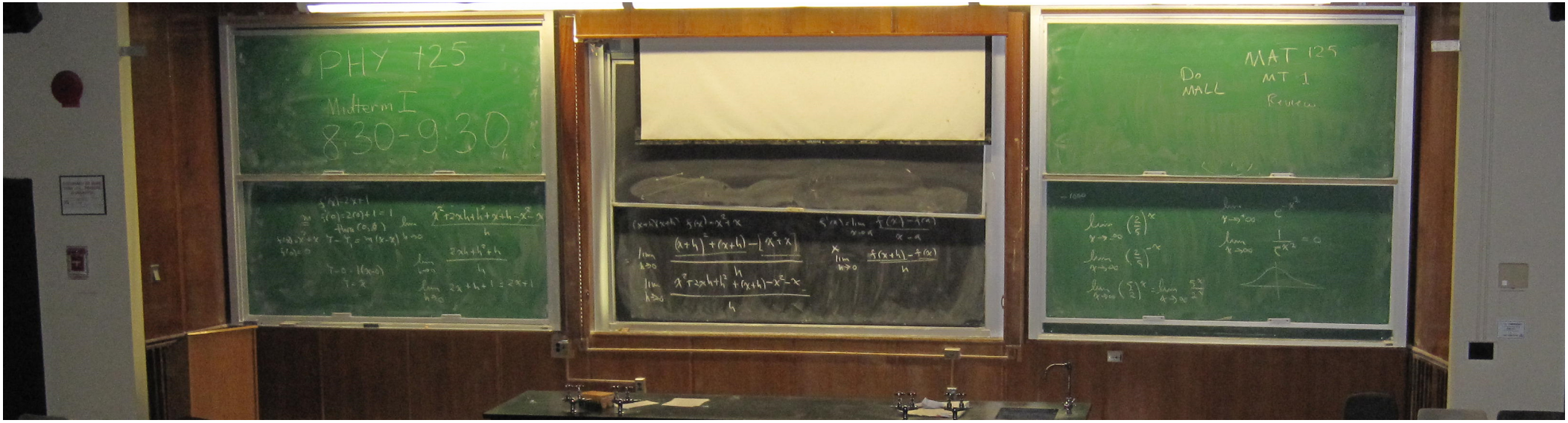
zühlke
empowering ideas



Learning and Teaching Modern C++: Challenges and Resources

@arne_mertz

“Modern C++”?

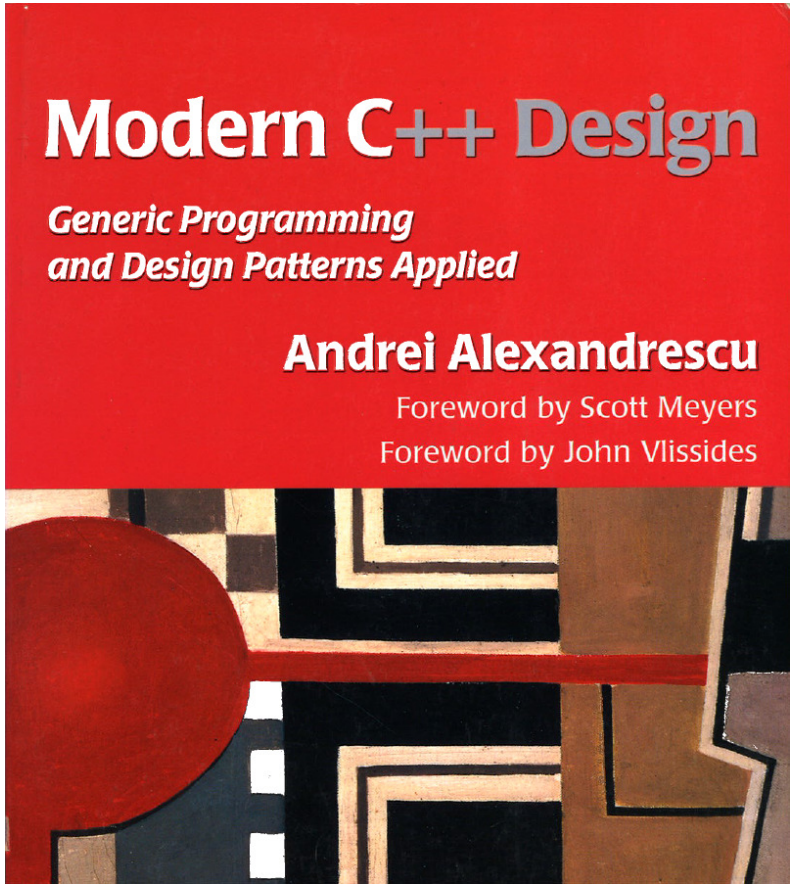
Emojis by emojione.com

“Modern C++ Design”

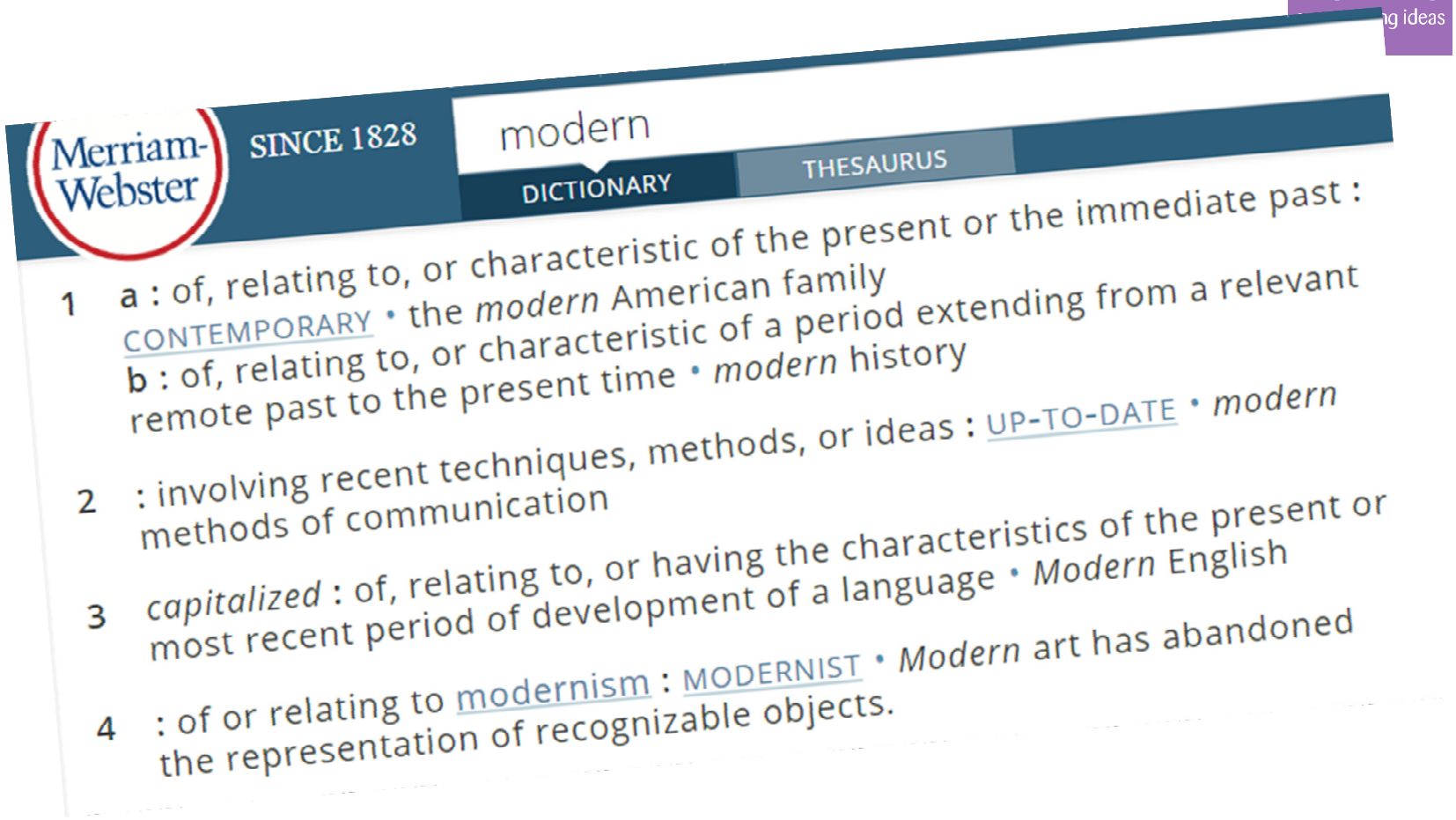
Andrei Alexandrescu

- From 2001 – C++ has evolved since then
- Very focused on templates
  - Policy based class design
  - Compile time assertions
  - Allocators
  - Type lists

- Still a good read
- Influenced our thinking about C++
- But not “Modern C++” anymore



Modern



Merriam-Webster SINCE 1828

modern

DICTIONARY THESAURUS

1 **a** : of, relating to, or characteristic of the present or the immediate past : CONTEMPORARY • the *modern* American family
b : of, relating to, or characteristic of a period extending from a relevant remote past to the present time • *modern* history

