Assignment B

Generating Random Inputs

- Where applicable, you may use the function shown below to generate input values randomly:

```
float random_data(float low, float hi)
{
    float r = (float)rand() / (float)RAND_MAX;
    return low + r * (hi - low);
}
```

- For example, you may use this function to populate the input matrices in Assignment 1, and initialize the input values in option pricing assignments.

Measuring Running Time using Chrono

- We use a timer to measure the execution times of a program.
- We have many choices to get such timing measurements. Code snippet below shows how to use chrono in the C++ Standard Library.

```
#include <chrono>

using namespace std::chrono;

int main()
{
   high_resolution_clock::time_point t1 =
       high_resolution_clock::now();

   do_the_works();

   high_resolution_clock::time_point t2 =
      high_resolution_clock::now();

   std::cout << "Elapsed time: " <<
      duration_cast<milliseconds>(t2 - t1).count() << " ms";

}
```

- Defined in chrono header.

Assignment B (C/C++)

- Due: June 17 by 6 PM CT.
- Write a function to multiply two NxN matrices.
- Use the function you wrote to multiply two 100x100 matrices.
- Measure the execution time.
- There are many ways to represent a matrix. Two options are shown below (next slide).
- You are not required to use techniques such as vectorization/multithreading for this assignment.
- Aim of this assignment is to get the students to think about performance and set the stage for week 1 lecture.
- As long as anyone makes a genuine and an honest attempt to solve this problem one will get full points for this assignment, even if the solution is not complete.

1. Use a third party library
   - Example: Eigen `http://eigen.tuxfamily.org/index.php?title=Main_Page`
   - Getting started with Eigen is easy: `https://eigen.tuxfamily.org/dox/GettingStarted.html`
2. Use std::vector:

```
using matrix = std::vector<std::vector<float>>;
```

or

```
typedef std::vector<std::vector<float>> matrix;
```

- Code snippet below shows
  - how to use a vector to represent a matrix (as defined above)
  - how to initialize it using random input values

```
int main()
{
    const int rows = 1000, columns = 1000;
    matrix m1;

    //allocate space for matrix elements
    m1.resize(rows);

    for (int i = 0; i < rows; ++i)
    {
        m1[i].resize(columns);
    }

    //populate the matrix using random values
    for (int i = 0; i < rows; ++i)
    {
        for (int j = 0; j < columns; ++j)
        {
            m1[i][j] = random_data(0, 10);
        }
    }
}
```