector<vector<double>> vals);`
  - `void display() const;`
  - `int size() const;`

Operator Overloading

- `operator +()` Adds two matrices of the same size
- `operator *()` Multiplies two matrices
- `operator ==(), !=()` Compares matrices for equality
- `operator <<()` Displays the matrix in a readable form

Tasks

1. Create two 3×3 matrices A and B
2. Compute $A + B$ and $A \times B$ using overloaded operators
3. Compute the determinant of A and B
4. Compute the inverse of A (if it exists)
5. Compare A and B using equality operators
6. Display all results using the overloaded `<<` operator

Expected Output Example

```
Matrix A:
1 2 3
0 1 4
5 6 0

Matrix B:
2 0 1
1 2 3
4 5 6

A + B:
3 2 4
1 3 7
9 11 6

A * B:
16 17 21
16 17 19
22 10 23

det(A) = 1
det(B) = -3

Inverse of A:
-24 18 5
20 -15 -4
-5 4 1

A == B : False
```

Project - 2: Number Slide Game Using C++ Classes

Problem Statement

Design and implement a **Number Slide Game** using **C++ classes**. The program should simulate a 4×4 sliding puzzle where the goal is to arrange numbers from 1 to 15 in order, leaving one empty space. The project should utilize object-oriented programming concepts such as classes, objects, encapsulation, and methods for handling game logic.

Project Requirements

1. Create a `Matrix` class to represent the 4×4 game board.
   (a) Include a method to generate a random matrix of size 4×4 using `srand()` in the range [1, 16].
   (b) Include a method to check that all entries of the matrix are distinct.
   (c) Remove the entry with the value 16 to represent the empty space.
2. Create a `Game` class to manage gameplay.
   (a) Display options for the user: `Play` or `Solution`.
   (b) If the user chooses `Play`, show the navigation commands:
   - `A/a` for Left
   - `S/s` for Down
   - `D/d` for Right
   - `W/w` for Up
   - `Q/q` for Quit

   (c) When the user presses `Q/q`, confirm again before quitting.
   (d) After each move, display the updated matrix in a nicely formatted way.
3. If the user chooses `Solution`, display the sequence of movements leading to the solved puzzle, and show the final output.
4. Ensure proper encapsulation of game logic and board operations within the respective classes.

Suggested Class Structure

1. `Matrix` Class:
   - Data member: 4×4 integer array representing the board
   - Methods: generateMatrix(), checkDistinct(), removeEmpty(), displayMatrix()
2. `Game` Class:
   - Data member: `Matrix` object, user choice
   - Methods: playGame(), showSolution(), processMove(char move), confirmQuit()

Reference

For more details about the game, visit: http://www.artbylogic.com/puzzles/numSlider/numberShuffle.htm