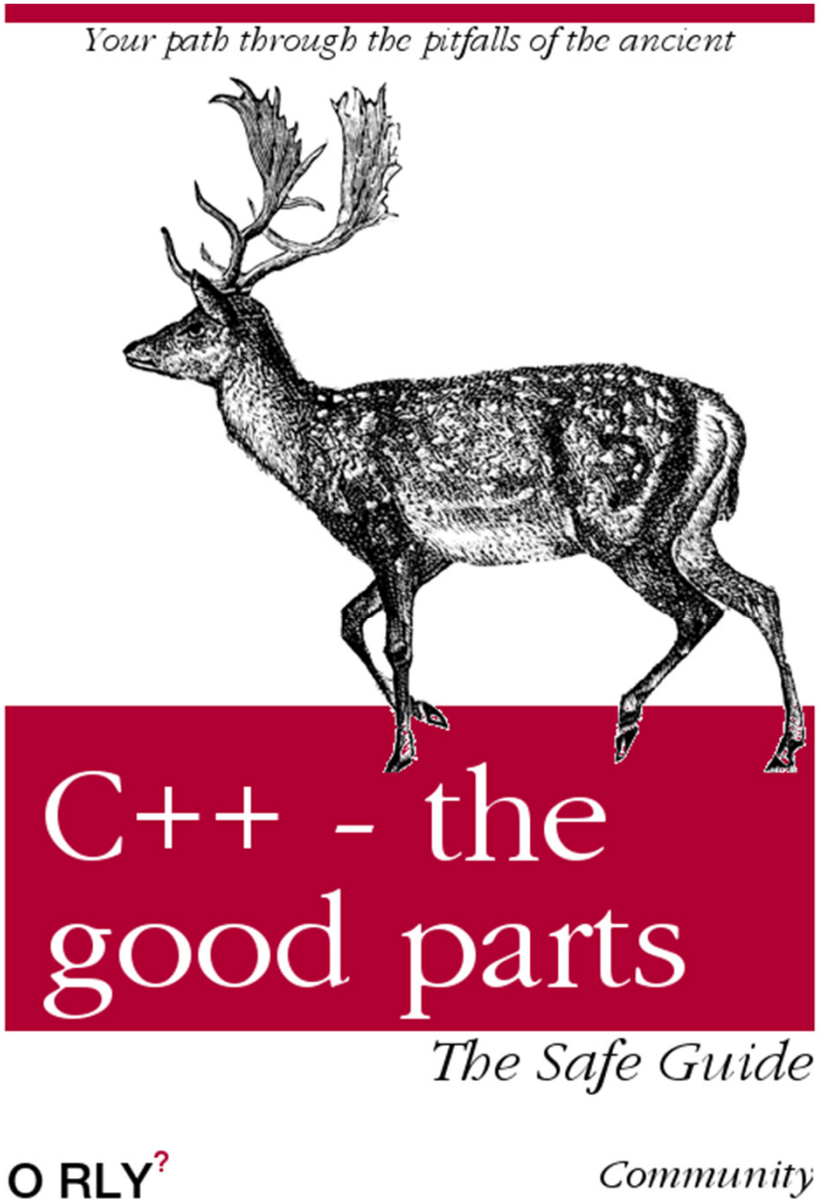
2 : involving recent techniques, methods, or ideas : UP-TO-DATE • *modern* methods of communication

3 *capitalized* : of, relating to, or having the characteristics of the present or most recent period of development of a language • *Modern* English

4 : of or relating to modernism : MODERNIST • *Modern* art has abandoned the representation of recognizable objects.

"Within C++ is a smaller, simpler, safer language struggling to get out."

Bjarne Stroustrup


Your path through the pitfalls of the ancient
C++ - the good parts
The Safe Guide
O RLY?
Community

The Good Parts

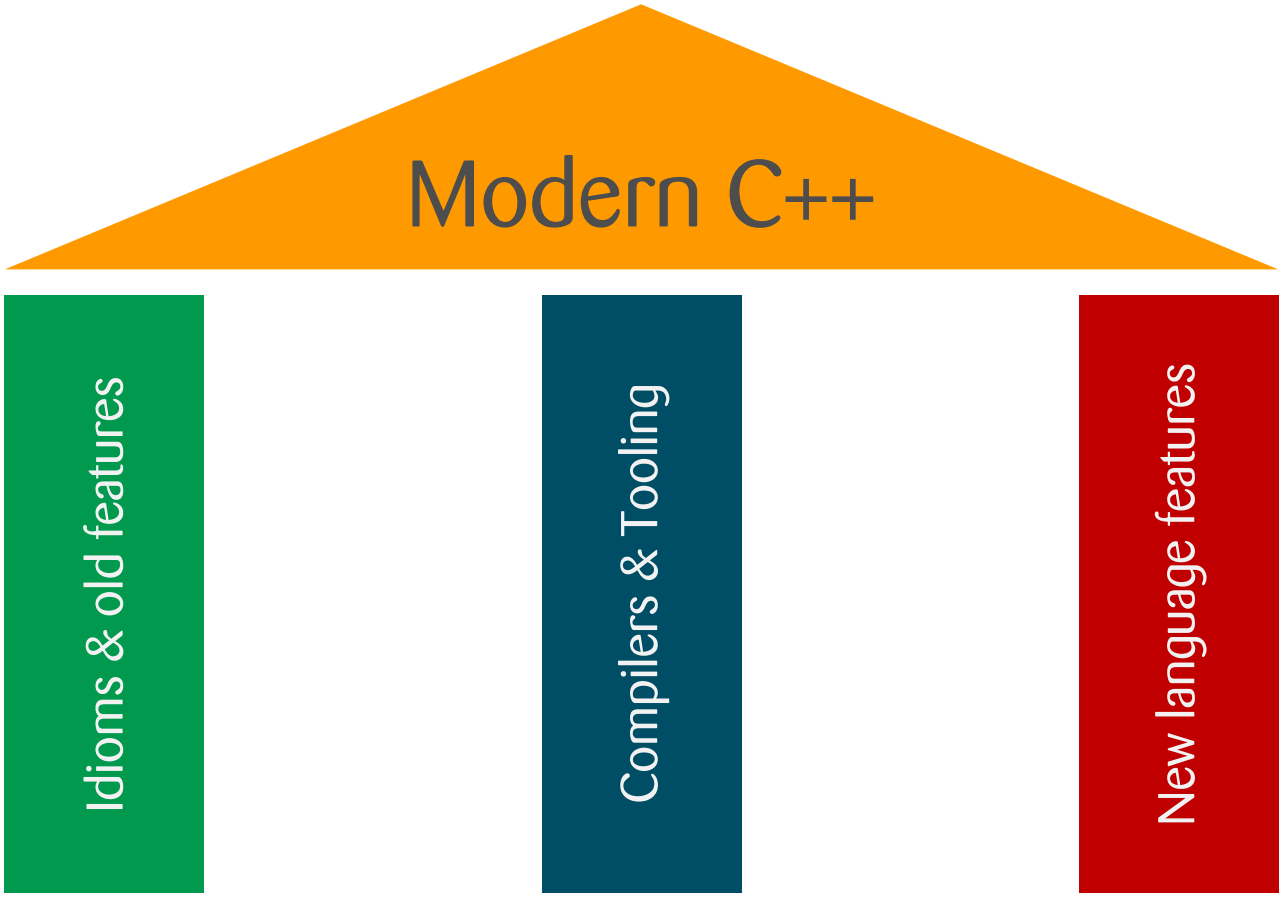
"When I was a young journeyman programmer, I would learn about every feature of the languages I was using, and I would attempt to use all of those features when I wrote. I suppose it was a way of showing off, and I suppose it worked because I was the guy you went to if you wanted to know how to use a particular feature.

Eventually I figured out that some of those features were more trouble than they were worth.[…] Most programming languages contain good and bad parts. I discovered that I could be a better programmer by using only the good parts and avoiding the bad parts. After all, how can you build something good out of bad parts?
It is rarely possible for standard committees to remove imperfections from a language because doing so would cause the breakage of all the bad programs that depend on those bad parts. […]

But *you* have the power to define your own subset. You can write better programs by relying exclusively on the good parts."

