**ECE 4960**

**Programming Assignment – 2**

**Modular Testing in Sparse Matrix Solvers**

**BY:**

**Vishisht Tiwari (vmt28)**

**Deepak Agarwal (da475)**

**Goal:** The goal of the assignment is to understand and implement different matrix solvers for both full and sparse matrices and implement as comprehensive testing methods as possible.

**Code Structure:**

The parent directory contains following files:

* Code Files:

1. Assignment2\_Header.h: It is a global header file included in all code files, containing globally defined functions and other classes declataions.

Class Global\_Functions: Global class defined in the header file containing functions to create dynamically allocated matrix from an ar    ray, create dynamic arrays, create full matrix from sparse etc. Exception handling check is added in all functions under a macro EXCEPTION\_    HANDLING

Class Matrix\_Operations: Class declared having matrix operations (permute, scaling and product) for full and sparse matrices

Class Jacobi: Class implementing iterative solver using Jacobi method on full and sparse matrices

Class LoadMat1: Class implementing functions to load csv files and generate sparse matrix from it

1. Assignment2\_Main.cpp: Main file loading the sparse matrix from the .csv files, performing Jacobi iterative solver and calculating the norm.
2. Matrix\_Operations.cpp: Implements the class Matrix\_Operations containing the three matrix operations (permute, scaling and product) for sparse and full matrice    s, with exception handling under EXCEPTION\_HANDLING macro.
3. Jacobi.cpp: Class implementing functions for iterative solver for full-matrix and row-compressed formats.
4. Load\_Mat1.cpp: Implements functions for loading the .csv files in different arrays.

* Report documents:

1. Report.pdf: Documentation of code design and testing strategies
2. Readme.txt: File describing code structure, compilation command and testing platforms
3. Output.txt: Log output on MacOS

* Data:

1. CSV files for mat1

**Testing Strategy:**

Following a modular programming approach, a class of globally defined testing functions was defined in the header file, so that they can be called by any other class.