C++ Today

The Beast is Back

C++ Today

The Beast is Back

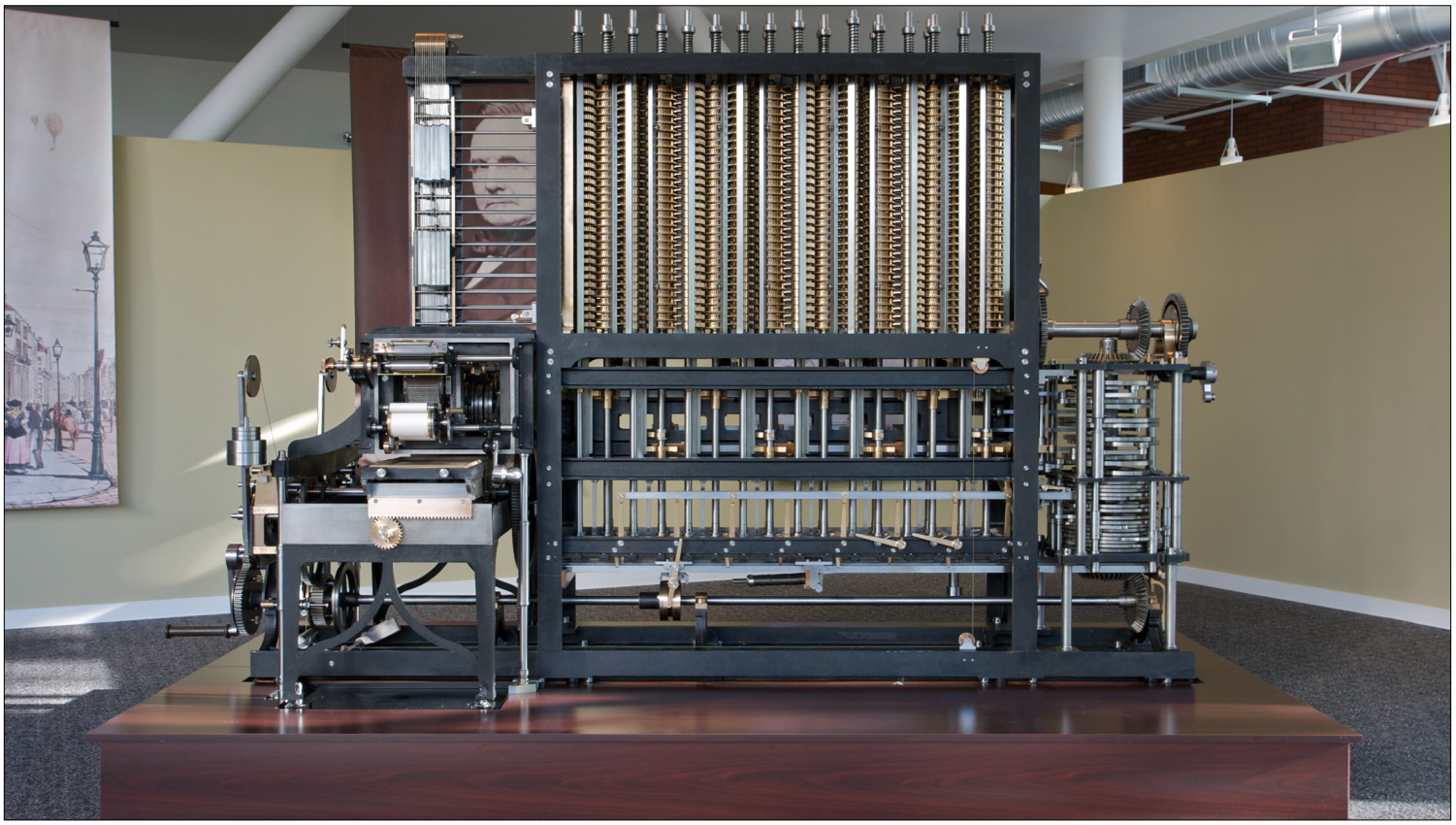
Jon Kalb

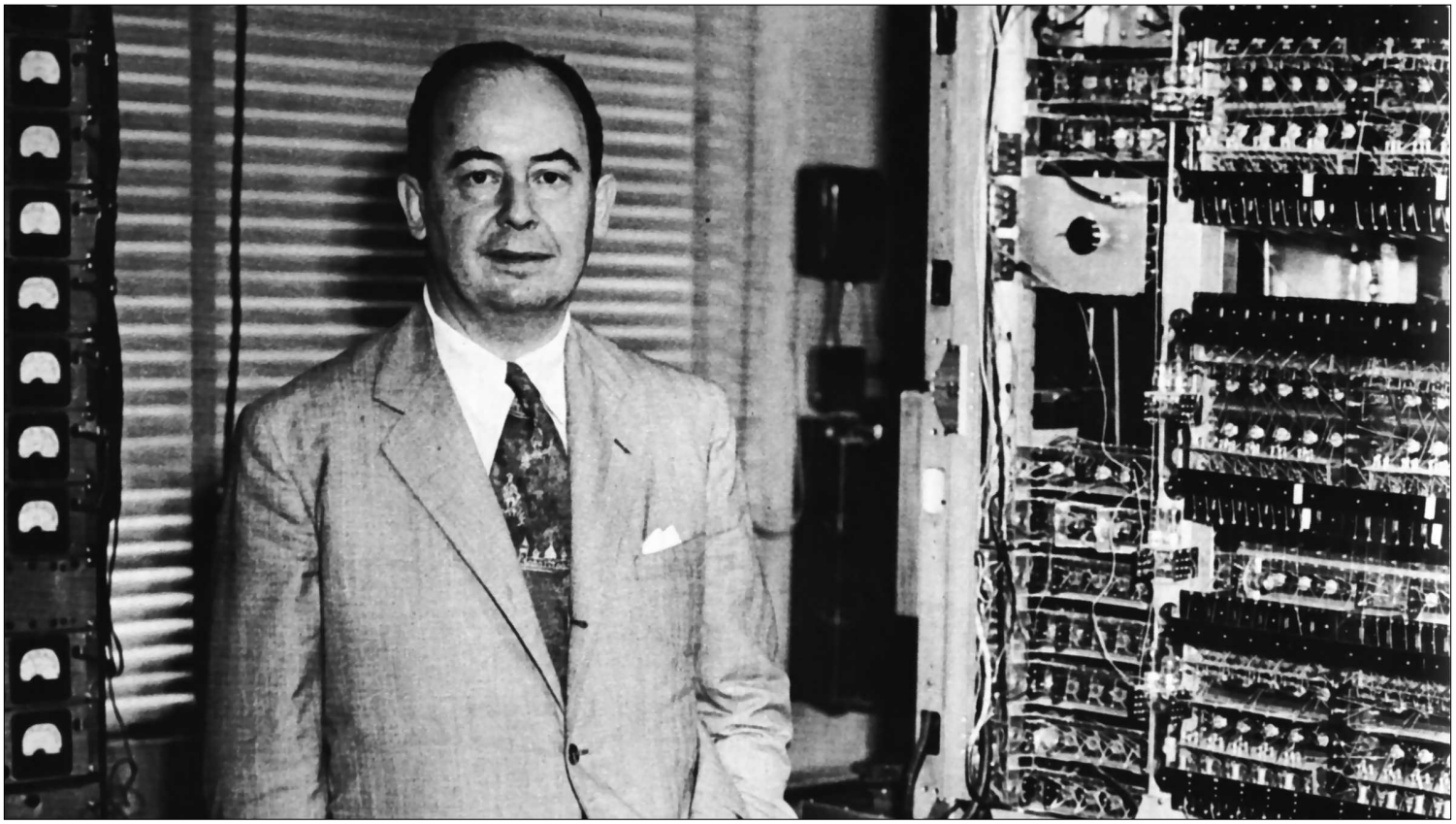
Free Download

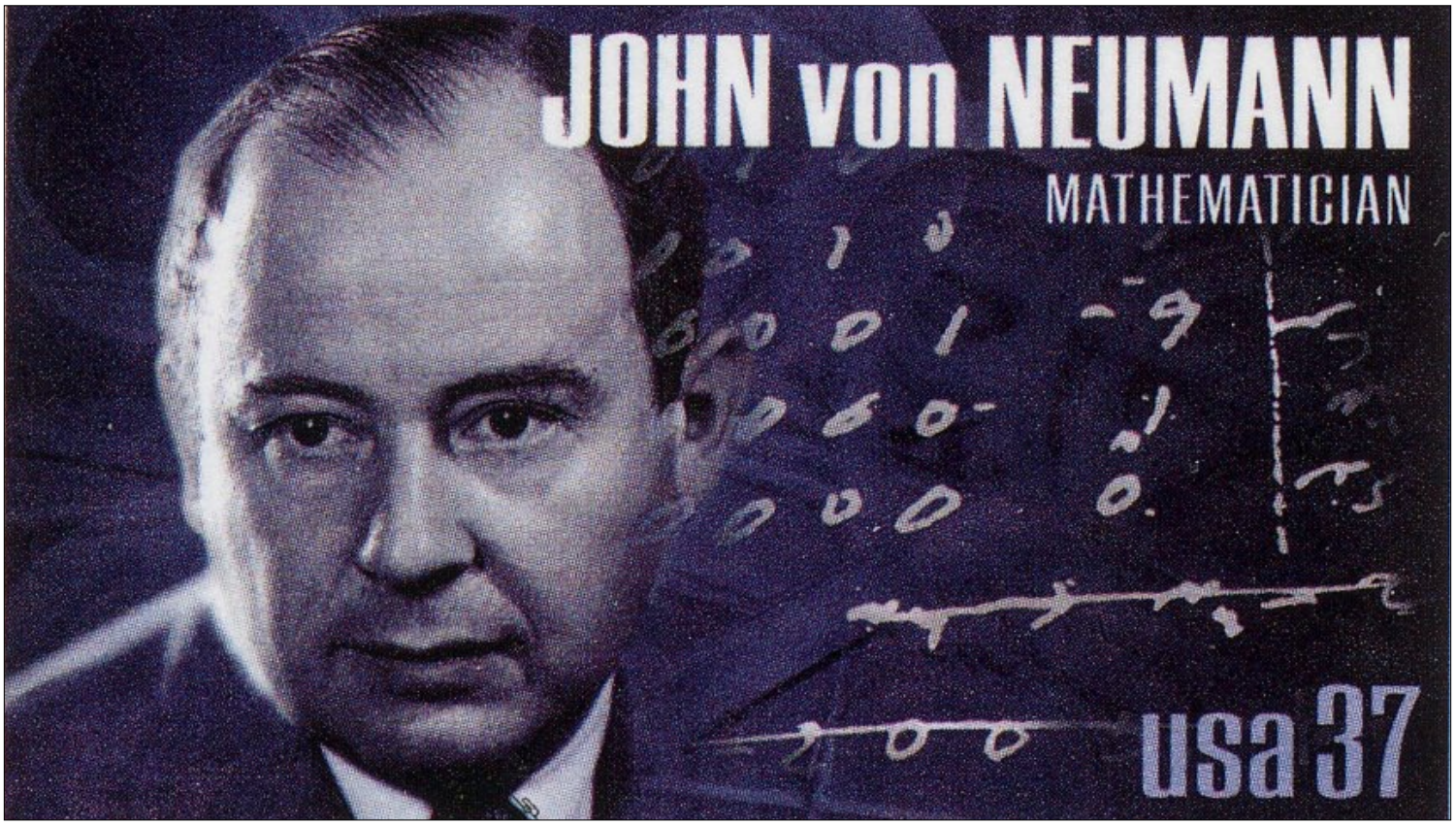
(with email address)