Douglas Crockford – "JavaScript: The Good Parts"

“Modern C++” != “New Standards”



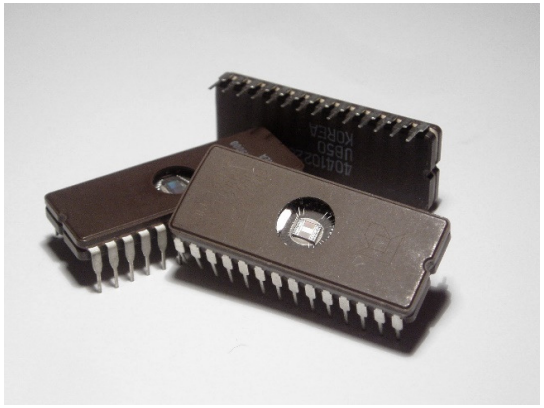
Modern C++ is…

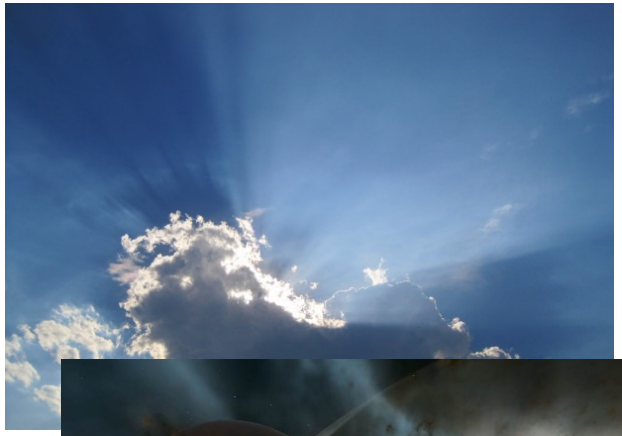
… a moving target

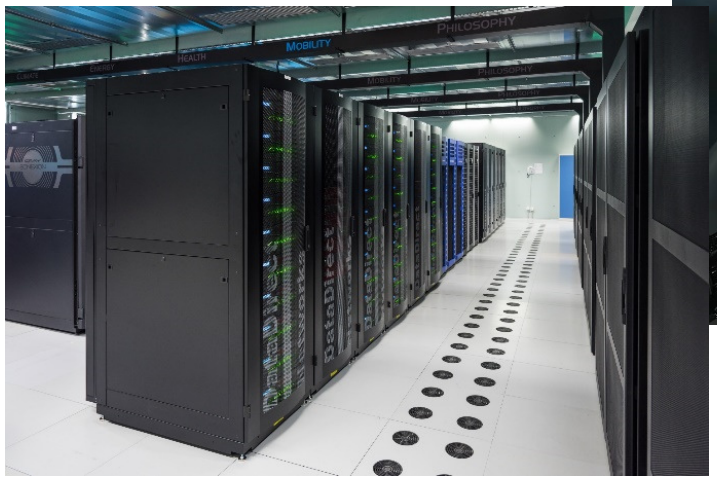
- Continuosly new features
  - New standards currently every ~3 years
  - Plus TS

- But besides that, our *understanding* of the language continually changes

Many moving targets…

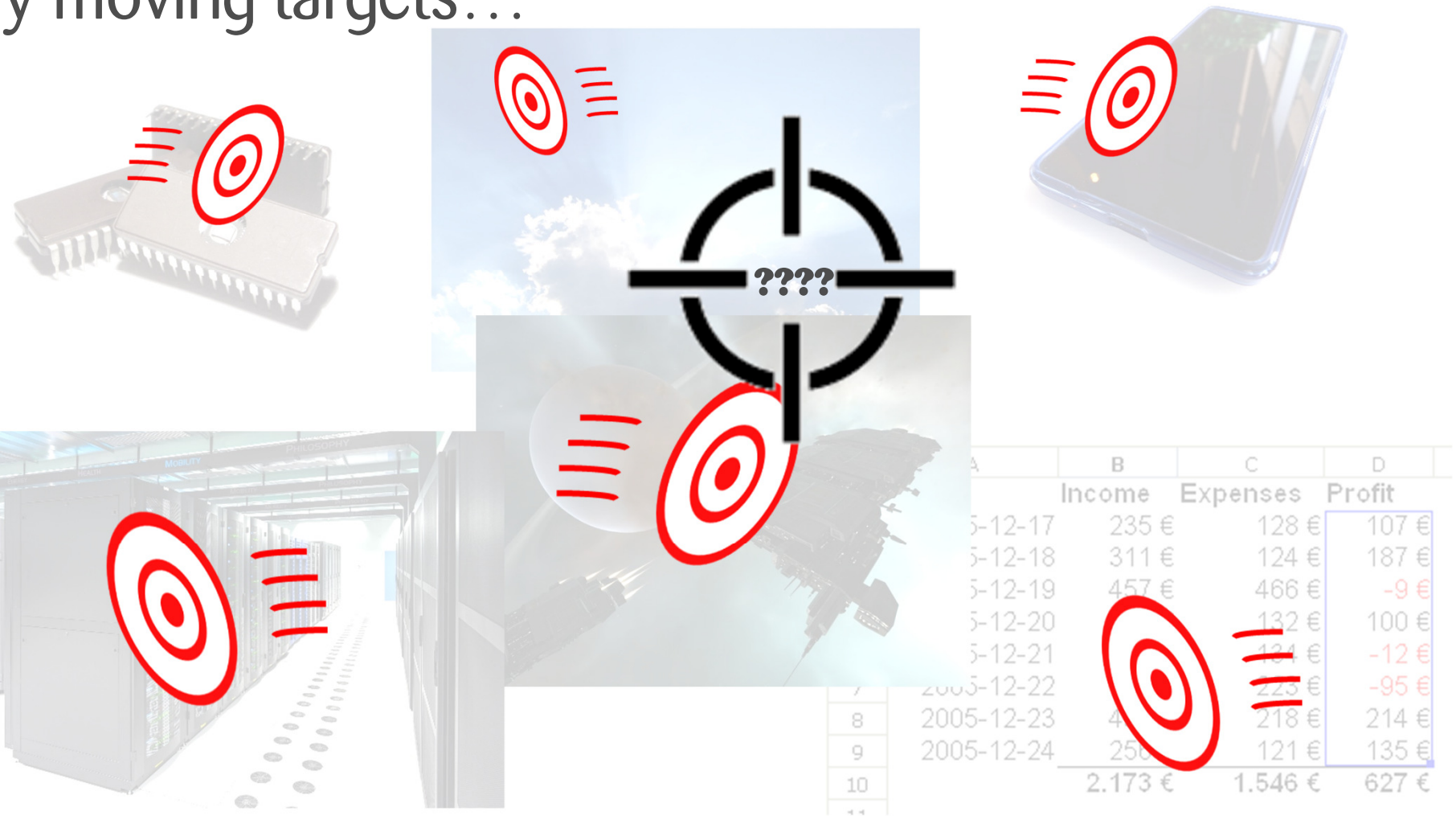






	A	B	C	D
		Income	Expenses	Profit
	5-12-17	235 €	128 €	107 €
	5-12-18	311 €	124 €	187 €
	5-12-19	457 €	466 €	-9 €
	5-12-20	232 €	132 €	100 €
	5-12-21	122 €	134 €	-12 €
7	2005-12-22	128 €	223 €	-95 €
8	2005-12-23	432 €	218 €	214 €
9	2005-12-24	256 €	121 €	135 €
10		**2.173 €**	**1.546 €**	**627 €**

Many moving targets…



Many moving targets…

… e.g. sales pitch for RAII

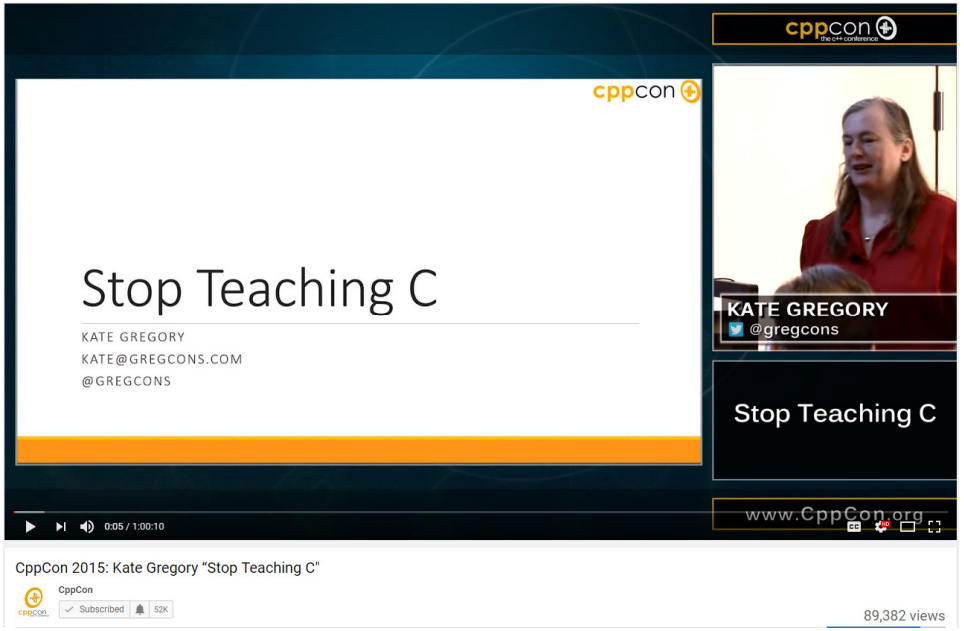
- Usual examples:
  - unique_ptr (memory)
  - lock_guard (mutexes)
  - fstream (files)
- Usual argument: Exception safety

