**Data Structure & Algorithm**

**Individual Assignment – Solving Systems of Linear Equations Using Matrix (weighting: 30%)**

**Problem Description**

Vector and matrix are data structures that are widely used in data science applications. In this assignment, you are required to implement them in **abstract data types (ADT) with Python class** and to apply matrix operations to solve systems of linear equations **without using any Python libraries for vector and matrix operations such as NumPy**. **Submissions that do not implement the data structures in Python classes will fail the assignment**. The minimal amount of data and methods to be defined in each corresponding ADT are as follows:

Class Vector:

Data:

vector – vector represented in a list

size – number of elements in vector

Methods:

add(aVector) – vector of aVector will be added to the vector of the current instance of Vector with the result being stored in a new instance of Vector. The new instance will be returned.

length() – return the length of vector (a non-negative numeric value)

dot\_product(aVector) – dot product of vector of aVector and the current vector, which is a numeric value, will be returned.

Class Matrix:

Data:

matrix – matrix represented in a list of lists

row\_nos – number of rows of the matrix

col\_nos – number of columns of the matrix

Methods:

add(aMatrix) – matrix of aMatrix is added to the matrix of the current instance of Matrix with the result being stored in a new instance of Matrix. The new Matrix instance will be returned.

isSquare() – test whether the matrix of the current instance of Matrix is a square matrix. A Boolean value will be returned.

det() – return the determinant of the matrix of the current instance of Matrix which is a numeric value.

transpose() – return the transpose of the matrix of the current instance of Matrix as a new Matrix instance.

inverse() – return the inverse of the matrix of the current instance of Matrix as a new Matrix instance.

In addition to defining and demonstrating the working of the Vector and Matrix classes, **a brief discussion on the worst-case time complexity of each method is to be included** too. **The submission must also include a case that uses the Matrix class to solve a system of linear equations.**

**Assignment Requirements**

The program must be developed in Python classes. Special purpose Python modules such as NumPy are not allowed to use. A report of no more than 10 pages (excluding any appendices) that describes the program design with illustrations of the proper working of each class method and their time complexities is needed. A full listing of program codes should be included in an appendix.

**Submission Requirements**

Submit your Python program (in .py or .ipynb extension) which should be properly commented, and the corresponding report to Moodle no later than 15 May 2022.

This assignment amounts to 30% of the course grade. Grading will be based on solution correctness and completeness (65%), the efficiency of the submitted solution (10%), and the quality of the documentation (15%).