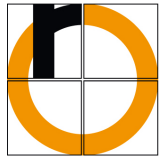


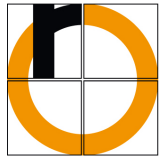


Programmieren technischer Anwendungen



◆ Termine:

- 9.10.17 → Einführung und Grundlagen
- 16.10.17 → Datentypen
- 23.10.17 → Klassen
- 30.10.17 → Vorlesung fällt aus
- 6.11.17 → Vererbung, Friends und Laufzeitpolymorphie
- 13.11.17 → Speicherverwaltung und Container
- 20.11.17 → Utilities in der STL
- 27.11.17 → Algorithmen in der STL
- 4.12.17 → Templates, constexpr und Compilezeitpolymorphie
- 11.12.17 → Nebenläufigkeit und Parallelität
- 18.12.17 → CI / CD / Tests in C++
- 8. 1.18 → Wie baut man eine Bibliothek
- 15. 1.18 → C++ 17/20 /Prüfungsvorbereitung
- 22. 1.18 → Prüfung



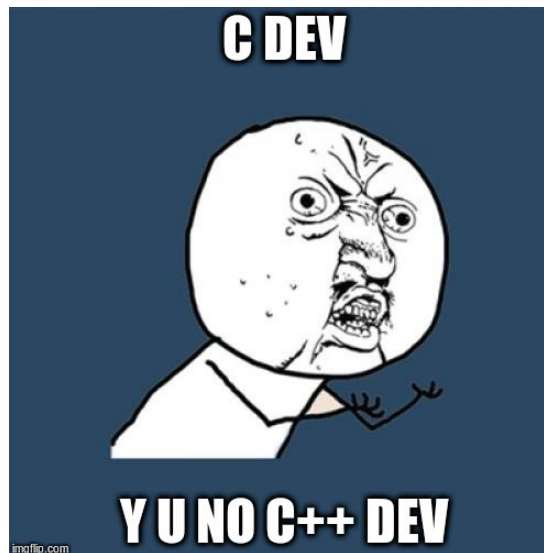
Was ist C++?

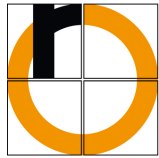


Was ist C++?

Wer hat schon mal C++ programmiert?

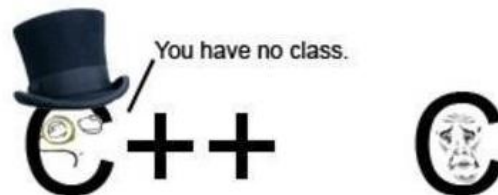
Und dabei kein C mit „Bonus“ geschrieben?