Additional challenge

… for teachers

- Modern C++ != layered evolution
- But what we learned first sticks best

→Teach straight to the point instead of retelling our personal history

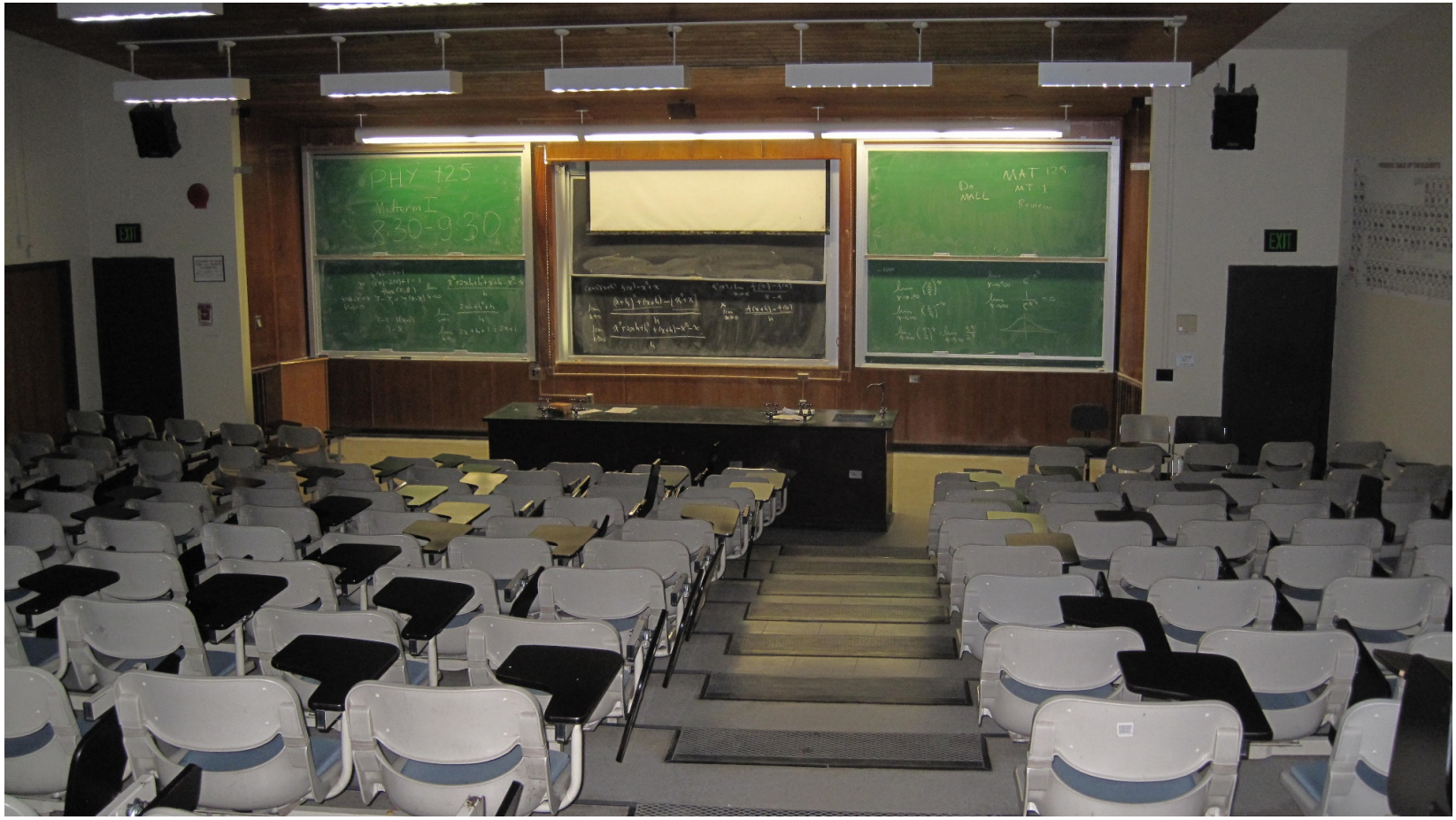


Resources

Emojis by emojione.com

Class room…

… and teachers/professors



Class room…

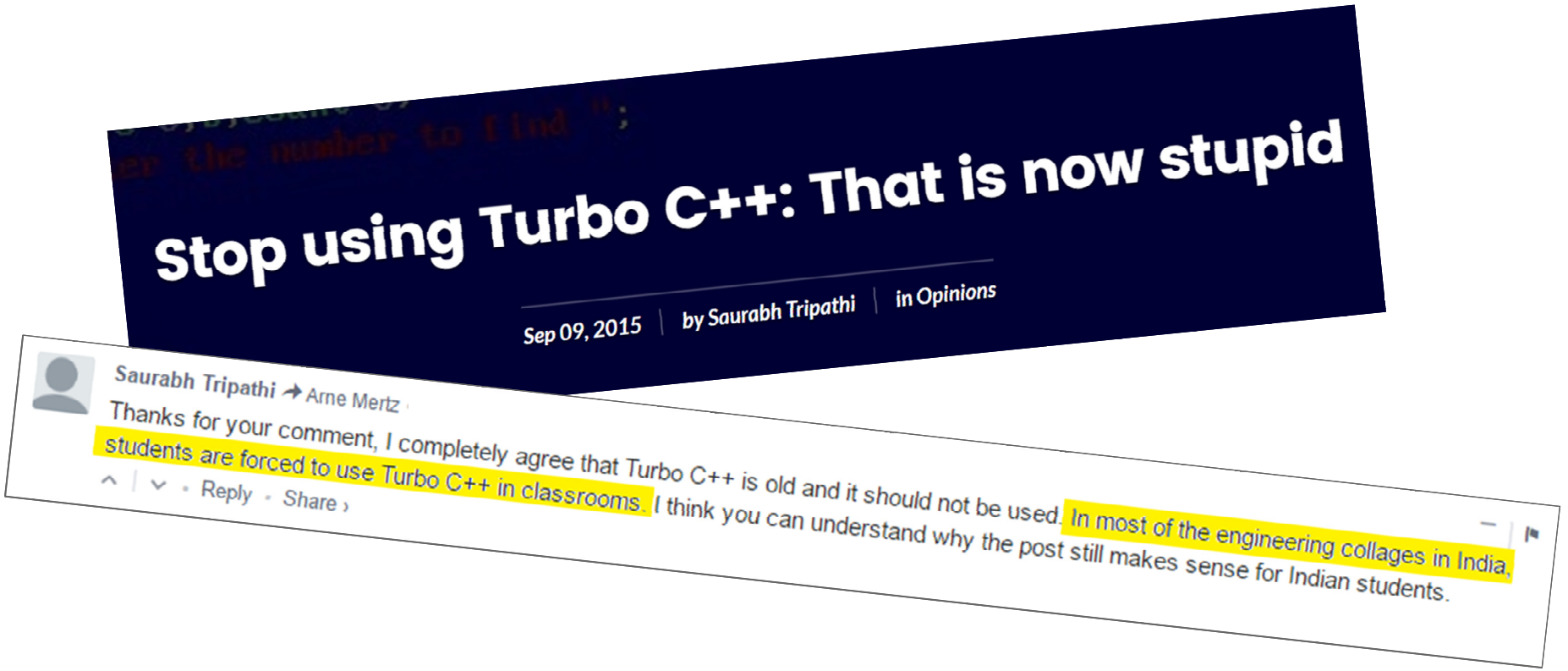
… and teachers/professors

- Some teachers are enthusiastic and love to teach

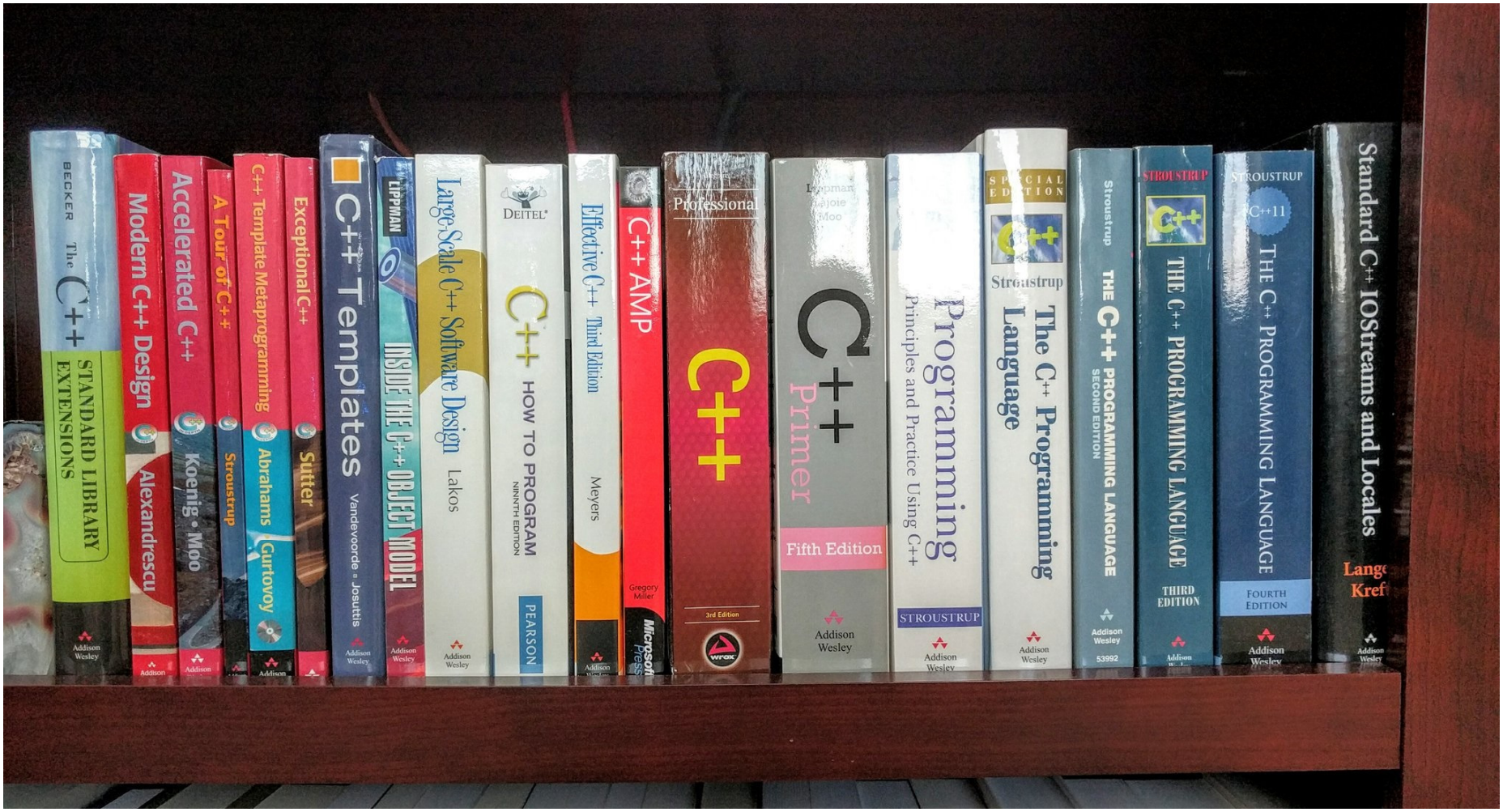
- Others don't but *have* to
  - Prepare class once
  - Repeat for N years

