# Lab 6 implement a 3D Matrix class.

Implement all the functions required by a Matrix.

https://github.com/PeterLowe/Lab6

An initial project is available from github for starting with, it contains the toString method and a default constructor and the member variables for the class. There is a sample test routine for checking that the constructor worked correctly.

You should add the line one at a time from the PDF (by reading & understanding the line before retyping it into your project. Then implement the necessary code for that line and append to the test method to ensure your code works. You will normally need to run the test a few times with different values to ensure correctness, you can do this by editing the test and re-compiling. You should leave the last version of the test for each line in the main.cpp file to show Pete that you do write code properly. When you run a test you need to work out the desired answer independently of your code, either by hand or a set of online resources. {While creating this assignment I discovered an online matrix inversion site that was wrong, but failed to say broken on the start page}.

You will also need to add an appropriate amount of [quality] comments.

* Do not break your code base, if it doesn’t compile then Zero grade.
* Do not change the definition of the methods then when I add my test code it won’t compile and Zero Grade.
* Do not retype blindly the entire header file and then spend the rest of the project trying to fix the problems caused by stupidly thinking it would be easier. Any declared but undefined methods will result in a Zero grade. {they are the green underlines}

There is no need for a research document as you all know how to perform these matrix operations and if not you must arrive at the same conclusion as the internet (good parts).

Save a zip file “Lab6**YourName**.zip” to your M: drive before 22:00 Wednesday 22nd February.

Below are some notes on the operations on 3x3 matrices,

The sum of two 3 × 3 matrices is defined as follows:

If **A** =  and if **B** =  then

**A + B =** .

**Multiplication**

Recall, if **A** =  and if **B** =  and **V = **then

k**A** = , a 3 × 3 matrix

**A.B =** 

**And A.V =  a vector**

Determinant and Inverse:

if **A** =  then the inverse of **A** exists as long as **detA is not zero**. It is found using the formula:

**A-1 = **

Where **detA =** A11(A22 A33 – A32A23) – A21(A33A12 – A32A13) + A31(A23A12 – A22A13)

If **** then the inverse is: 