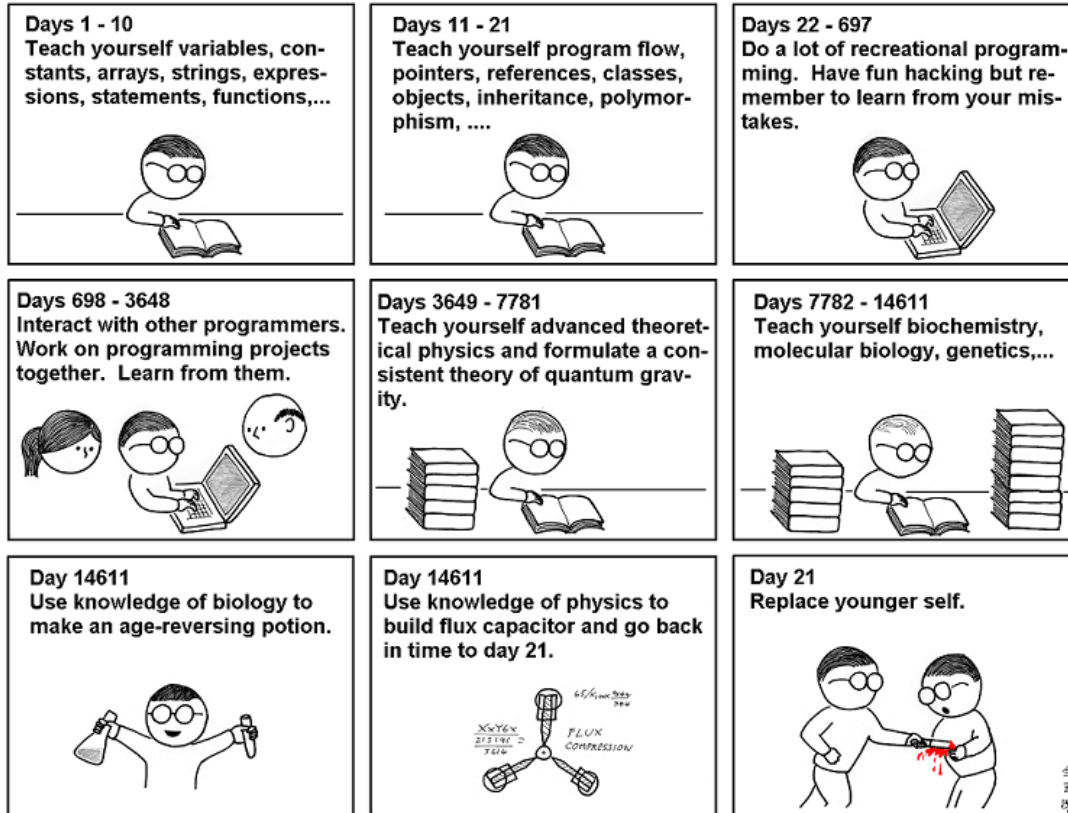


Was ist C++?

- ◆ C++ ist kein(e)
 - C mit Klassen (auch wenn es das von 1797 bis 1985 war)
 - Low Level Java
 - rein prozedurale Sprache
 - rein objektorientierte Sprache
- ◆ C++
 - hat Kompatibilität zu C
 - kann Low Level Operationen implementieren
 - kann prozedural/objektorientiert/funktional Verwendet werden
 - Ist generisch

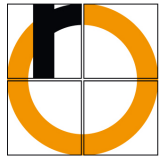


Was ist C++?



As far as I know, this is the easiest way to "Teach Yourself C++ in 21 Days".





Was ist C++?



```
const uint32_t x_mod = x % world_width;
const uint32_t y_mod = y % world_height;
uint32_t x_mod_minus_1 = x_mod == 0 ? world_width - 1 : x_mod - 1;
uint32_t x_mod_plus_1 = x_mod + 1 == world_width ? 0 : x_mod + 1;
uint32_t y_mod_minus_1 = y_mod == 0 ? world_height - 1 : y_mod - 1;
uint32_t y_mod_plus_1 = y_mod + 1 == world_height ? 0 : y_mod + 1;
w = w_mask == -1 ? x_mod_minus_1 : 0;
e = e_mask == -1 ? x_mod_plus_1 : 0;
n = n_mask == -1 ? y_mod_minus_1 : 0;
s = s_mask == -1 ? y_mod_plus_1 : 0;

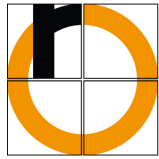
// Calculate offsets within map memory.
w = y * width + w;
e = y * width + e;
n = n * width + x;
s = s * width + x;

// Extract neighbours heights. Apply validity filtering: 0 is invalid.
w_crust = map[w] * (w_mask & (map[w] < map[index]));
e_crust = map[e] * (e_mask & (map[e] < map[index]));
n_crust = map[n] * (n_mask & (map[n] < map[index]));
s_crust = map[s] * (s_mask & (map[s] < map[index]));
```



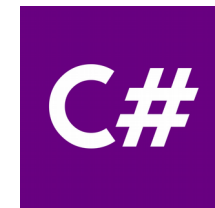
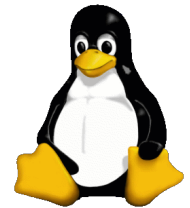
Was ist C++?

- ◆ Warum C++?
 - Komplizierte Syntax
 - „alte“ Sprache
 - Seiteneffekte
 - Memory-Leaks / kein Garbage Collection
 - Unlesbare Compiler-Fehler
 - Komplizierte Build-Tools
 - (Fast) kein Package-Management



Was ist C++?

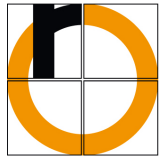
Wer verwendet die Sprache überhaupt noch?





Was ist C++?

- ◆ Warum ist C++ noch so weit verbreitet?
 - Zero-Overhead abstraction
 - Plattform unabhängig
 - Performant
 - „What you don't use, you don't pay for“
 - Möglichkeit für eigene Low-Level funktionen
 - Freie/Kostenlose Compiler
 - Moderner Standard



Was ist C++?