O'Reilly Media:

j.mp/Cpp_Today



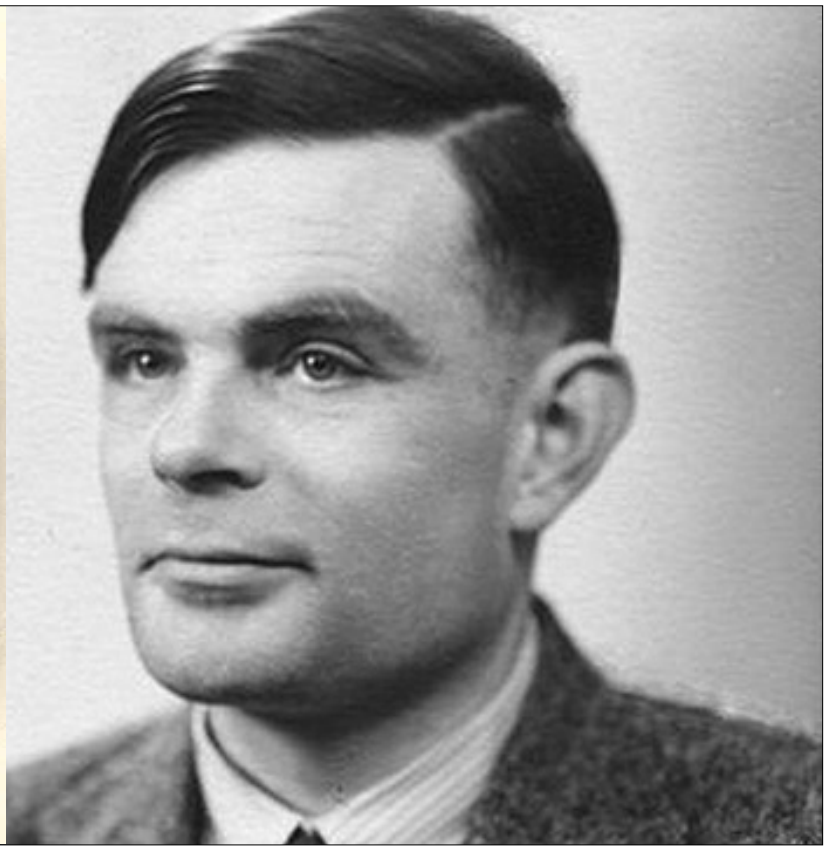




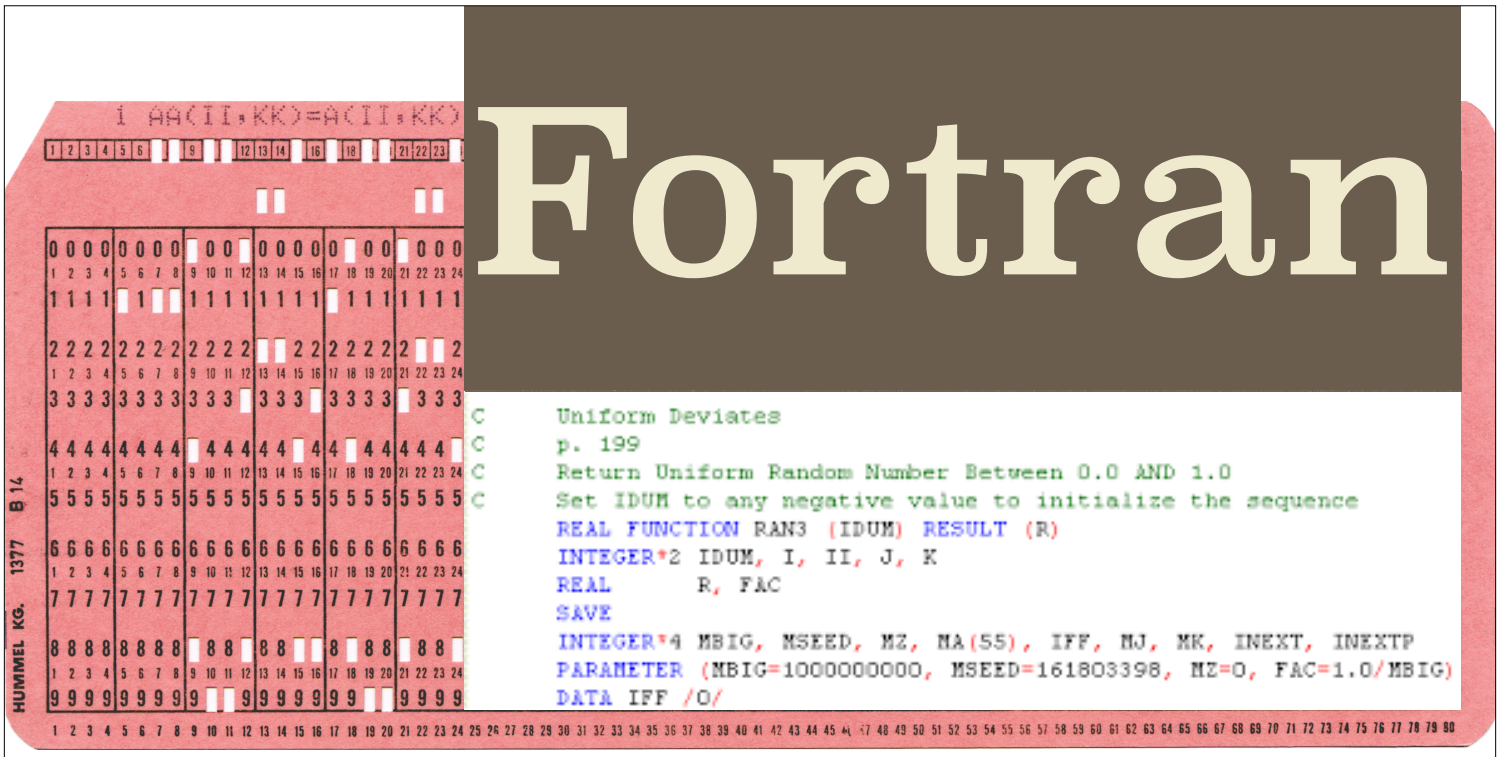

```
38:A1 DE     LDA (#DE,X)  } (ANY2+A)
3A:91 DE     STA (#DE),Y  } =(ANY2)
3C:A9 DE     LDA ##DE     } ANY2=ANY2
3E:20 10 0E  JSR DEC2PG   } -1
41:A5 DE     LDA #DE
43:C5 DB     CMP #DB
45:D0 F1     BNE #0E38
47:A5 DF     LDA #DF
49:C5 DC     CMP #DC
4B:D0 EB     BNE #0E38
4D:A1 DE     LDA (#DE,X)
4F:91 DE     STA (#DE),Y
51:[illegible]88   [illegible] DEY
[illegible]
52:A9 20     LDA #20
54:91 DB     STA (#DB),Y
56:[illegible]C0 00   [illegible]CPY #
```

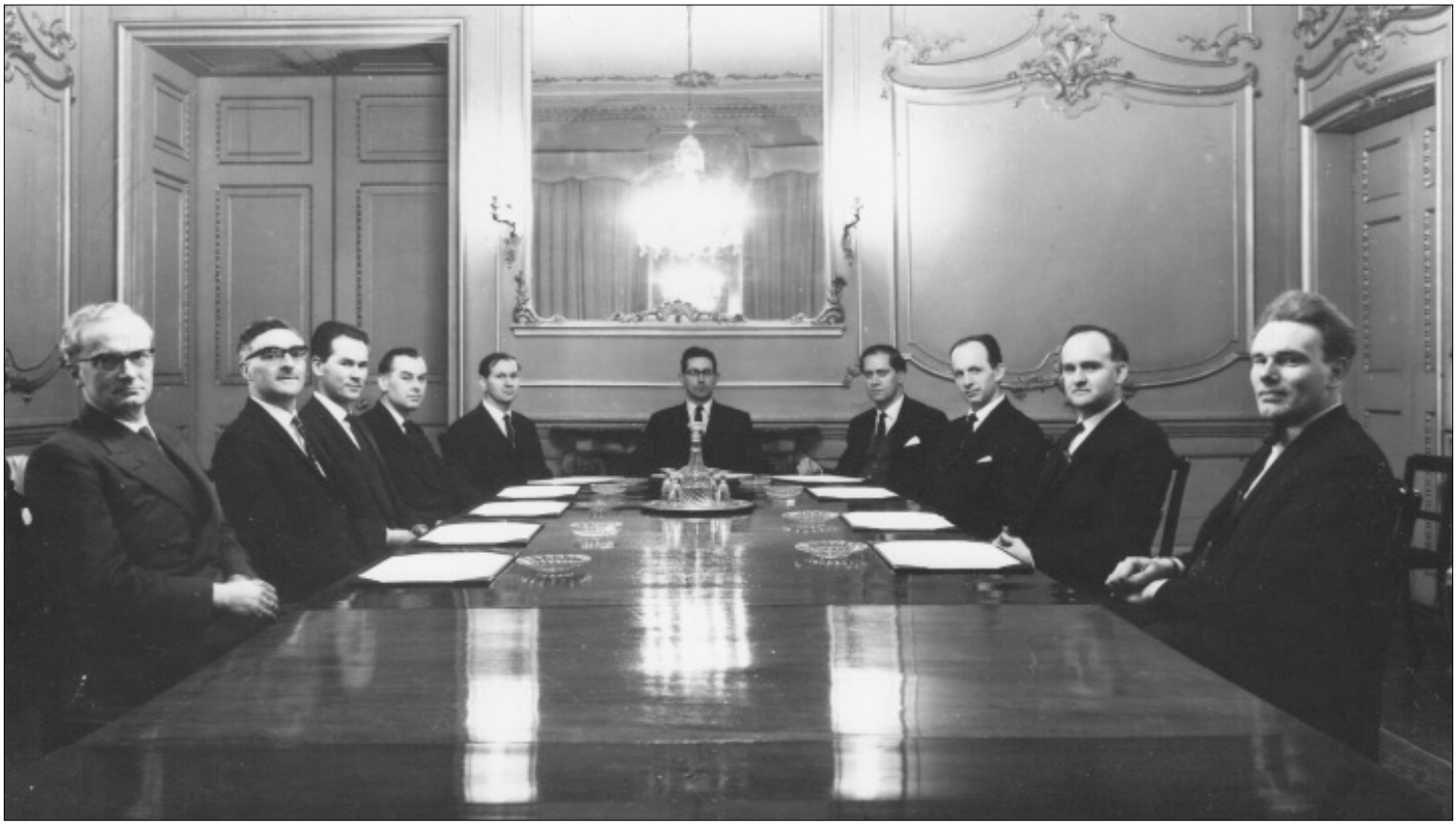
FOR-NEXT MEMORY MOVING ROUTINE.

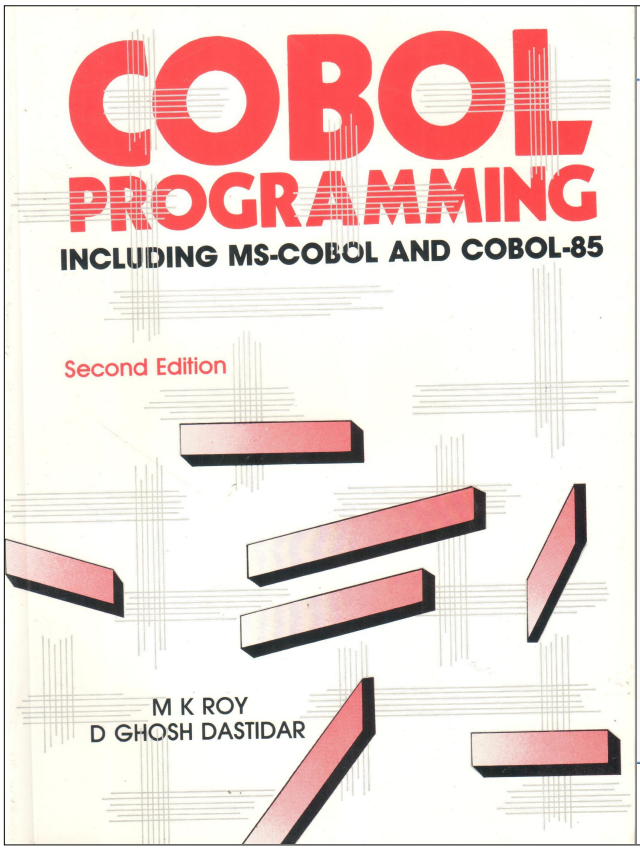
Stores spaces (#20) in ne



Fortran


I AA(II,KK)=A(II,KK)
HUMMEL KG. 1377 B 14
C Uniform Deviates
C p. 199
C Return Uniform Random Number Between 0.0 AND 1.0
C Set IDUM to any negative value to initialize the sequence
REAL FUNCTION RAN3 (IDUM) RESULT (R)
INTEGER*2 IDUM, I, II, J, K
REAL R, FAC
SAVE
INTEGER*4 MBIG, MSEED, MZ, MA(55), IFF, MJ, MK, INEXT, INEXTP
PARAMETER (MBIG=1000000000, MSEED=161803398, MZ=0, FAC=1.0/MBIG)
DATA IFF /0/