  → fresh legacy C++ students

Class room…



Books



Books

- Pros:
  - Consistency, didactic line of thought
  - Usually reviewed (but not always)
- Cons:
  - Last longer than the validity of the content
  - Are often promoted to make money – not because they are good
  - Errors are hard to fix
  - Usually not free of charge (this *does* matter!)

Books

C++ <iostream.h> Error

0

I am at the absolutely newest level of new when it comes to C++. It may seem like a noob mistake, but I think I'm missing something with my first program, "Hello World!".

I'm running from Ubuntu (not sure if this is any different from working with Windows), and I'm using a book called *Teach Yourself C++ in 21 Days*.

The code I'm resembling looks exactly like this:

```
#include <iostream.h>
int main()
{
cout <<"Hello World!\n";
    return 0;
}
```

I have this exactly in my text editor, but I keep getting greeted by the same error whenever I try to compile it!

> first.cpp:2:22: fatal error: iostream.h: No such file or directory compilation terminated.

I'm pretty distressed as this is literally the first step in my coding career! I'm not sure if ubuntu needs to be treated differently than Windows (which is what the book is using as reference).

Help!

asked Jul 13 '13 at 20:16

Blogs & Tutorials

Meeting C++ Blogroll 166

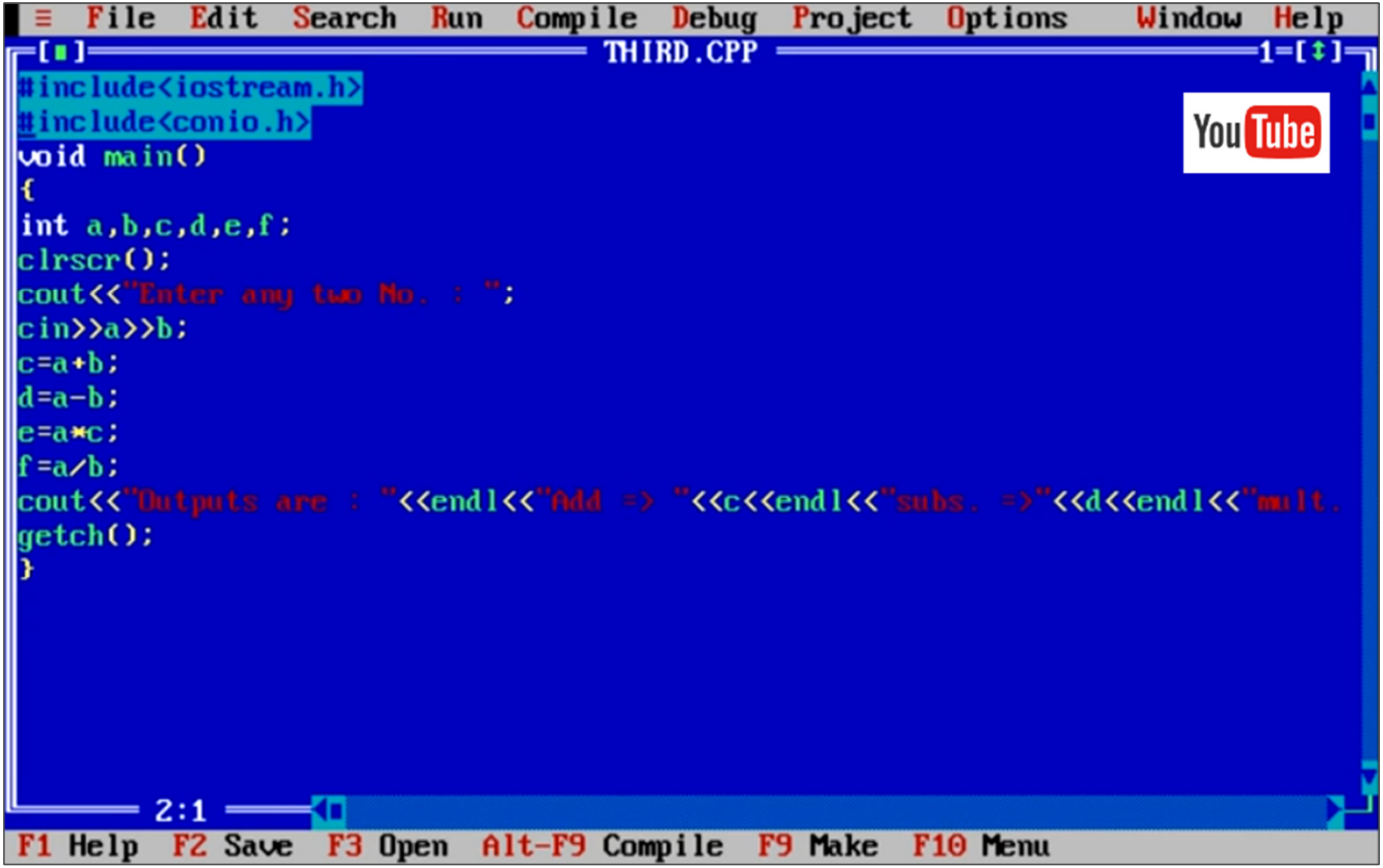
published at 09.11.2018 10:02

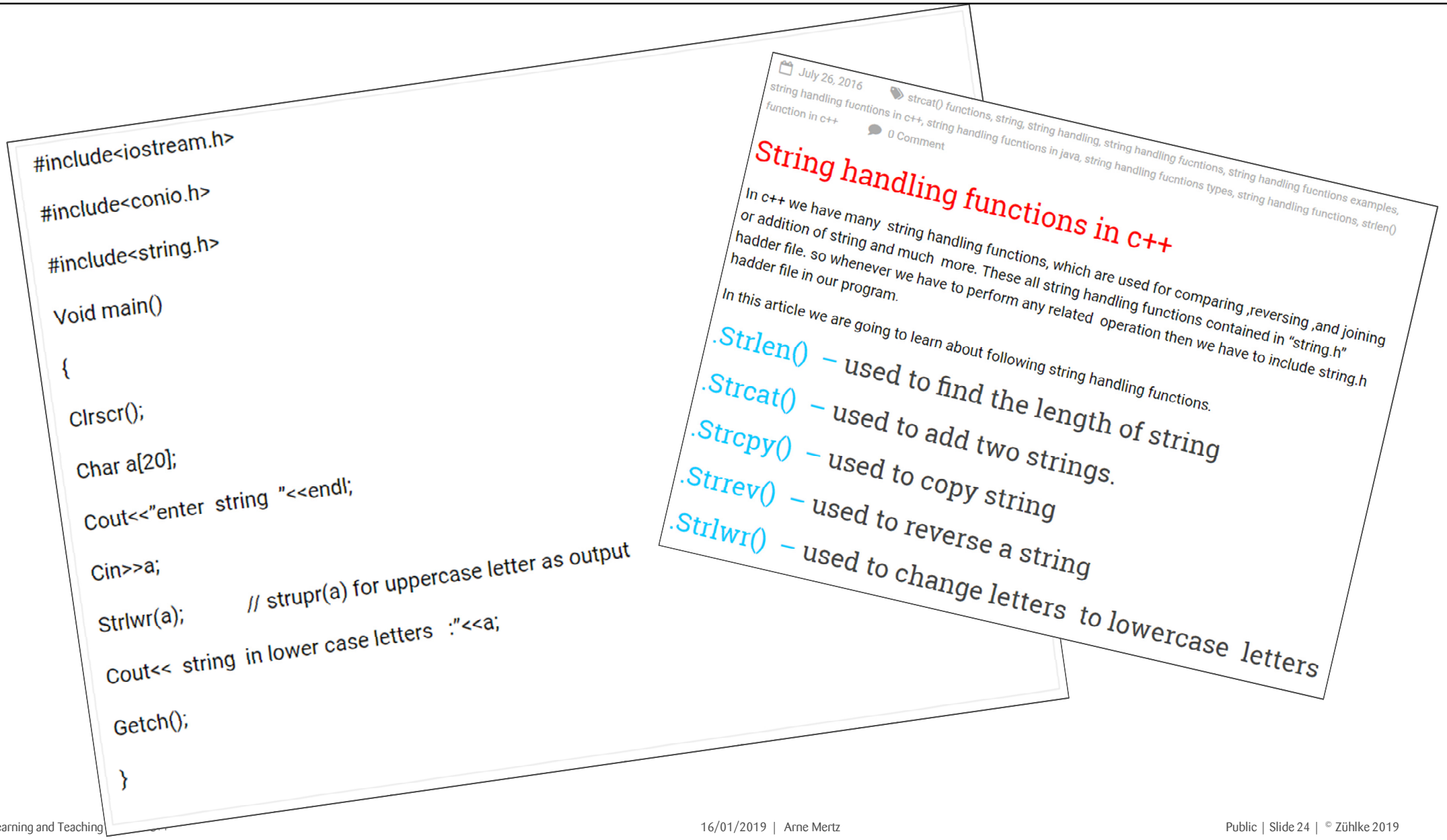
Blogroll No. 166 - 9. November

- Arne Mertz - C++ Quizzes and Trick Questions – Do we Have to Know the Answers?
- Arthur O'Dwyer - An argument _pro_ liberal use of `[[nodiscard]]`
- Arthur O'Dwyer - WG21: Avoid creating ADL situations on functions with semi-common names
- Bartek's Coding Blog - Parallel Algorithms Chapter in C++17 In Detail
- CLion - CLion 2018.3 EAP: clangd-based navigation and search, CPU Profiler and remote mode improvements
- Conan Releases - 1.9.1 (08-Nov-2018)
- CppCast - Learning C++ with Devon Labrie
- CppCon - Call for Proposals for CppCon 2019 Classes
- Dirk Eddelbuettel - Rcpp 1.0.0: The Tenth Birthday Release
- Dirk Eddelbuettel - Happy 10th Bday, Rcpp – and welcome release 1.0 !!
- Embedded in Academia - Learning When Values are Changed by Implicit Integer Casts
- Embedded in Academia - Synthesizing Constants
- Fluent C++ - How to Retrieve the Firsts from a Collection of Pairs
- Fluent C++ - How to Transfer unique_ptrs From a Set to Another Set
- Godot Engine - Dev snapshot: Godot 3.1 alpha 2
- Godot Engine - Meet the community at FOSDEM and GodotCon 2019
- ICS - Qt World Summit Boston 2018 Was a Huge Success
- KDAB - Qt World Summit 2018 Boston
- Meeting C++ Blogroll - Meeting C++ Blogroll 165
- Meeting C++ Jobs - Great Team Player, passionate about algorithms and fluent in C++
- Meeting C++ News - C++ User Group Meetings in November 2018