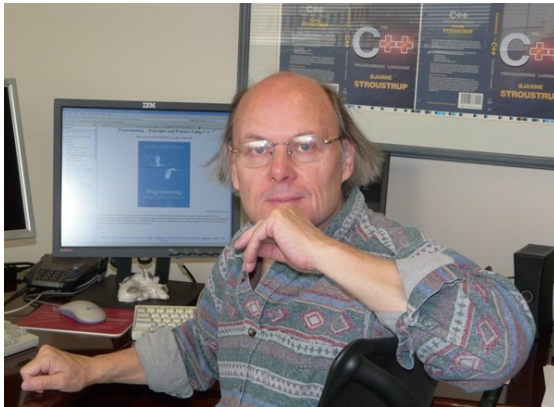
- ◆ Kurze Geschichte von C++
 - 1979 erste „C with Classes“ Implementierung von Bjarne Stroustrup an den AT&T Bell Laboratories
 - 1985 erstes mal als C++ erwähnt (Erweitert mit new/delete/const/string)
 - 1987 C++ unterstützung in GCC
 - 1990 Einführung von Templates, Exceptions
 - 1991 ISO C++ Komitee
 - 1992 Implementierung der Standard Template Library (STL)
 - 1998 C++98 Standard
 - Dynamische Casts, bool, containers, algorithm, Funktionsobjekte
 - 2003 C++03 (kleine Fehlerbehebungen)
- ◆ Dann kam JAVA / Python / Web 2.0



Was ist C++?

- ◆ Die C++ Renaissance
 - Der C++11 Standard
 - „Feels like a new Language“ (Stroustrup)
 - Auto, move-operationen, constexpr, range for loops, lambda, smart pointers, threads, mutex, futures, uvm.
 - Ende der „Free Lunch“ Epoche
 - C++14
 - Fixes für C++ 11
 - C++17
 - Filesystem, variant, any, structured bindings, parallel Algorithms

Was ist C++?



Bjarne Stroustrup

- „Erfinder“ von C++
- Gründer und Leiter des ISO Komitees bis 2002
- Erhielt vor September 2017 die Farady Mediallie



Herb Sutter

- Leiter des ISO Komitees seit 2002
- Autor von C++ Coding Standards

Was ist C++?



Alexander Stepanow

- „Schöpfer“ der STL



Scott Meyers

- Autor der Effective C++ Reihe

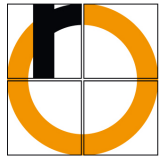


Chandler Carruth

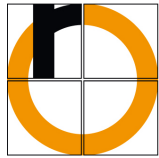
- Clang und LLVM Entwickler



Sean Parent



Ziele der Vorlesung



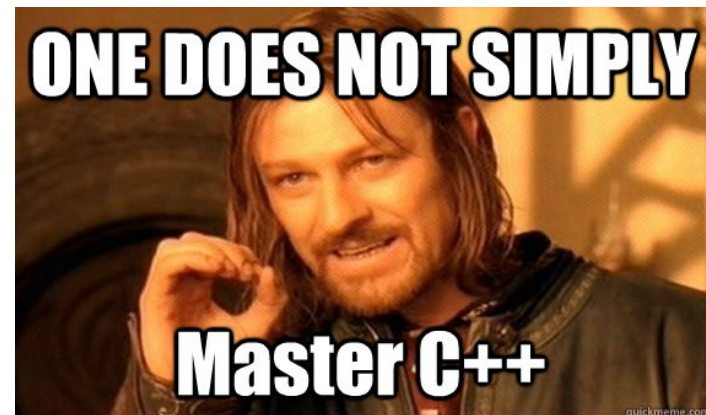
Ziele der Vorlesung

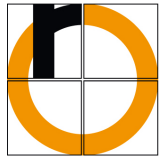
- ◆ C++ Best Practice / C++ Core Guidelines
- ◆ Sauberer, lesbarer C++ Code
- ◆ Verständnis für Optimierung
- ◆ Entlarven falscher „Wahrheiten“
- ◆ Verstehen von Assembler



Ziele der Vorlesung

- ◆ Core Guildelines
 - github.com/isocpp/CppCoreGuidelines/blob/master/CppCoreGuidelines.md
 - „Allgemeine“ Regeln für guten C++ Code
- ◆ Best Practice
 - Teilweise aus eigener Erfahrung
 - Teilweise aus Vorträgen/Bücher etc.
 - Diskussion Erwünscht!!!!