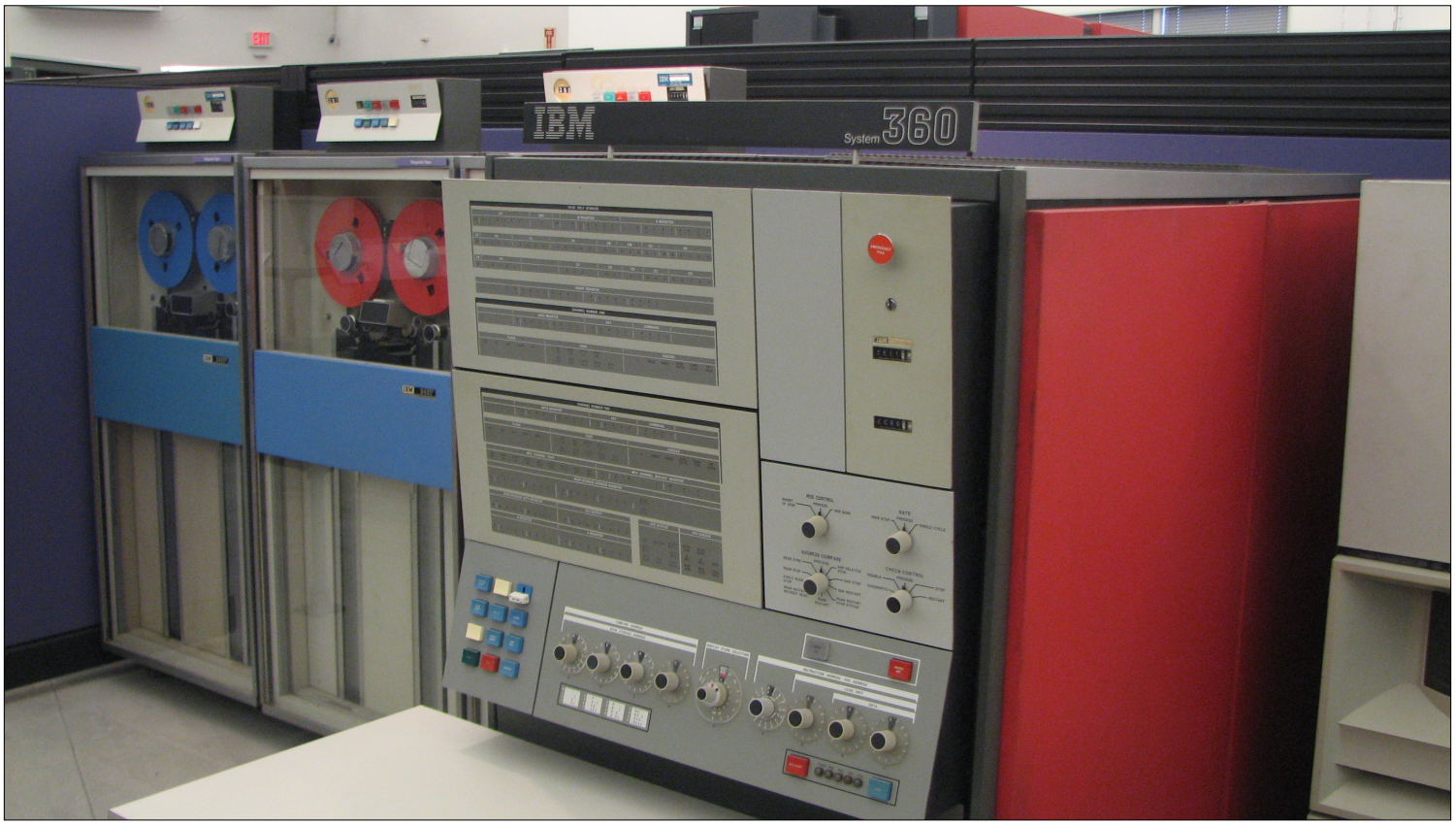





```
//COBUCLG  JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELOWRLD EXEC COBUCLG,PARM.COB='MAP,LIST,LET'
//COB.SYSIN DD *
  001   IDENTIFICATION DIVISION.
  002   PROGRAM-ID.  'HELLO'.
  003   ENVIRONMENT DIVISION.
  004   CONFIGURATION SECTION.
  005   SOURCE-COMPUTER.   IBM-360.
  006   OBJECT-COMPUTER.   IBM-360.
  0065  SPECIAL-NAMES.
  0066      CONSOLE IS CNSL.
  007   DATA DIVISION.
  008   WORKING-STORAGE SECTION.
  009   77  HELLO-CONST    PIC X(12) VALUE 'HELLO, WORLD'.
  075   PROCEDURE DIVISION.
  090   000-DISPLAY.
  100       DISPLAY HELLO-CONST UPON CNSL.
  110       STOP RUN.
//LKED.SYSLIB DD DSNAME=SYS1.COBLIB,DISP=SHR
//            DD DSNAME=SYS1.LINKLIB,DISP=SHR
//GO.SYSPRINT DD SYSOUT=A
//
```


PROBE INTERFACE
PRINTER / 360 / CALCOMP
USE IN LEFT SIDE
DATA PROP
MOTOROLA PRINTER
BRIGHTEN
Scotch
MUSIC


1960s



IBM SYSTEM/360

Now one new computer fills all your data processing needs

You can easily increase the size of SYSTEM/360 when your business grows or you want to add new applications.

You don't have to revise most of your programs. You don't have to switch to new input and output devices.

Any program that works on the smallest configuration can work on the largest.

Same goes for the programming systems. The simplest operating system, the simplest language translator or object program can work on any SYSTEM/360.

Same goes for input and output devices. Any printer, tape, storage unit, reader or terminal that works in a small configuration works in a larger one. You choose what you need now. You add new components when you need them.

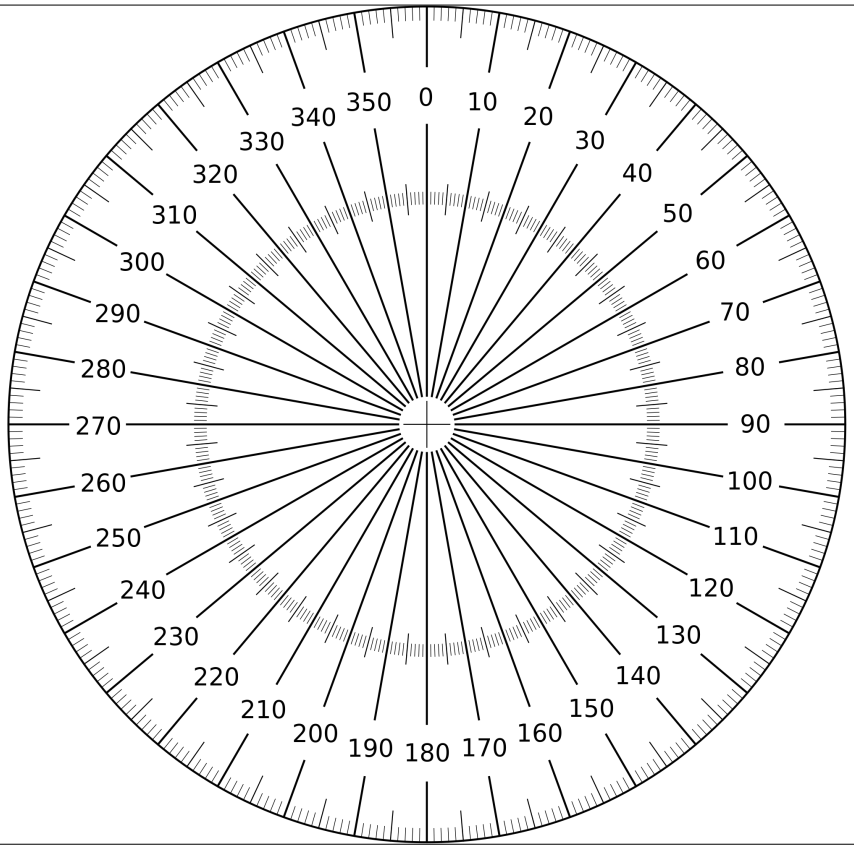
This is true from the smallest configuration to the largest configuration.

SYSTEM/360 solves today's problems. And it expands to solve tomorrow's problems, too.

It cuts today's costs....and it will also cut tomorrow's. There's never been a system quite like it.

IBM DATA PROCESSING

Litho in U.S.A. 520-0958



IBM SYSTEM/360

- Designed to be a machine for science and business
- Featured binary, decimal (BCD), and hexadecimal floating-point calculations
- First instruction set implemented in microcode
- 8-bit byte addressing

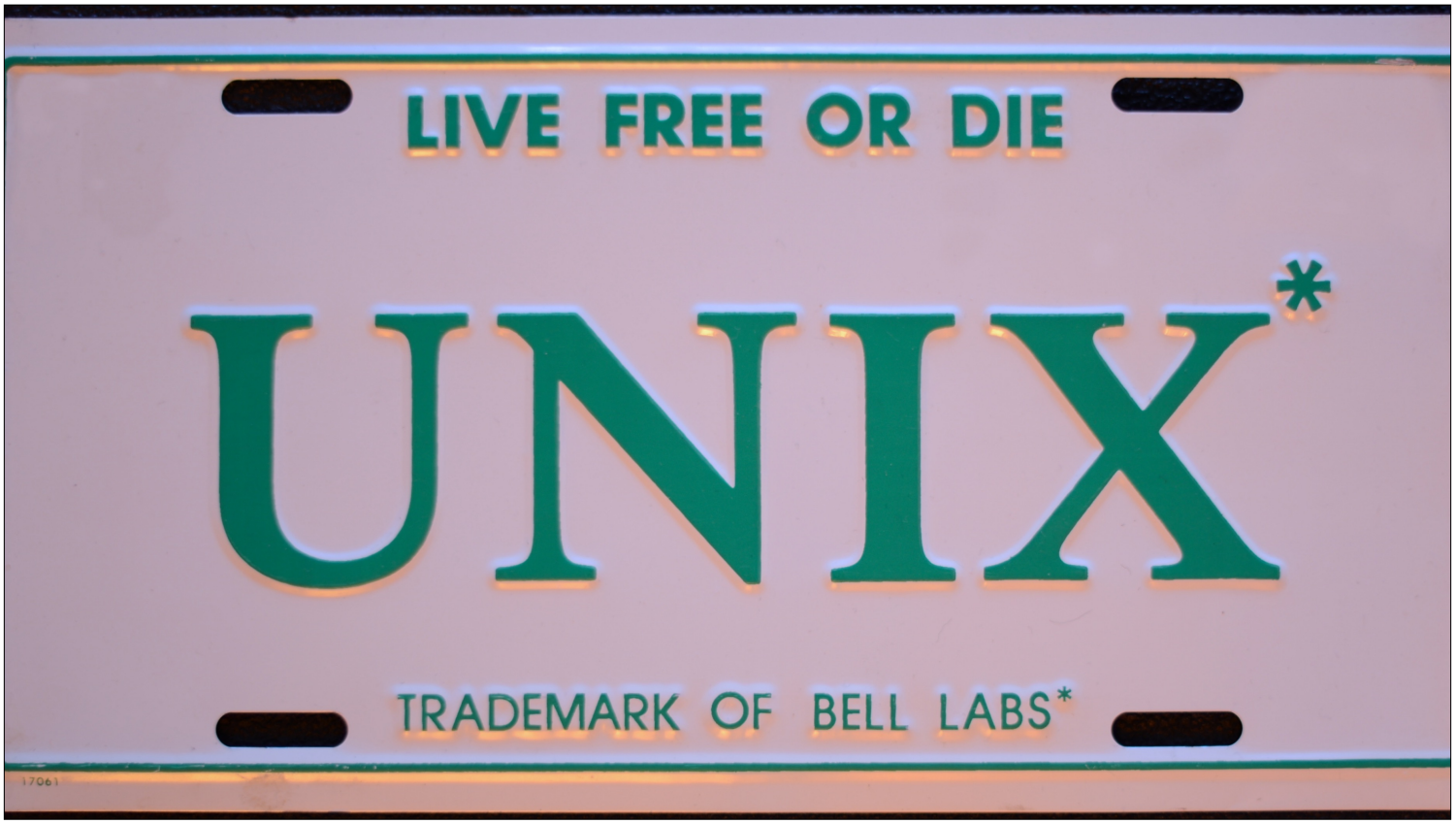
IBM SYSTEM/360

- Designed to be a machine for science and business
- Featured binary, decimal (BCD), and hexadecimal floating-point calculations
- First instruction set implemented in microcode
- 8-bit byte addressing

Established the "C machine" architecture a decade before "C."


1970s


DIGITAL EQUIPMENT CORPORATION
PDP-7
JANUS 71
SYSTEM SOFTWARE ONLY


LIVE FREE OR DIE
UNIX*
TRADEMARK OF BELL LABS*
17061

THE C PROGRAMMING LANGUAGE

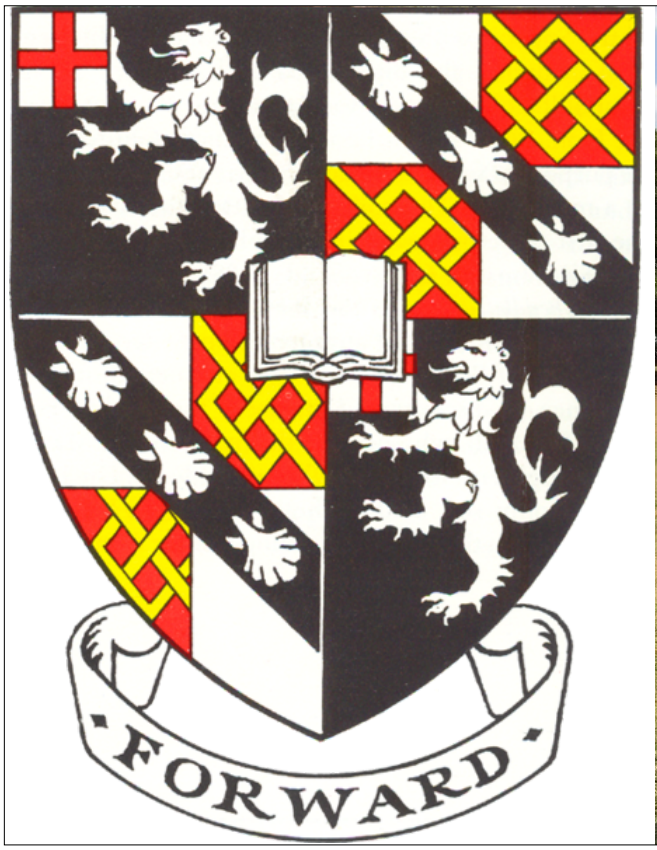
C
PROGRAMMING
LANGUAGE
CREATOR

DENNIS RITCHIE
1941 - 2011

Turing Award

For ideas fundamental to the emergence of object-oriented programming, through their design of the programming languages Simula I and Simula 67

Dahl and Nygaard at the time of Simula's development



"So I rewrote my simulator from Simula into BCPL and all of the high-level structure disappeared. All of the nice organization that had helped me debug and helped me design disappeared. But the resulting BCPL, once I had debugged it and lost half of my hair in the process, ran really fast. It could use all of the resources of the machine. It could communicate with anything on the machine and I got my data and I got my PhD…

"I came away with the opinion that I would never again want to attack a problem with tools that [were] fundamentally unsuitable. And, in particular, I don't want to make the choice between elegant, which Simula was for this problem, and efficient, which BCPL was. I want both. And that has been sort of one of my guiding lights. If you give people the choice of writing good code or fast code there's something wrong. *Good code should be fast.*"


C / C++


THE
C++
PROGRAMMING LANGUAGE

Why C++?