- Modernes C++ - C++ Core Guidelines: Pass Function Objects as Operations
- ninepoints - Thoughts on the Cpp Graphics Proposal
- NVIDIA DevBlog - How to Speed Up Deep Learning Inference Using TensorRT
- NVIDIA DevBlog - Accelerated Ray Tracing in One Weekend in CUDA
- Paul Fultz II - Using STL algorithms with cppcheck
- Qt Blog - What's New in Qt Help?
- Rainer Grimm - C++ Core Guidelines: Übergabe von Funktionsobjekten als Operationen
- SObjectizer - SObjectizer 5.5.23 and so_5_extra 1.2.0 Released!
- Stellar Group - Compiling and Running BlazeTest
- The Old New Thing - Using linker segments and __declspec(allocate(…)) to arrange data in a specific order
- ThePHD - San Diego 2018 - The Plan™ (Ready to Roll)
- ThePHD - 📝👍 - How to Write a Good Proposal to C++
- vector of bool - Levelling the try/catch Mountain
- vector of bool - The Dumbest Allocator of All Time
- Visual C++ Team Blog - Use the official range-v3 with MSVC 2017 version 15.9
- Visual C++ Team Blog - Exploring Clang Tooling Part 3: Rewriting Code with clang-tidy

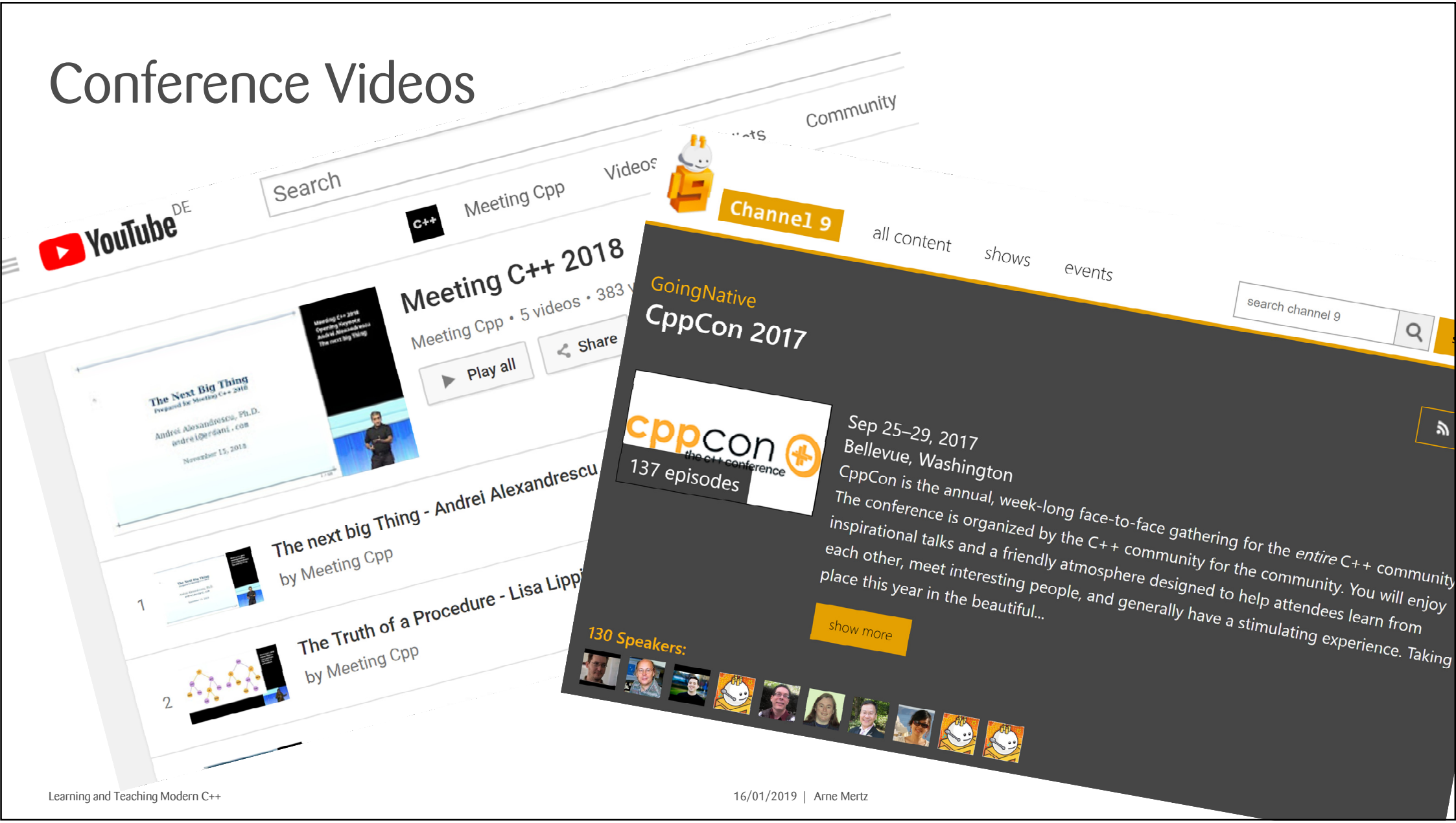
Blogs & Tutorials

- Pros:
  - Usually free of charge
  - Authors can always fix things or take them offline
- Cons:
  - But we usually don't
  - Hardly any review
  - Less coherent bits and pieces


File Edit Search Run Compile Debug Project Options Window Help
THIRD.CPP
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b,c,d,e,f;
clrscr();
cout<<"Enter any two No. : ";
cin>>a>>b;
c=a+b;
d=a-b;
e=a*c;
f=a/b;
cout<<"Outputs are : "<<endl<<"Add => "<<c<<endl<<"subs. =>"<<d<<endl<<"mult.
getch();
}
2:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


#include<iostream.h>
#include<conio.h>
#include<string.h>
Void main()
{
Clrscr();
Char a[20];
Cout<<"enter string "<<endl;
Cin>>a;
Strlwr(a); // strupr(a) for uppercase letter as output
Cout<< string in lower case letters :"<<a;
Getch();
}
July 26, 2016
strcat() functions, string, string handling, string handling fucntions, string handling fucntions examples, string handling fucntions in c++, string handling fucntions in java, string handling fucntions types, string handling functions, strlen() function in c++
0 Comment
String handling functions in c++
In c++ we have many string handling functions, which are used for comparing ,reversing ,and joining or addition of string and much more. These all string handling functions contained in "string.h" hadder file. so whenever we have to perform any related operation then we have to include string.h hadder file in our program.
In this article we are going to learn about following string handling functions.
.Strlen() – used to find the length of string
.Strcat() – used to add two strings.
.Strcpy() – used to copy string
.Strrev() – used to reverse a string
.Strlwr() – used to change letters to lowercase letters

