CS 3460

Introduction to Random Number Generation

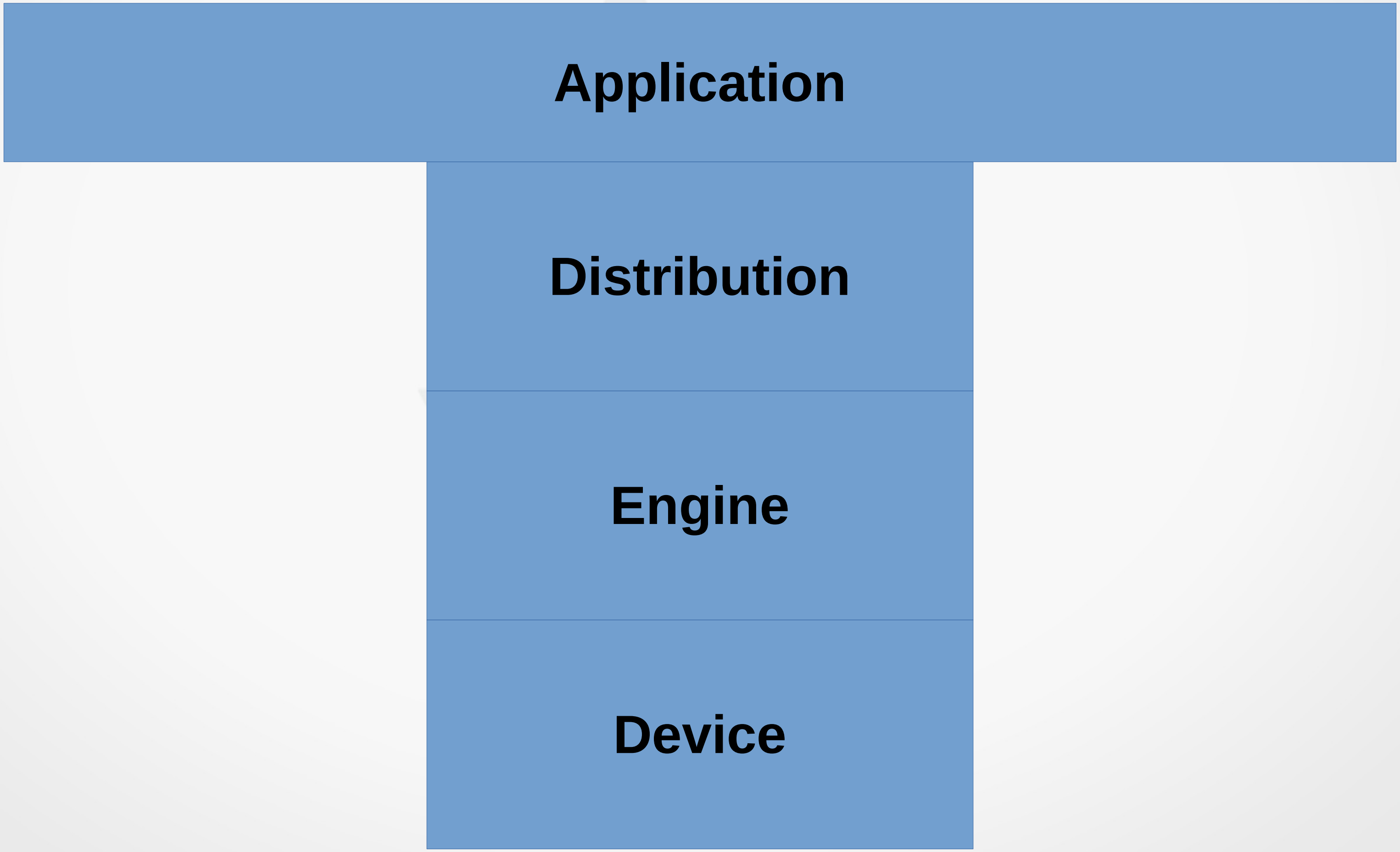
Random Number Generation

- Sophisticated capability
- “easily” generate different distributions
  - uniform, normal, exponential, plus more
- Part of the standard library `<random>`
- Legacy random number generation
  - `rand` and `srand`
  - don’t use them, just don’t!

Concepts

- Device
  - A non-deterministic uniform random number generator
- Engine
  - Pseudo random number generator
- Distribution
  - What kind of random number distribution
  - uniform, normal, binormal, exponential, plus more

Concepts – Fitting Together



Application

Distribution

Engine

Device

Random Device

- Uniform random number generator
  - Specifically; non-deterministic uniform random *bit* generator
- Could be a hardware device, might be software based
  - If software, pseudo-random number generator
- Implements the `()` operator
  - Updates its internal state
  - Returns a generated value
- Important Note: This isn't what you want your application to use for the random numbers. This is just something to provide entropy (initial seed) to a random number engine.

std::random_device – Code Demo

Random Number Engine

- Produces pseudo-random numbers, from an initial seed
- Implements the `()` operator; same as random device
- C++ provides several different engines
  - Linear Congruential
  - Mersene Twister
  - Subtract with carry

Random engine– Code Demo

Random Number Distributions

- Don’t always want uniform distributions, might want something else
  - Nature of application determines which to use
- Provided distribution categories
  - Uniform (2 types)
  - Bernoulli (4 types)
  - Poisson (5 types)
  - Normal (6 types)
  - Sampling (3 types)

Random distribution – Code Demo

Random Shuffle – Code Demo