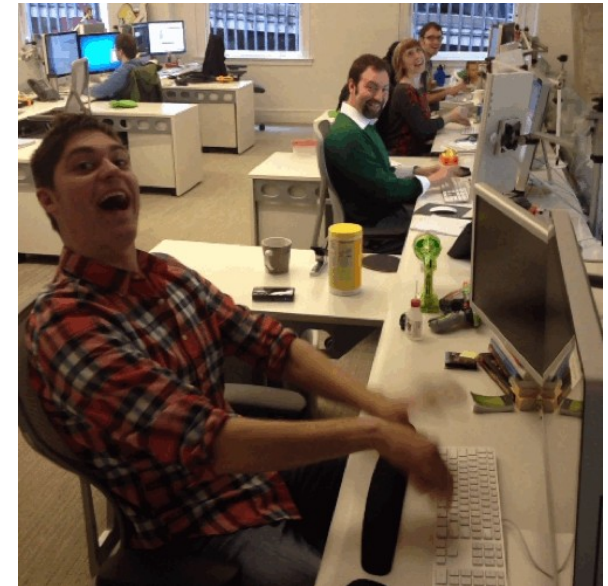
Ziele der Vorlesung

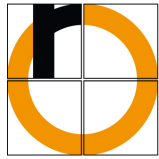
- ◆ Sauberer, lesbarer C++ Code
 - Verwenden von c++11/14 neuerunge
 - Keep it Small and Simple
 - Tipps und Tricks für „Legacy Code“



Ziele der Vorlesung

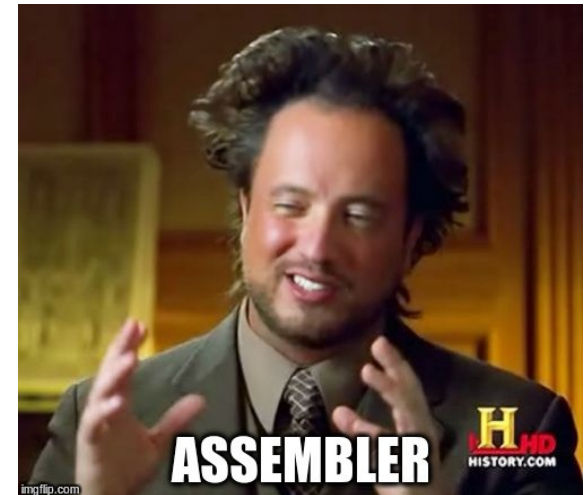
- ◆ Verständnis für Optimierung
 - Wir programmieren für den Compiler!!
 - Was bringen die verschiedenen Optimierungen
 - Versuche nicht schlauer zu sein als der Compiler
 - <http://www.quick-bench.com/>

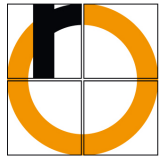




Ziele der Vorlesung

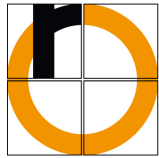
- ◆ Verstehen von Assembler
 - Nicht selber schreiben, aber verstehen was passiert
 - godbolt.org



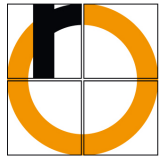


Ziele der Vorlesung

- ◆ Entlarven falscher „Wahrheiten“
 - Gleitkommazahl langsamer als Ganzzahl ?? → nicht wahr
 - Division langsamer als Multiplikation ?? → nicht wenn optimiert

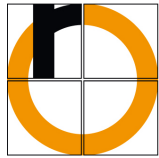


Der „moderne“ Standard



Der moderne Standard

- ◆ Modern???
- Wird verwendet für die Standards ab C++ 11.
- Beschreibt erstmals einen Multi-Threaded Maschine
- Besser lesbarer Code
- Vorlesung behandelt auch ältere Standarts



Der moderne Standard

C / C++ 98

```
int i = 0;
for(int j = 0; j <= 20; ++j)
    i += j;
```

C++ 11, Algorithmen, geht ohne „auto“ auch schon in C++98

```
std::vector<int> x(10);
std::iota(std::begin(x), std::end(x), 0);
auto j = std::accumulate(std::begin(x), std::end(x), 0);
```

C++ 11, `for_each` Schleife

```
std::vector<int> x(10);
std::iota(std::begin(x), std::end(x), 0);
auto i {0};
for(auto j : x)
    i +=j;
```




Weitere Quellen

- ◆ The Design and Evolution of C++ (Stroustrup, Verlag: Pearson Education)
- ◆ github.com/isocpp/CppCoreGuidelines/blob/master/CppCoreGuidelines.md
- ◆ isocpp.org
- ◆ youtube.com/user/CppCon
- ◆ C++ Weekly on Youtube (fortgeschritten)