- High-Level Abstractions at Low Cost
- Low-Level Access When You Need It
- Wide Range of Applicability
- Highly Portable
- Better Resource Management

High-Level Abstractions at Low Cost

- user-defined types
- type templates
- generic algorithms
- type aliases
- type inference
- compile-time introspection
- runtime polymorphism
- exceptions
- deterministic destruction…

High-Level Abstractions at Low Cost

- user-defined types
  - power and expressiveness of built-in types
  - almost any thing that can be done with a fundamental type, can be done with a user-defined type
  - for example: a programmer can defined a type that functions as a
    - fundamental arithmetic type (operator overload support mathematical operations)
    - object pointer (smart pointer)
    - function pointer (function object)

High-Level Abstractions at Low Cost

- No programming paradigm is forced on the user.
- Instead there is support for:
  - procedural
  - object-based
  - object-oriented
  - generic
  - functional
  - value semantics
- freely mixing any of these paradigms

High-Level Abstractions at Low Cost

- Libraries!
- C++ can be thought of a language for making libraries
- Libraries can be created that have:
  - power
  - efficiency
  - safety
  - natural syntax
  - type-specific optimizations
  - automatic resource management
  - generic features that are as efficient as custom code

High-Level Abstractions at Low Cost

- Little or no abstraction penalty
- Bjarne Stroustrup refers to his goal as:

"the zero-overhead principle"

If you don't use the feature, you pay nothing.

If you do use it, you pay no more than you would if you coded it by hand.

Low-Level Access When You Need It

- systems-programming language:
  - low-level hardware control
  - responding to hardware interrupts
  - device drivers
  - manipulate memory in arbitrary ways — down to the bit level
  - on par with assembler, but if you really need it, allows inline assembly code
- all of this comes free with being a superset of “C”

Low-Level Access When You Need It

- management of user-defined type
- most languages create objects by
  - allocating memory from the heap
  - running construction function
- C++ can construct
  - on the heap
  - static memory
  - stack memory
  - arbitrary memory

Low-Level Access When You Need It

- can be cache-optimized
  - control necessary to exploit caches
  - avoid false sharing
- most managed language containers
  - don't hold memory in contiguous memory
  - can't exploit look-ahead buffers
- C++ containers can do this
  - arrays, vectors, deques,
  - user-defined containers

Wide Range of Applicability

- don't solve the specific problem, solve the general one
- scale
  - software engineers are all about scale
  - algorithms, but also languages
- low end
  - embedded app, device driver, CGIs, mobile apps
  - "you only pay for what you use"
  - low-memory footprint
- high end
  - projects with
    - hundreds of engineers
    - scores of modules
  - separate module compilation

Highly Portable

- “C machine” model
  - minimalistic requirements
  - universally supported by hardware
- one or more C++ toolchains on almost all platforms
- C++ is the only high-level language alternative available on all of the top mobile platforms
- C++ can be written to support all these platforms without rewriting

Highly Portable

- even a “perfect” language has no value if we can’t build it on our target platform
- factors outside the language
  - tool chains
  - analyzers / non-build tools
  - experienced engineers
  - software libraries
  - books / instructional materials
  - trouble shooting support
  - training opportunities
- large installed base / industry support

Better Resource Management

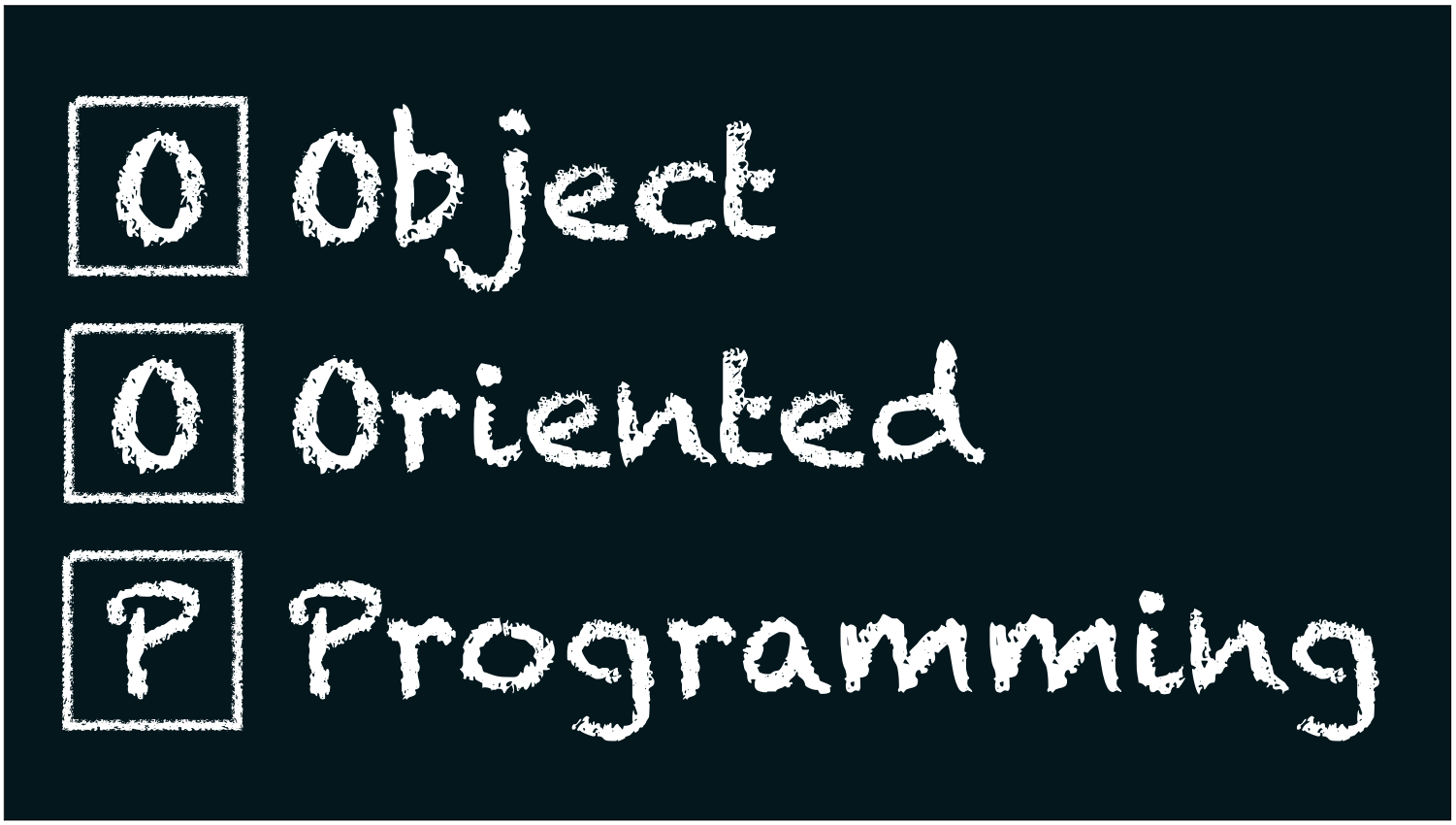
- garbage collection
- the good…
  - no leaks or double dispose
  - very, very annoying problems
- … and the bad
  - memory not released until unspecified time later
    - … if at all
    - not under the control of the programmer
    - may result in “freezing”
    - requires a large buffer of unused memory

Better Resource Management

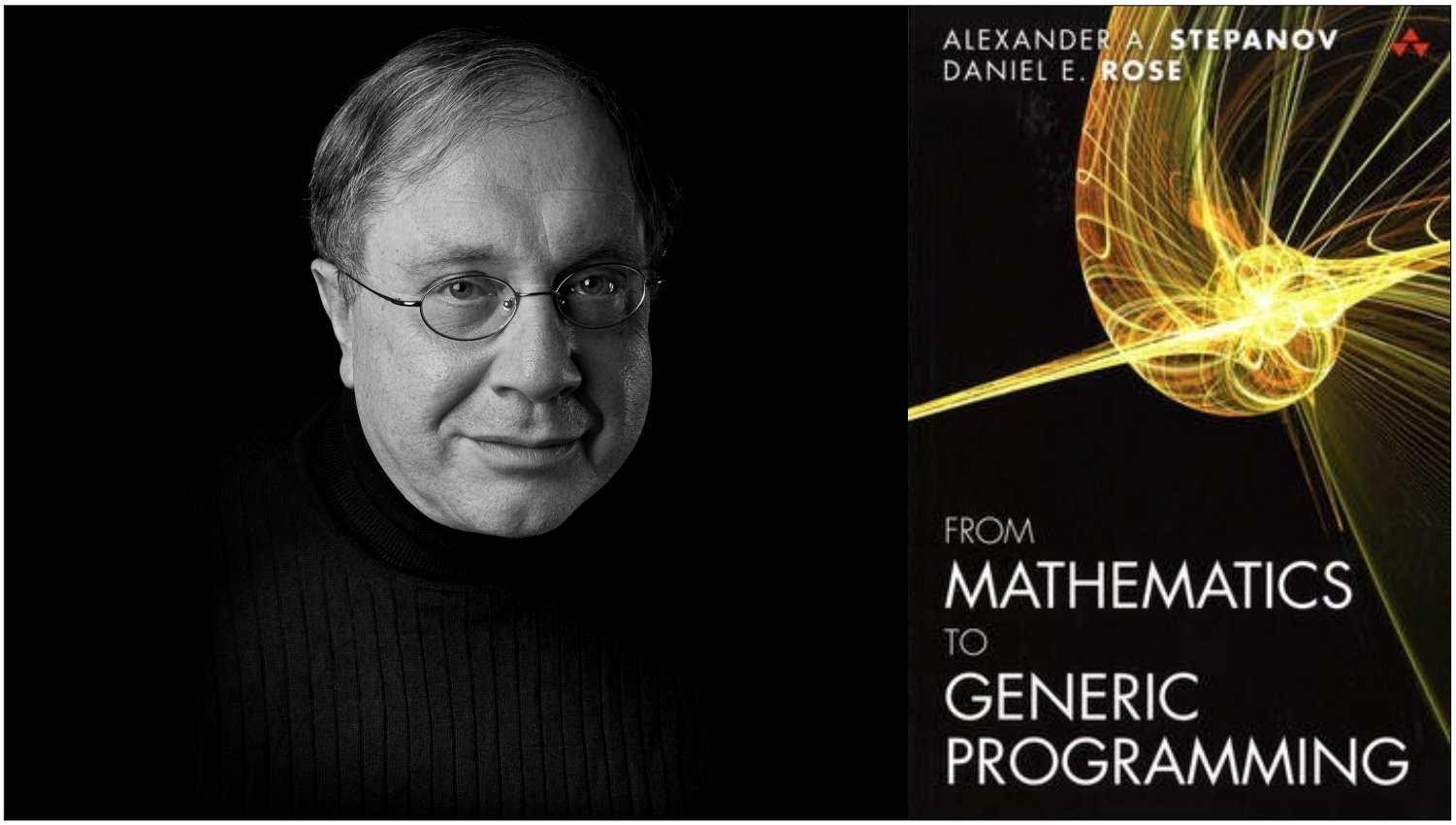
- C++ requires a better solution
- must generalize
  - memory is only one resource
  - other resources need to be similarly managed
  - release must happen immediately
- deterministic destruction
- RAII
  - resource acquisition is initialization
  - responsibility acquisition is initialization
  - every responsibility that must be discharged is
    - an object with appropriate lifetime (usually on the stack)
    - discharged when the object goes out of scope
  - “finally” violates DRY

