

Курс C++. Уровень: средний. Не для людей нулевого уровня. Рекомендуется: базовое знание хоть одного языка программирования, желательно C-подобного (C, C++, C#, Java). Если вы знаете чем `++i` отличается от `i++`, и что означает выражение `a>10 ? b-1 : c+1`, то проблем не будет.

1. Introduction -- Введение

Что такое C++? Зачем нужен C++? Компиляторы (C, C++) vs интерпретаторы (JS) vs bytecode (Java, Python, C#). Пример Hello World и простейший пример с `class`, `std::vector`. C++ компиляторы (g++, clang, CL) и IDE для разных платформ.

Литература.

2. C++ Language Basics 1 -- Основы языка 1

Версии C++ (98, 11, 14, 17). Built-in types (Встроенные типы). Operators (операции). Statements (команды). Пример использования CMake. `using/typedef`, `auto`, `decltype`.

3. C++ Language Basics 2 -- Основы языка 2

References (ссылки). Pointers (указатели). `const` modifier. Functions (функции). Overloaded functions (перегруженные функции). Headers (хедеры). Namespaces.

4. Classes basics -- Основы классов

Access modifiers (модификаторы доступа) : `public`, `private`, `protected`. Constructors. Destructors. Friends. Inheritance (наследование). Overriding (переопределение), virtual and abstract methods. Multiple inheritance. Diamond problem and virtual inheritance.

5. IO streams. Type casts. `enum class`.

Потоки ввода-вывода (IO streams). Type casts. Перечисляемые типы (`enum`, `enum class`).

6. Smart pointers. Exceptions -- Умные указатели. Исключения

`unique_ptr`, `shared_ptr`, `weak_ptr`. Object life cycle (Жизненный цикл объектов). Exceptions, try+catch statement.

7. Containers -- Контейнеры.

`vector`, `array`, `list`, `set`, `map`. Массивы. Итераторы. Алгоритмы.

8. Strings. Time and date. Random numbers. -- Строки. Дата и время. Случайные числа.

9. Function objects. Lambda expressions -- Объекты функция. Лямбда-выражения.

`std::function`, lambda expressions, function pointers, functors

10. Operator overloading. Copy/move constructors. -- Перегрузка операторов. Copy/move конструкторы.

`operator=`, `operator()`, `operator[]`, `operator+=` ...

11. Using cmake and make -- Использование cmake и make

Различные возможности CMake. Статические и динамические библиотеки. Основы make. Кратко: популярные внешние библиотеки (boost, opencv, opengl, etc.)

12. Concurrency -- Многопоточность

`async`, `future`, `promise`, `thread`, `packaged_task`, `atomic`, `mutex`, `condition_variable`

13. Templates -- Шаблоны.

14. Move semantics -- Семантика Move

Move operations, `std::move`, `std::forward`, rvalue references, universal references