Conference Videos



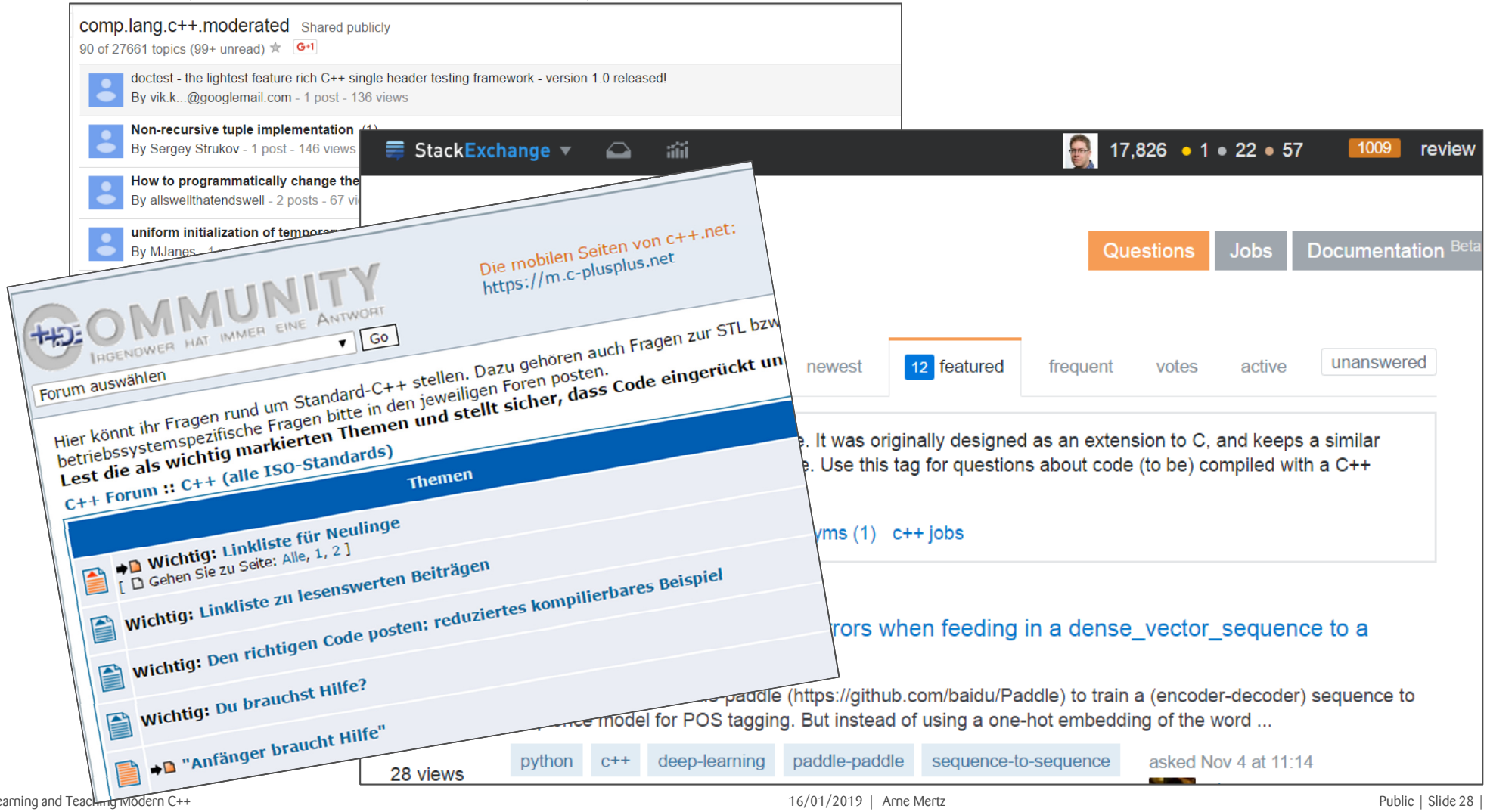
Colleagues & Reading Code

Colleagues & Reading Code

- Pros:
  - Colleagues know the quirks of our niche
- Cons:
  - Need to have someone who knows the language well

- Pros:
  - Large variety of open code bases (GitHub etc.)
- Cons:
  - Code alone usually is not suitable as a learning resource

Forums, Stackoverflow, …



Forums, Stackoverflow, …

- Pros:
  - Multiple views, i.e. a community
- Cons:
  - Very specific detailed topics
  - Too much noise
  - Trolls

What do we need for *better* learning resources?

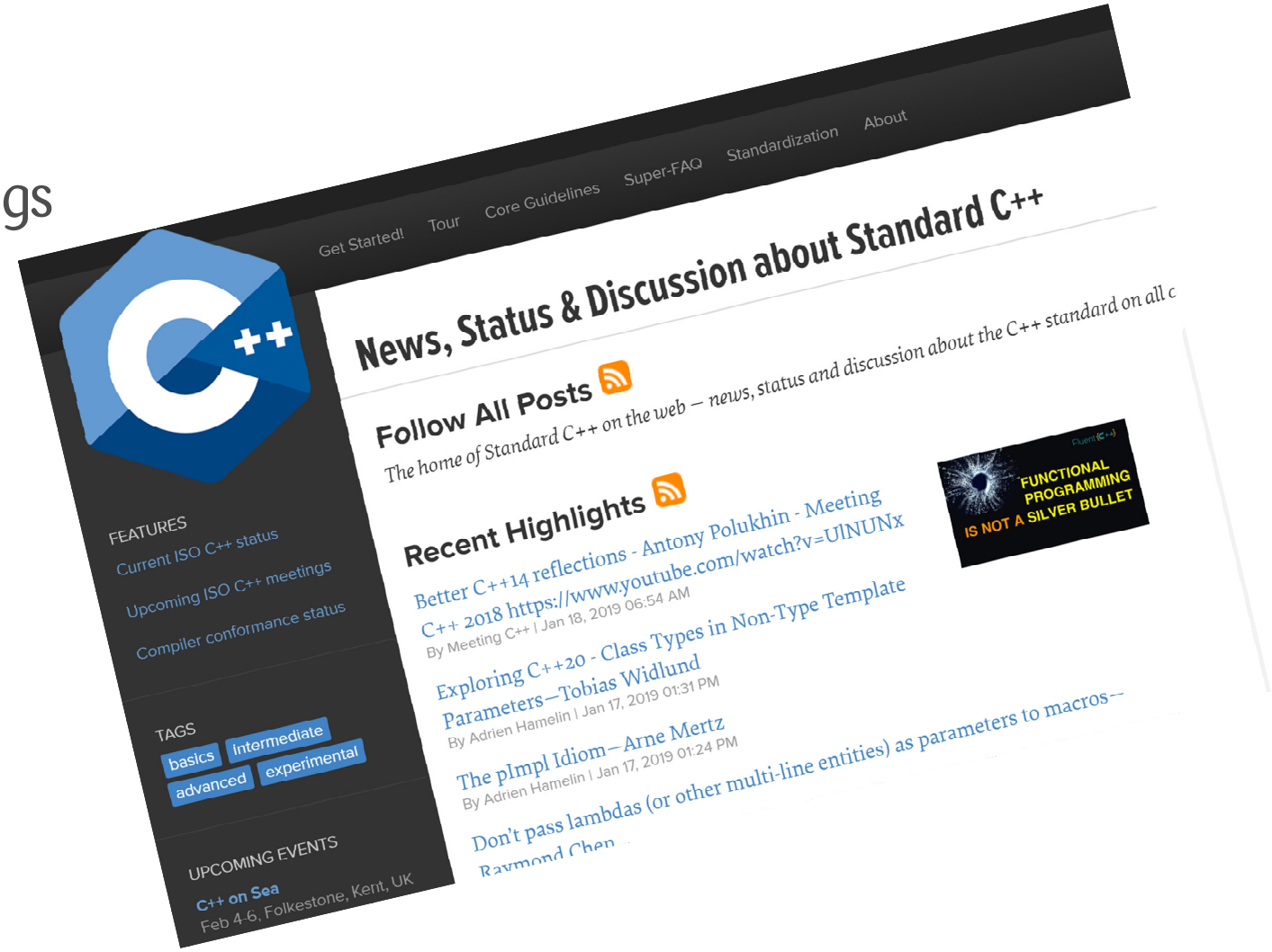
And *teaching* resources, of course.