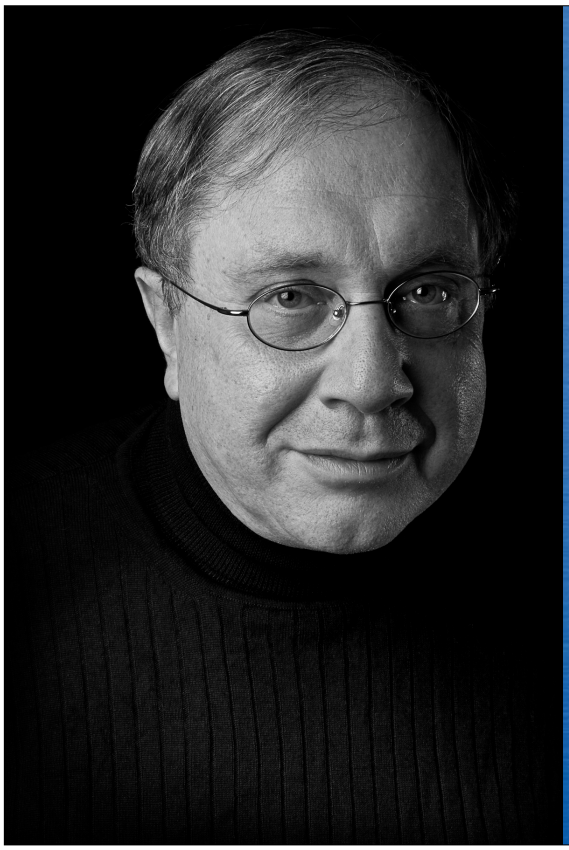
Michael Caisse

λ ciere


O Object
O Oriented
P Programming


ISO
International
Organization for
Standardization


ALEXANDER A. STEPANOV
DANIEL E. ROSE
FROM
MATHEMATICS
TO
GENERIC
PROGRAMMING



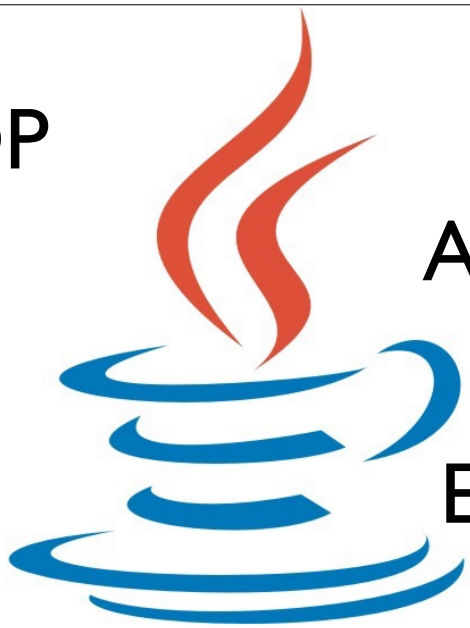
The C++ Standard Template Library

boost
C++ LIBRARIES

Supports OOP

Language of the Web

Language of the Future



Almost as fast

Easier to learn and teach

Java™

Almost as fast

- desktop computers in the 2000s were so powerful that the “need for speed” seemed to be sated
- Why write in a much more complicated language to gain only a little improvement in speed?
- C++ came with
  - a lot baggage from C syntax
  - a lot of complexity to get a little more power
    - templates
    - generic programming
    - operator overloads
    - pointers are hard!
  - managed languages were seen as “fast enough”

Managed Languages

- Two virtual machines
- Java Virtual Machine
  - Java, Scala, Jython, Jruby, Clojure, Groovy
- Microsoft's Common Language Interface
  - C#, F#, IronPython, IronRuby, C++/CLI
- colleges learned that it is easier to teach managed languages than "C machine" languages
- pointers are hard!

International Organization for Standardization

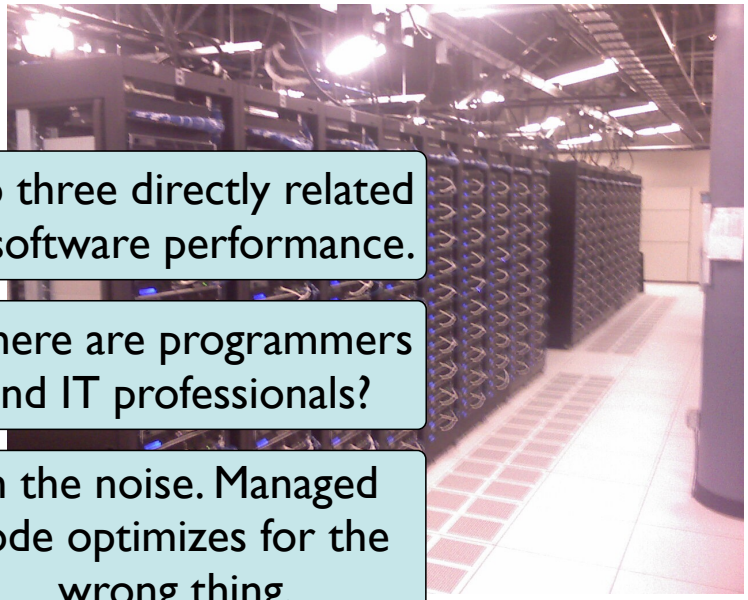
International Organization for Standardization

Performance Matters

- mobile devices
  - performance with less expensive process
  - better battery performance
- cloud computing
  - performance per dollar of hardware
  - performance per watt of power
  - performance per watt of cooling

Performance Matters

- James Hamilton of AWS
  - costs of modern high-scale data centers:
    - servers
    - power distribution and cooling
    - power
    - networking equipment
    - other infrastructure



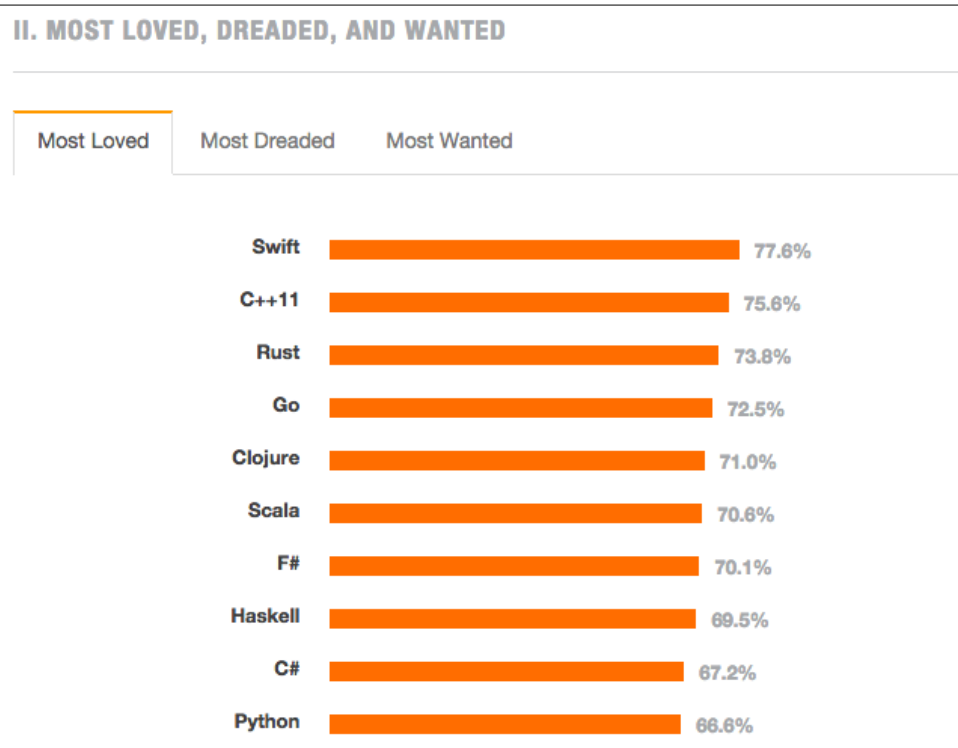
Top three directly related to software performance.

Where are programmers and IT professionals?

In the noise. Managed code optimizes for the wrong thing.

2011

Feels like a new language!



source:
StackOverflow
2015 Developer Survey

C++11

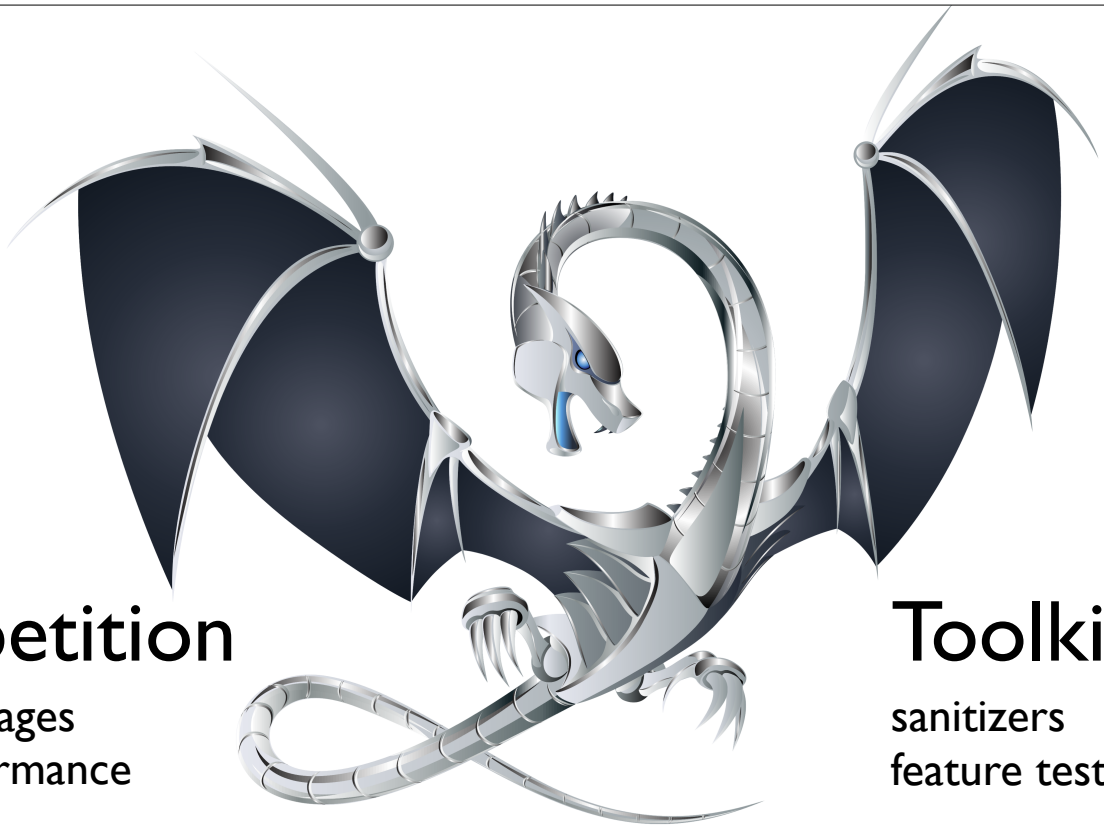
- page count
  - C++03: 776
  - C++11: 1353
- “simplifying” by adding
  - auto
  - range-base for loops
  - >>
  - enums more consistent
  - uniform initialization

C++11

- lambda expressions
  - simplifying
  - leveraging generic algorithm
  - more functional
- improve support for characters
  - character sets / Unicode
  - custom literals
- “TR1” libraries
  - tuples
  - smart pointers

C++11

- better support for library authors
  - better introspection of types
  - variadic templates
  - perfect forwarding
- big new areas
  - multithreading
  - move semantics



