Project – Phase 1

1- Objective

The aim of this project is to compare and analyze the behavior of the different numerical methods used for solving system of linear equations:

1. Gauss Elimination.
2. Gauss Jordan.
3. LU Decomposition.
4. Gauss Seidel.
5. Jacobi Iteration.

2- Description

You are required to implement linear system of equation program which takes as an input the system of linear equations, the method to use and its required parameters if exists for this method.

3- Specification

You're required to implement an interactive GUI application that does the following:

1. Accepts an input for a system of linear equations:
    a. The equations can be of any format.
    b. The input validation should be bullet-proof.
    c. Number of variables must equal the number of equations.
    d. Coefficients must be numbers (0 or non-existing is allowed).
2. Choose any of the previously mentioned methods to solve the given equation via a drop-down list.
3. Parameters -if it applicable for the chosen solving method- (will be explained).
4. A way to enter the precision. (If user chooses not to provide it, a default value must be applied)
5. A Solve button to display the output if exists, the run time must be displayed.
6. Always apply partial pivoting if applicable.

The following table summarizes the required methods and their input parameters:

Method	Input	Parameters
Gauss Elimination	System of Linear Equations	None
Gauss-Jordan	System of Linear Equations	None
LU Decomposition	System of Linear Equations	Drop-down list for the format of L & U: 1- Doolittle Form 2- Crout Form 3- Cholesky Form
Gauss-Seidel	System of Linear Equations	1- Initial Guess 2- Stopping Condition: a. Number of Iterations b. Absolute Relative Error
Jacobi-Iteration	System of Linear Equations	1- Initial Guess 2- Stopping Condition: a. Number of Iterations. b. Absolute Relative Error.

4- Bonus

You can deliver **any** of the following and get a bonus:

1. Single step mode simulation showing each step of the algorithm.
2. Coefficients can be letters and express the output in term of this letters, for example X1 = a*a-b/ c.
3. Scaling as explained in the lecture.

5- Deliverables

You must deliver the following:

1. A fully commented and clean code. Make sure you use best OOP programming practices in your implementation. You may use any programming language.
2. A well-formatted report that contains the following:
    a. Flowchart or pseudo-code for each method.
    b. Sample runs for each method; they must include different cases (normal and tricky cases).
    c. Comparison between different methods (time complexity, convergence, best and approximate errors)
    d. Data structure used and how helpful was your choice.

6- Notes

Please note the following:

1. You should work in teams of 5 persons.
2. Code plagiarism detection will be applied and if a violation is detected, the cheating policy will be applied.
3. This same code base will be extended in Phase 2 by adding other methods to solve non-linear equations.