

What's new (C++11)

Basics

1. double d {2.3}: use {} (list form) when in doubt: saves from conversions:
int i {7.8} // error;
2. auto b = true: auto defines the type (use if don't have a reason to specify the type explicitly);
3. constexpr: to be evaluated at compile time, used primarily to specify constants, to allow placement of data in memory where it is unlikely to be corrupted, and for performance;
4. range-for: for (auto x: v)..., where v is a vector;
5. enum class Color {red, blue, yellow}; enum class Apple{red, green}; Color z = Color::red, Apple a = Apple::red;
6. Exceptions: std::exception...; try...catch; static_assert: check constant expressions;

Abstraction Mechanisms

1. Initializing containers: Vector (std::initializer_list<int>);
2. Pure virtual functions:
class Container {
public:
virtual double& operator[] (int) = 0;
virtual int size () const = 0;
virtual Container () {}
};
3. unique_ptr, shared_ptr;
4. copy and move constructors and assignments;
5. Suppressing operations: Shape (const Shape&) = delete; used to suppress any operation;
6. Templates, function templates, function objects, variadic templates;
7. lambda expressions: [&] (int a){return a < x}, aliases;

Containers and Algorithms

- 1.