

Random Numbers in C++11

Exercises

- Give an outline description of how to obtain random numbers in modern C++

- What is the purpose of a random number engine?
- Which random number engine is usually the best one to use?

- What is the purpose of a distribution type?
- Which distributions are most useful when generating random numbers?
- What does "uniformly distributed" mean in the context of random numbers?

- Write a simple program which
 - a) Prints out ten integers with random values between 0 and 100
 - b) Prints out ten floating point numbers with random values between 0 and 1

- Explain how `random_device` generates random numbers
 - How does this differ from a random number engine?
- Why can we not assume `random_device` will work as expected on every system?
- Why is `random_device` not suitable for generating large quantities of numbers?
- What is the most common situation where `random_device` is used?

- Write a simple program which demonstrates the most common way to use `random_device`

- Why should we avoid `default_random_engine`?
- Why is it necessary to check the documentation before using `random_device`?
- Why is it good practice to make engine and distribution instances static?