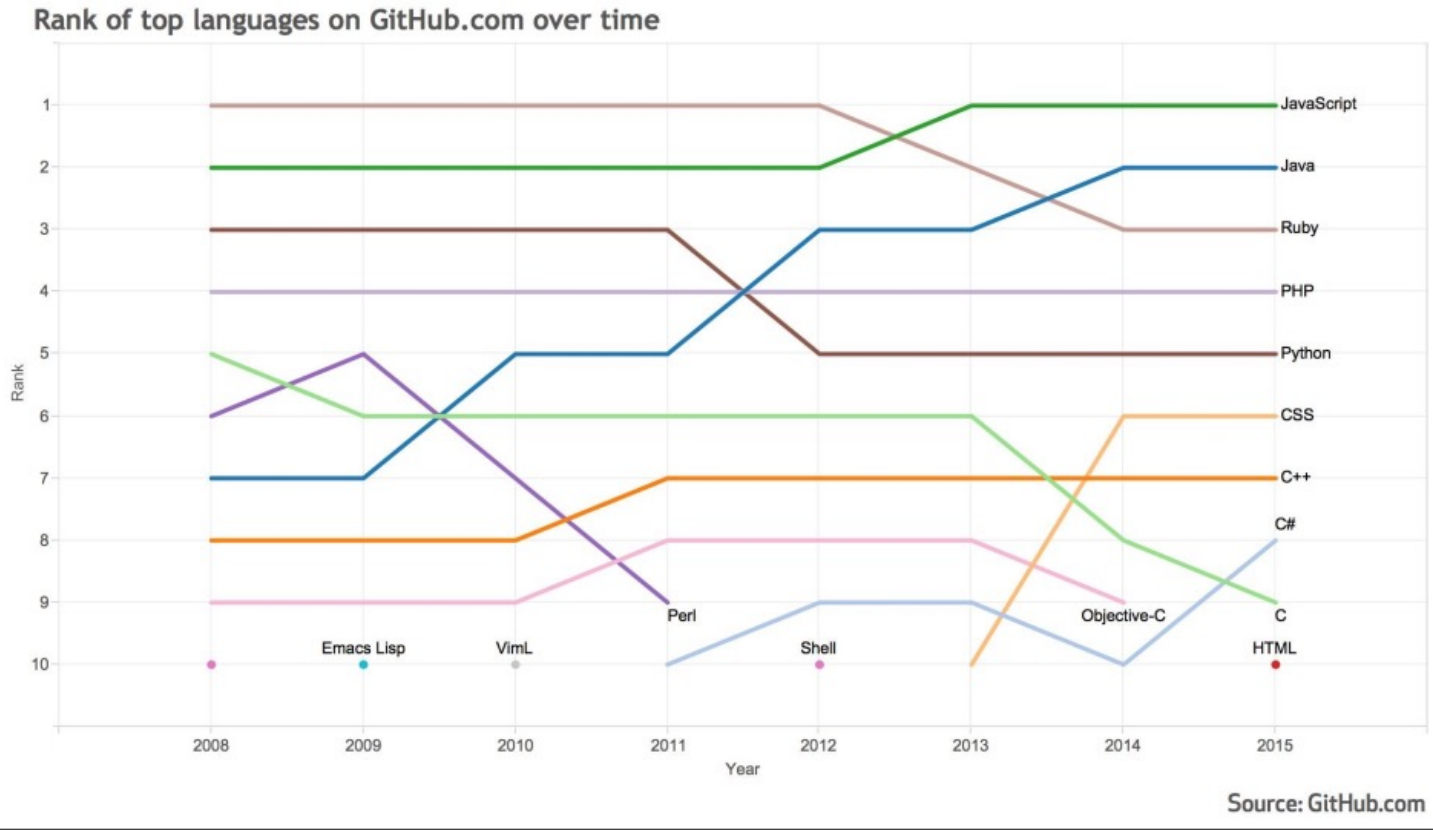
Competition

error messages
build performance

Toolkit

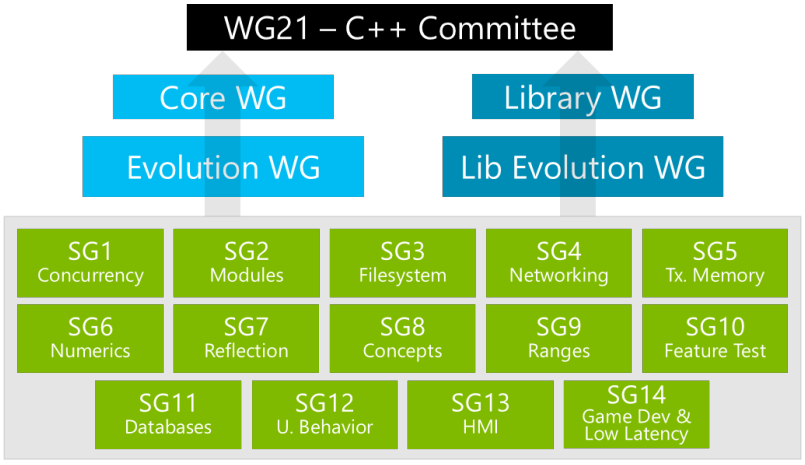
sanitizers
feature testing


I WANT YOU
TO OPEN SOURCE!
WWW.MIL-OSS.ORG


Rank of top languages on GitHub.com over time
Rank
1
2
3
4
5
6
7
8
9
10
2008
2009
2010
2011
2012
2013
2014
2015
Year
JavaScript
Java
Ruby
PHP
Python
CSS
C++
C#
C
HTML
Objective-C
Shell
Perl
VimL
Emacs Lisp
Source: GitHub.com

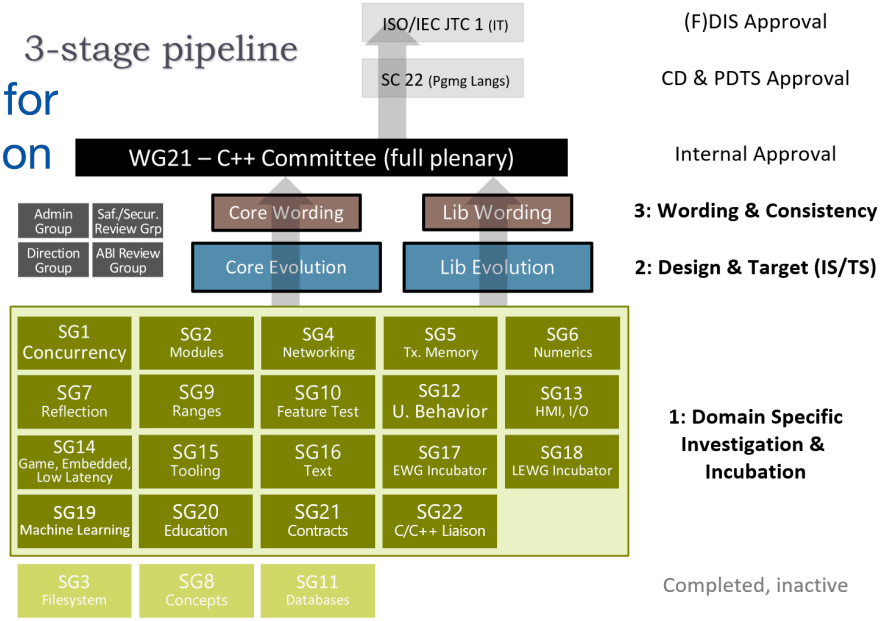
International Organization for Standardization

International Organization for Standardization



International Organization for Standardization

3-stage pipeline



(F)DIS Approval

CD & PDTS Approval

Internal Approval

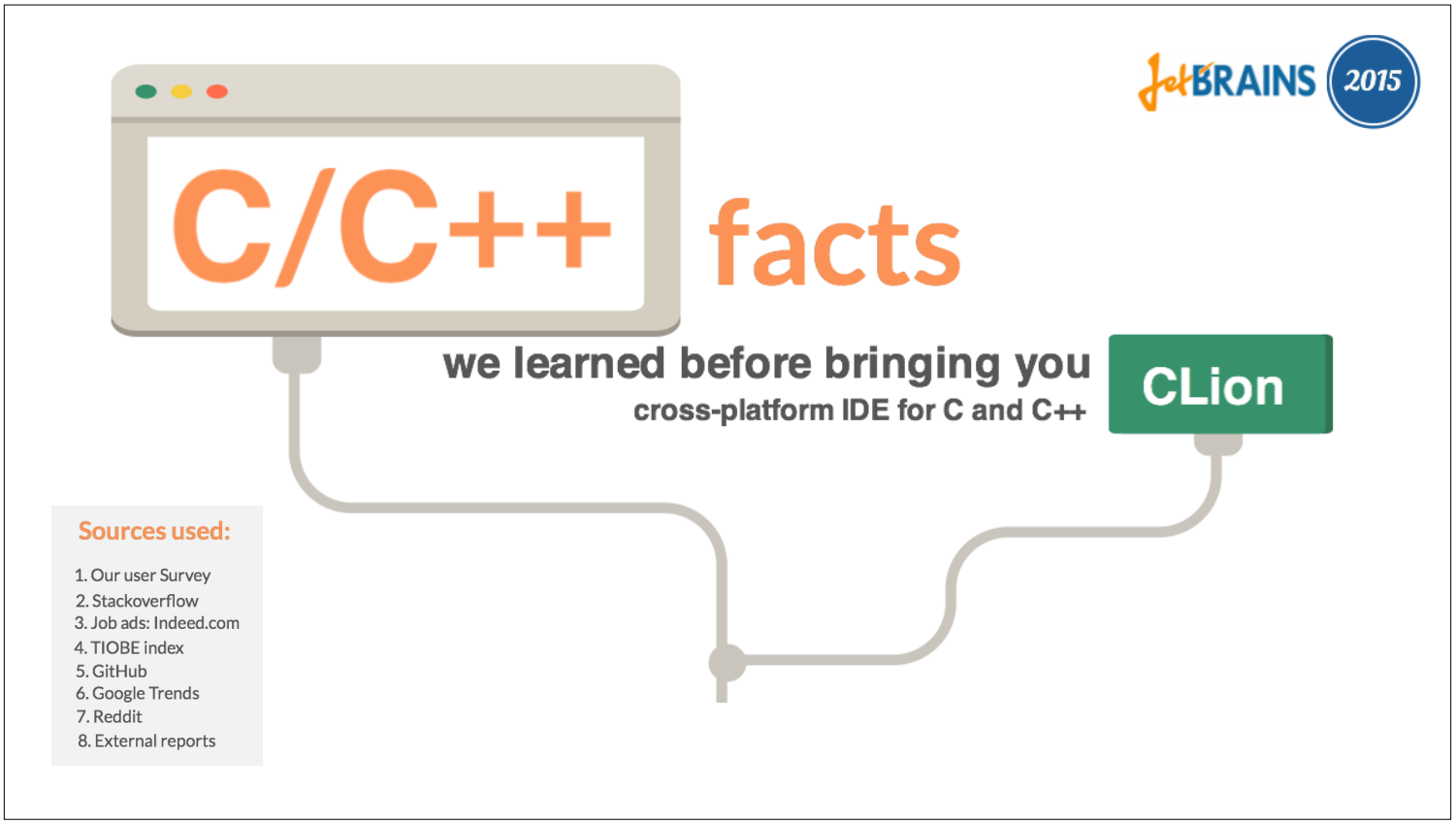
3: Wording & Consistency

2: Design & Target (IS/TS)

1: Domain Specific Investigation & Incubation

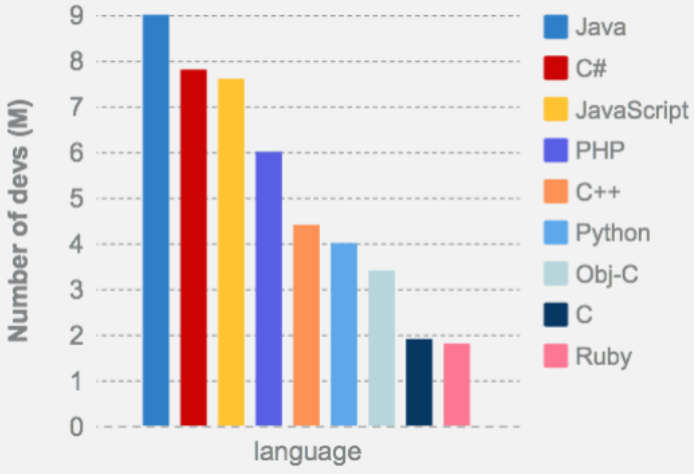
Completed, inactive


Just Released
CLion
Cross-platform C/C++ IDE
by JetBrains


JetBRAINS 2015
C/C++ facts
we learned before bringing you
cross-platform IDE for C and C++
CLion
Sources used:
1. Our user Survey
2. Stackoverflow
3. Job ads: Indeed.com
4. TIOBE index
5. GitHub
6. Google Trends
7. Reddit
8. External reports

~4.4 million C++ devs
~1.9 million C devs

There are 4.4 million C++ developers and nearly 2 million C developers in the world.



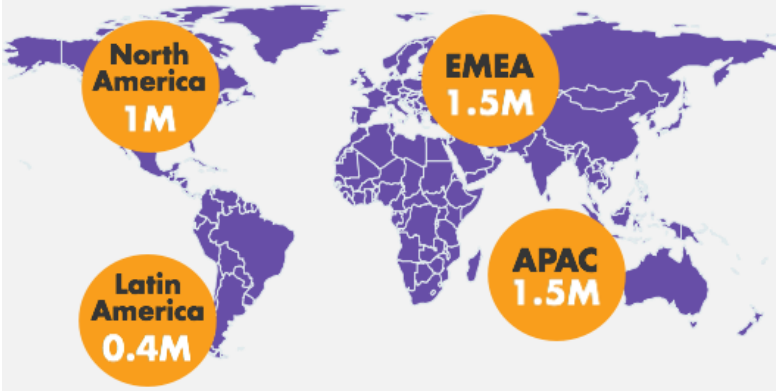
As many C++ developers as Python developers

We've analyzed a range of sources to estimate the number of worldwide developers using the most popular languages.

C++ is on par with Python, while the adoption of C is similar to that of Ruby.