- Authority – to attract more students
  - When someone asks for a resource, there should be one or two answers, not tens or hundreds
- Flexibility or even volatility
  - To avoid outdated resources
  - To adapt resources to different needs (niches)
- A community
  - Healthy discussion about modern C++
  - More than a single opinion

What we have

https://isocpp.org/

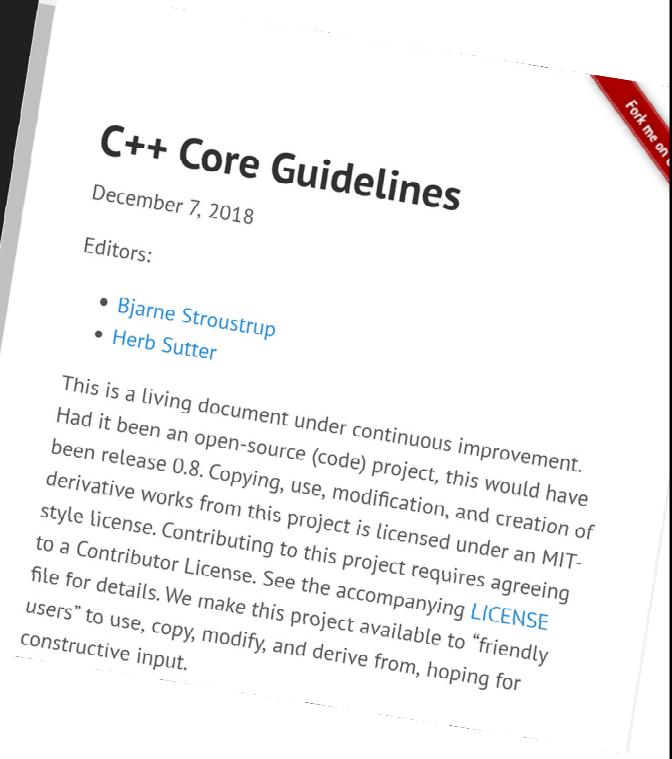
- Hub for C++-related things
- Forum
- Blog
  - Blogs & Books
  - Videos
  - Events & Training
- Standardization Info
  - Status, Notes,…
- C++-FAQ



What we have

http://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines

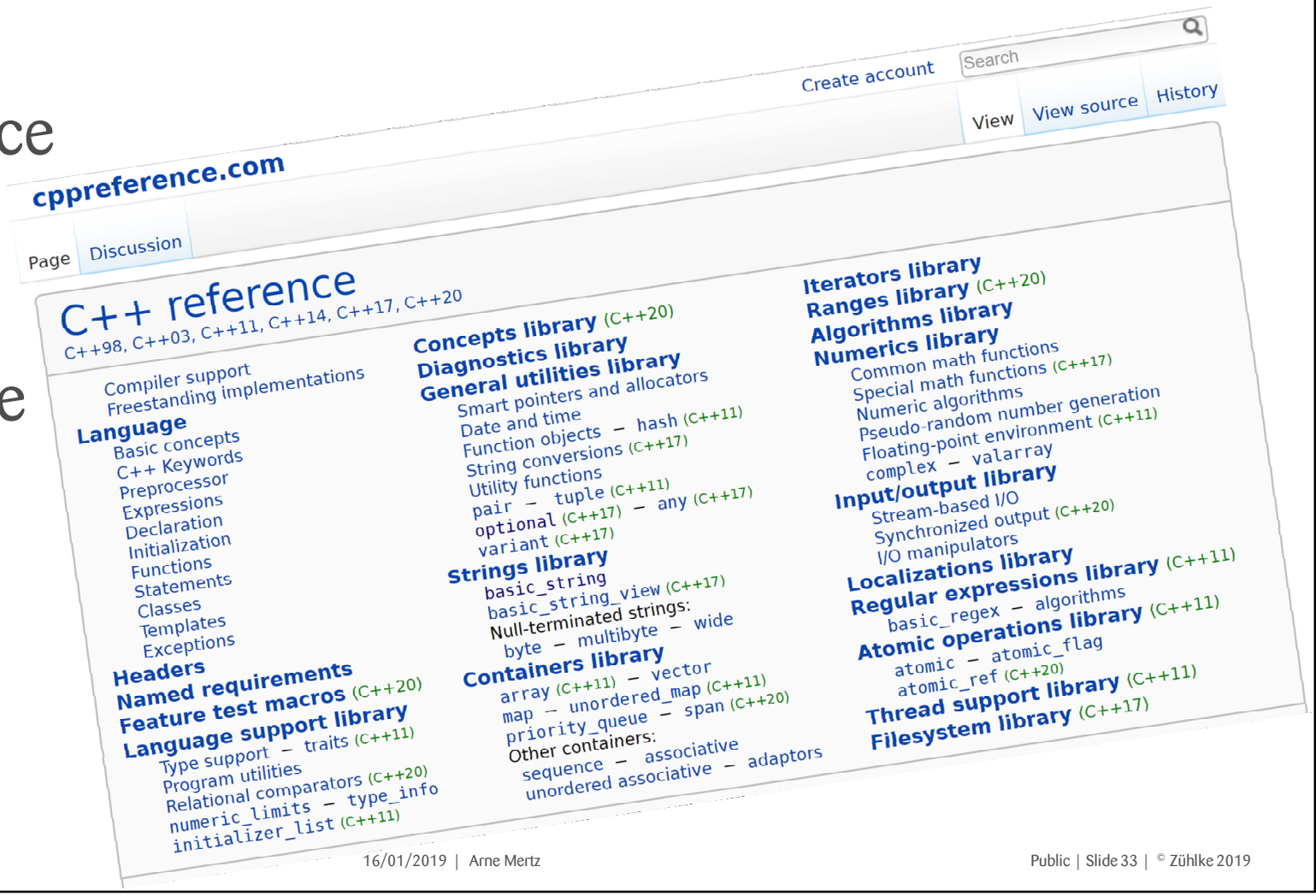
- Best practice guidelines
  - With rationale
  - alternatives for unsafe and outdated idioms
- Static analysis rules
  - Static analyzers can find a lot of code smells
- Guideline Support Library



What we have

cppreference.com

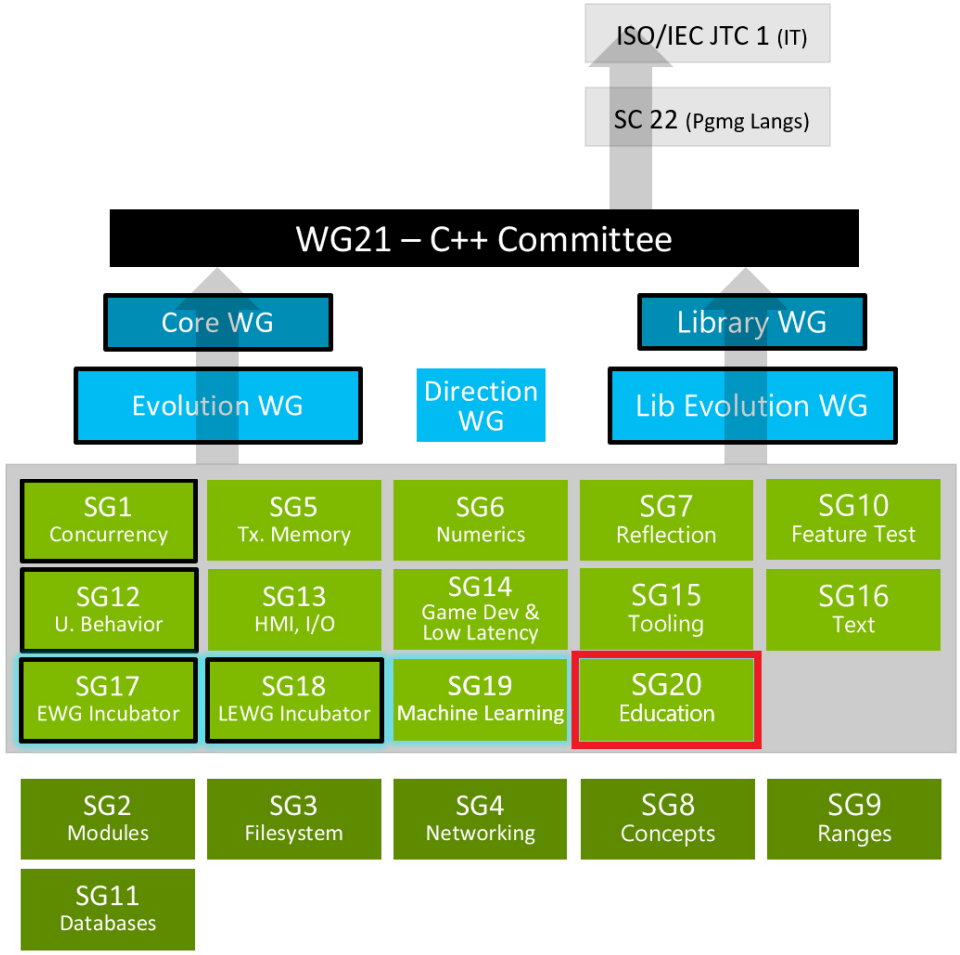
- Online C++ reference
  - Wiki format
  - Up to date
- Still only a reference
  - Few basics
  - No reading order
- Executable code



SG 20

Study Group „Education“

- Forum for *teachers*
- Started end of 2018
- charter
  - Discuss teaching strategies
  - Build curriculum guidelines



zühlke
empowering ideas

Thank you  Let‘s talk!

 Simplify C++! – www.arne-mertz.de

 @arne_mertz

 arne.mertz@zuehlke.com

 `#include<C++>` Discord (includecpp.org)

Emojis by emojione.com

