C++ Standard Library: Random Numbers with <random>

Ben Morgan

THE UNIVERSITY OF WARWICK

Pseudo-Random Number Generation

- Most scientific software makes use of random number generation at some level
  - https://en.wikipedia.org/wiki/Random_number_generation
- Generally you should use the interfaces provided by the framework of your project to guarantee consistent and reliable behaviour.
- If required, C++11 does provide a reasonable set of classes in the `<random>` library

<random>

- C++11 divides random number generation into three main areas
  - *Creating seeds (initialization) for the generator*
  - *Uniform random number generation in a range*
    - Provides Mersenne Twister and RANLUX “engines”
  - *Random number distributions*
    - Uses output of “engine”, returns random numbers drawn from, e.g. Normal Distribution

<random>

https://github.com/cpp-pg-mpags/mpags-cpp-extra

```cpp
#include <random>
#include <list>
#include <iostream>

int main(int, char**) {
  std::random_device seeder; // (May) use hardware to create seed value

  std::mt19937 engine(seeder()); //Mersenne Twister, with seed from seeder

  std::normal_distribution<> gauss(1.23, 2.5); // Normal, mu=1, sigma=2.5

  std::list<double> data;
  std::generate_n(std::back_inserter(data), 1000000,
                  [&gauss, &engine](){return gauss(engine);});

  double mu {std::accumulate(data.begin(), data.end(), 0.0)/data.size()};
  std::cout << "Mean : " << mu << "\n";
  return 0;
}
```