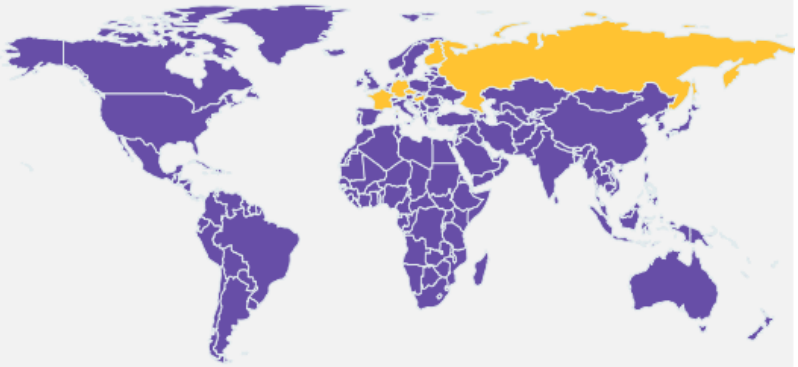
C++ developers by world region

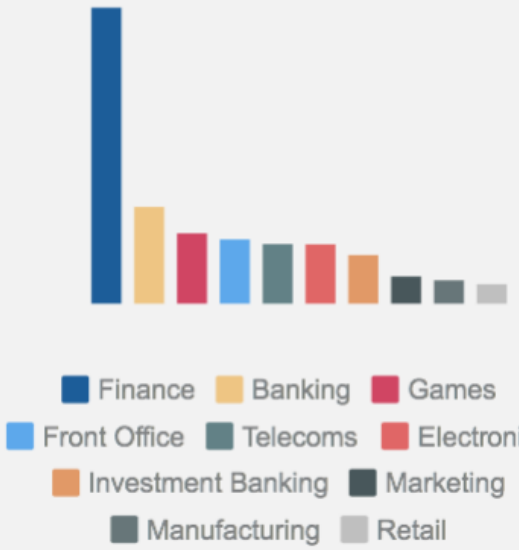
EMEA and Asia-Pacific are the most densely populated regions with regard to developers in general and C++ developers in particular.



Where C++ is relatively ahead of other languages

C++ is relatively more popular than other languages and technologies in Russia, Czech Republic, Hungary, France, Singapore, Finland, Israel and Germany.



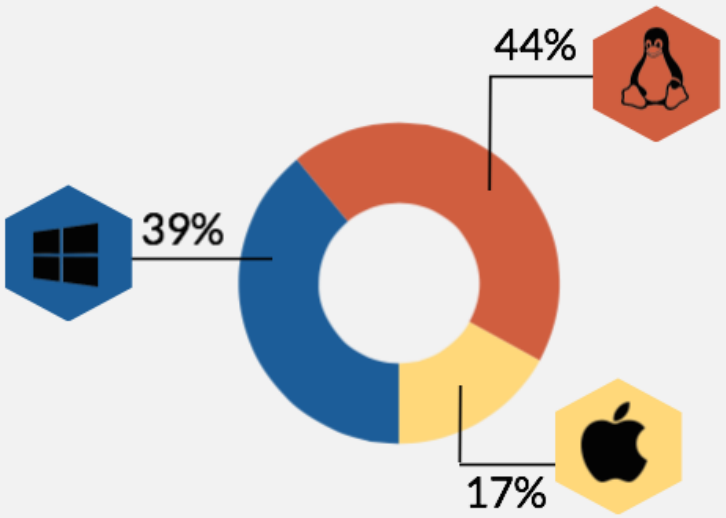


Industry distribution

Based on the C/C++ job ads we analyzed, C++ is most used in industries like Finance and Banking.

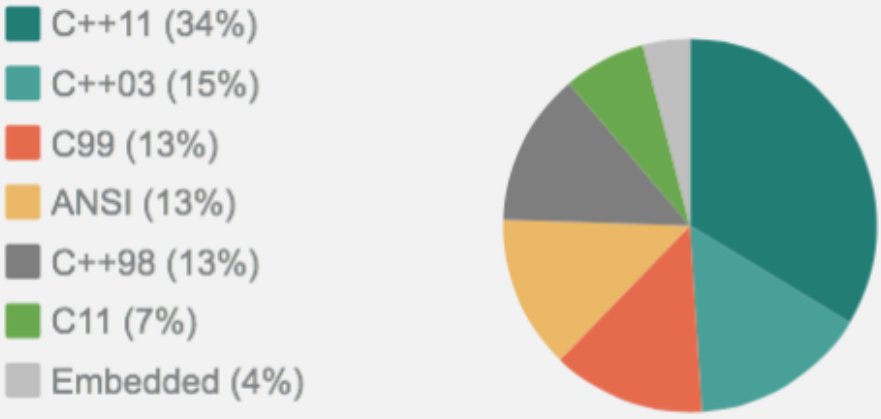
Languages used with C++

Judging from job ads, the languages that are most typically used together with C++ include Java, C, C#, Python, SQL and JavaScript.



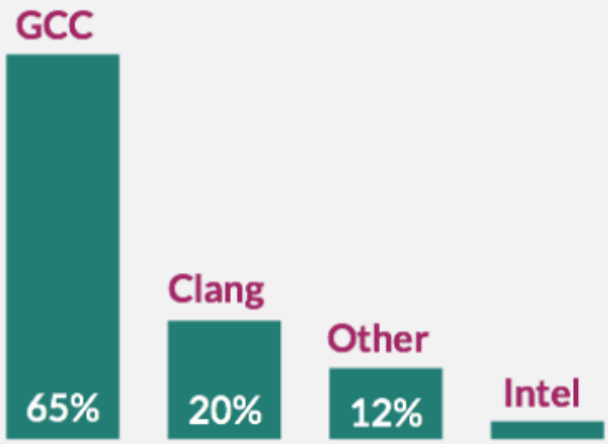
C++ developers by platform

The most popular platform is Linux, used by 44% of C/C++ developers, followed by Windows (39%) and OS X (17%).



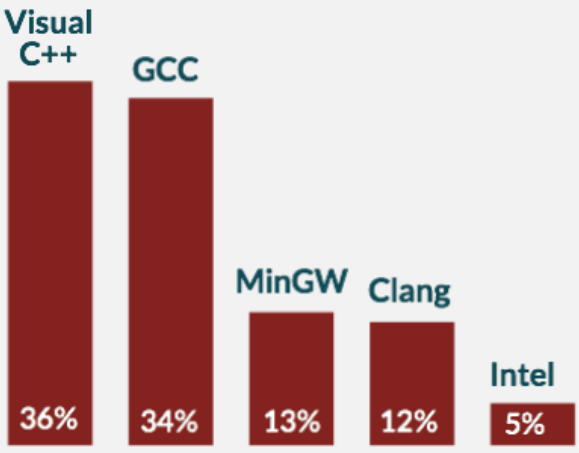
C++ versions

The most popular C++ version is currently C++11, with a share of 34%.



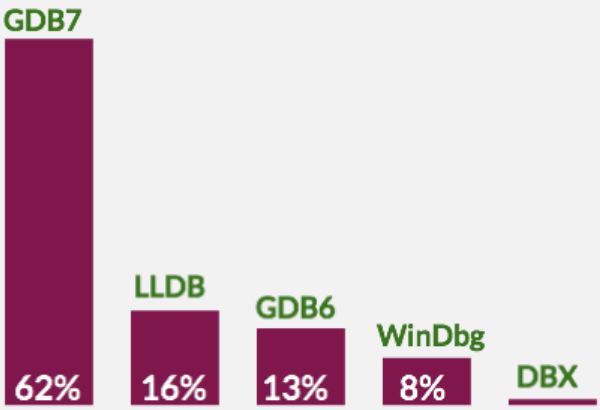
Popular C++ compilers

GCC is by far the most popular compiler with 65%, followed by Clang with 20%. Together they cover 85% of all C++ developers.



C++ compilers on Windows

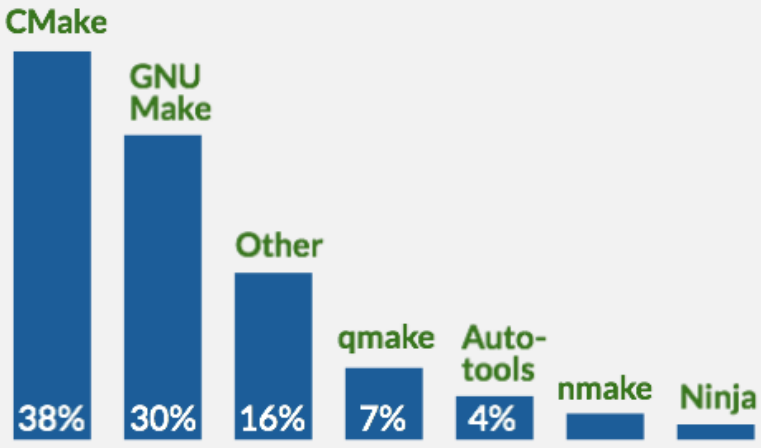
The title of the most popular C++ compiler on Windows is a virtual tie between Visual C++ (36%) and GCC (34%). MinGW (with no exact compiler name) and Clang are also tied with ~12% each. Intel comes in last at 5%.



Popular C++ debuggers

The runaway leader among debuggers used by C++ developers is GDB7 (62%), with LLDB and GDB6 trailing with 16% and 13%, respectively.

The distribution on OS X is noticeably different: LLDB is ahead at 39% with GDB7 a close second at 32%.



Popular C++ build systems

CMake and GNU Make build systems are the two close leaders, together accounting for 68% of all C++ developers.

Compiler + Build tool + Debugger

GCC + CMake + GDB7 toolchain takes the top spot for all C++ developers.

On OS X, however, that honor goes to Clang + CMake + LLDB.

Over five years.

CppCast

```
auto CppCast = pod_cast<C++>("http://cppcast.com");
```

Cpp.chat

Algorithms + Data Structures = Programs

https://cpplang-inviter.cppalliance.org/

reddit CPP

Milestone	Date	Note
60K	Feb 19, 2018	/r/cpp hits 60K subscribers
TREND	Jan 10, 2018	/r/cpp is trending – Trend thread
50K	Jun 15, 2017	/r/cpp hits 50K subscribers
40K	Sep	
30K	Jul 2	
20K	Apr 17, 2014	/r/cpp hits 20K subscribers
TOP 1K	Nov 30, 2012	/r/cpp enters TOP 1K subreddits
10K	Oct 30, 2012	/r/cpp had 10K subscribers when
Creation	May 26, 2008	/r/cpp is born

192K users

stackoverflow

c++ × 536057

a general-purpose programming language. It was originally d tension to C, and keeps a sim s now a

700 K ???

day, 1485 this week

177 channels

20K members

~ 1K weekly active users

cppreference.com

Page Discussion

C++ reference

C++98, C++03, C++11, C++14, C++17, C++20, C++23

Cprogramming.com

Your resource for C and C++

https://cpplang-inviter.cppalliance.org/

Join **Cpplang** on Slack.

18108 users are registered so far.

you@yourdomain.com

I'm not a robot

reCAPTCHA

Privacy - Terms

GET MY INVITE

or sign in.

https://cpplang.now.sh/

Join **Cpplang** on Slack.

136 users online now of **7092** registered.

