24-780 Engineering Computation
Problem Set 11

You need to create a ZIP file (It may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Blackboard course. The file name of the ZIP file must be:

PS11-YourAndrewID.zip

For example, if your Andrew account is *hummingbird@andrew.cmu.edu*, the file name must be:

PS11-hummingbird.zip

If your ZIP file does not comply with this naming rule, you will automatically lose 5% credit from this assignment. If we are not able to identify who submitted the file, you will lose another 5% credit. If we finally are not able to connect you and the submitted ZIP file, you will receive 0 point for this assignment. Therefore, please make sure you strictly adhere to this naming rule before submitting a file.

The ZIP file needs to be submitted to the 24-780 Blackboard course. If you find a mistake in the previous submission, you can re-submit the ZIP file with no penalty as long as it is before the submission deadline.

Notice that the grade will be given to the final submission only. If you submit multiple files, the earlier version will be discarded. Therefore, if you re-submit a ZIP file, the ZIP file MUST include all the required files. Also, if your final version is submitted after the submission deadline, late-submission policy will be applied no matter how early your earlier version was submitted.

Make sure you upload your Zip file to the correct location. If you did not upload your assignment to the correct location, you will lose 5%.

The ZIP file needs to include:

- C++ source file of your program (ps11.cpp)

Submission Due: 11/17 (Tue) 23:59

START EARLY!

Unless you are already a good programmer, there is no way to finish the assignment overnight.

PS11 Templated matrix class [ps11.cpp] (100 points)

(1) Write a templated NxM matrix class called MatrixTemplate, which

a. is a matrix of double-precision floating point, and

b. takes two template parameters, const int NR and const int NC (number of rows and columns), and

c. has a protected member variable that is an array of double with length of NC*NR,

d. has a public member function called Set that takes row, column, and value to be set, and

e. has a public member function called Value that takes row and column, and returns the value at the given row and column, and

f. has a public member function called Print that prints the values of the matrix on the console window, and

g. correctly uses const qualifiers.

(2) Inherit Matrixtemplate <4,4> to create a class called Matrix4x4, which has a public function called Transpose. As it says, it must transpose the matrix.

(3) Add your code in the following test program and make sure your matrix class works correctly.

5 extra points: If you are able to add a function Invert that inverts the matrix in Matrix4x4 class, you get 5 extra points. For extra credit, uncomment three lines in the test program so that the test program will demonstrate your Invert function.

```
#include <stdio.h>


int main(void)
{
    Matrix4x4 mat;
    const double v[]=
    {
        6.0,1.0,4.0,9.0,
        9.0,8.0,6.0,1.0,
        7.0,2.0,9.0,4.0,
        1.0,7.0,5.0,9.0
    };
    for(int i=0; i<16; ++i)
    {
        const int r=1+i/4;
        const int c=1+i%4;
        mat.Set(r,c,v[i]);
    }
    mat.Print();
    mat.Transpose();
    printf("\n");
    mat.Print();
    // If you go for extra credit, uncomment the following three lines.
    // printf("\n");
    // mat.Invert();
    // mat.Print();
    return 0;
}
```