you@yourdomain.com

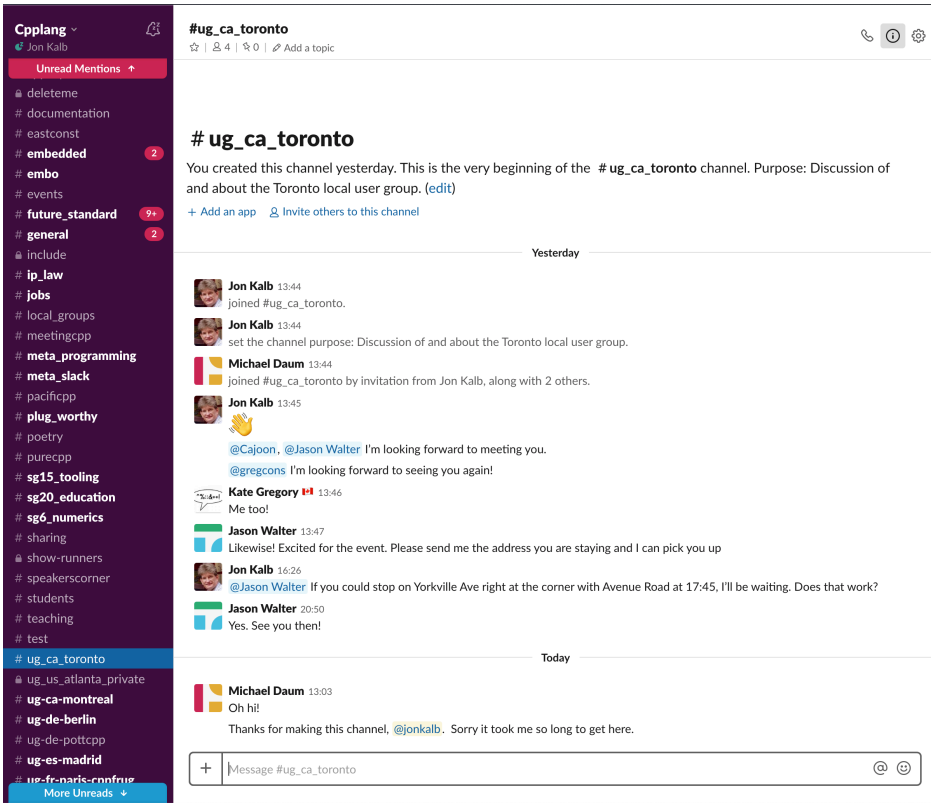
I'm not a robot

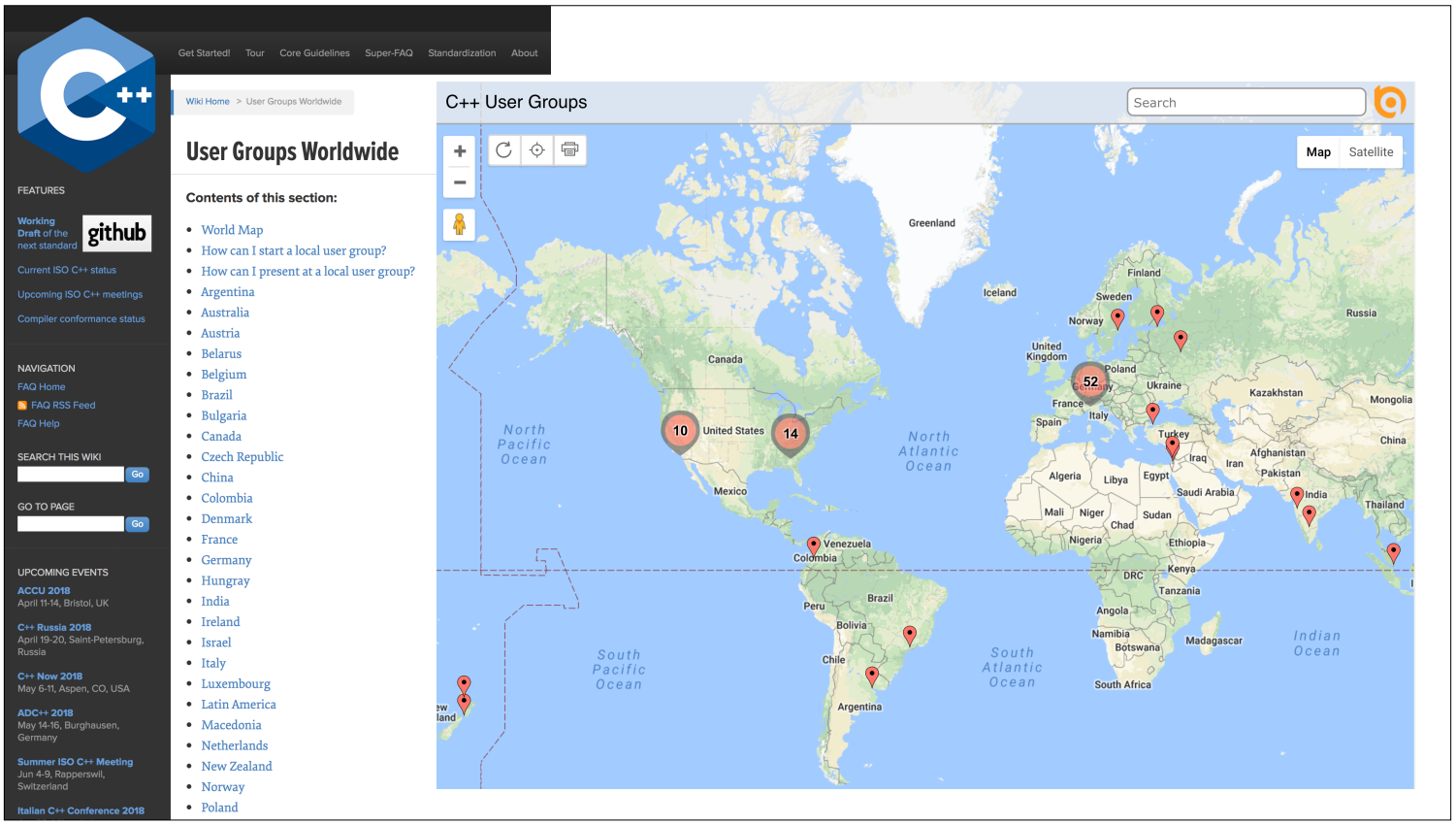
reCAPTCHA

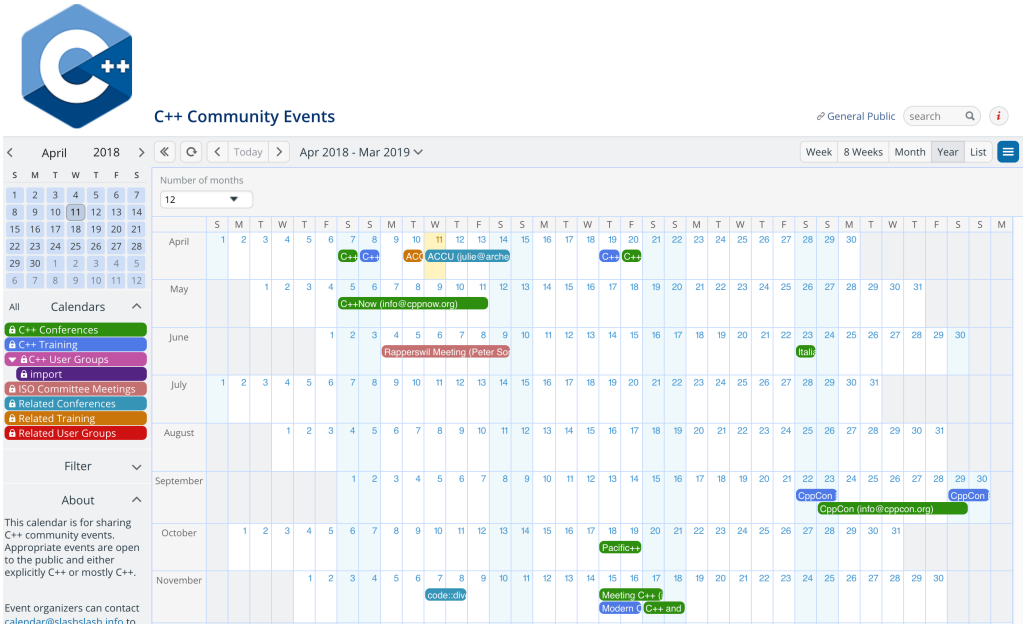
Privacy - Terms

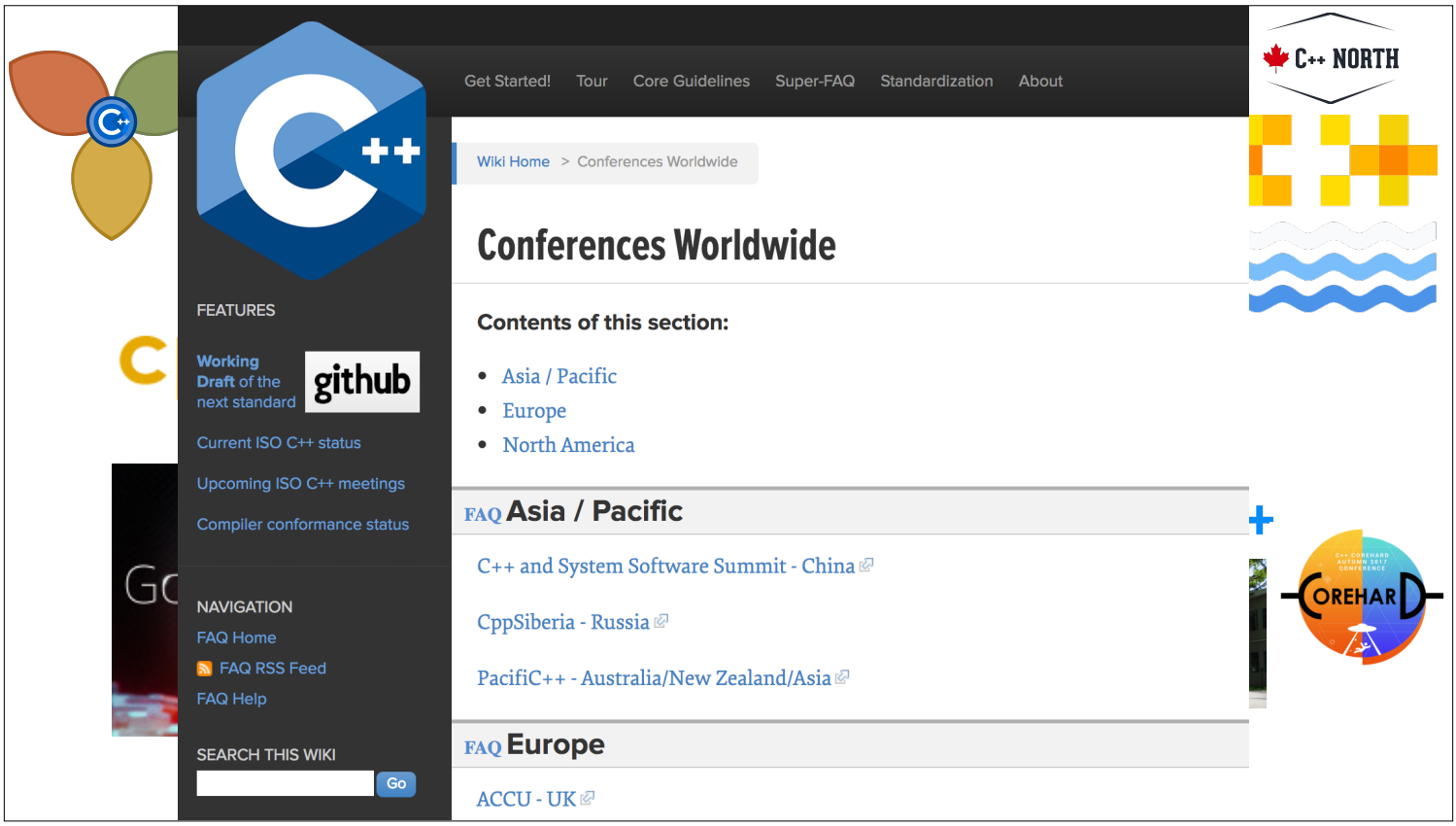
GET MY INVITE

or sign in.




Get Started!
Tour
Core Guidelines
Super-FAQ
Standardization
About
FEATURES
Working Draft of the next standard
github
Current ISO C++ status
Upcoming ISO C++ meetings
Compiler conformance status
NAVIGATION
FAQ Home
FAQ RSS Feed
FAQ Help
SEARCH THIS WIKI
Go
GO TO PAGE
Go
UPCOMING EVENTS
ACCU 2018
April 11-14, Bristol, UK
C++ Russia 2018
April 19-20, Saint-Petersburg, Russia
C++ Now 2018
May 6-11, Aspen, CO, USA
ADC++ 2018
May 14-16, Burghausen, Germany
Summer ISO C++ Meeting
Jun 4-9, Rapperswil, Switzerland
Italian C++ Conference 2018
Wiki Home > User Groups Worldwide
User Groups Worldwide
Contents of this section:
World Map
How can I start a local user group?
How can I present at a local user group?
Argentina
Australia
Austria
Belarus
Belgium
Brazil
Bulgaria
Canada
Czech Republic
China
Colombia
Denmark
France
Germany
Hungray
India
Ireland
Israel
Italy
Luxembourg
Latin America
Macedonia
Netherlands
New Zealand
Norway
Poland
C++ User Groups
Search
Map
Satellite
Greenland
Iceland
Canada
United States
Mexico
Venezuela
Colombia
Peru
Brazil
Bolivia
Chile
Argentina
Finland
Sweden
Norway
Russia
United Kingdom
Poland
Ukraine
France
Spain
Italy
Kazakhstan
Mongolia
Turkey
China
Iraq
Iran
Afghanistan
Pakistan
Algeria
Libya
Egypt
Saudi Arabia
India
Thailand
Mali
Niger
Chad
Sudan
Nigeria
Ethiopia
Kenya
DRC
Tanzania
Angola
Namibia
Botswana
Madagascar
South Africa
Indian Ocean
North Pacific Ocean
North Atlantic Ocean
South Pacific Ocean
South Atlantic Ocean
10
14
52


C++ Community Events
General Public
search
April 2018
Today
Apr 2018 - Mar 2019
Week
8 Weeks
Month
Year
List
Number of months
12
All
Calendars
C++ Conferences
C++ Training
C++ User Groups
import
ISO Committee Meetings
Related Conferences
Related Training
Related User Groups
Filter
About
This calendar is for sharing C++ community events. Appropriate events are open to the public and either explicitly C++ or mostly C++.
Event organizers can contact
April
May
June
July
August
September
October
November
C++Now (info@cppnow.org)
Rapperswil Meeting (Peter So
ACCU (julie@arche
Pacific++
CppCon
CppCon (info@cppcon.org)


Get Started!
Tour
Core Guidelines
Super-FAQ
Standardization
About
C++ NORTH
Wiki Home > Conferences Worldwide
Conferences Worldwide
Contents of this section:
Asia / Pacific
Europe
North America
FAQ Asia / Pacific
C++ and System Software Summit - China
CppSiberia - Russia
PacifiC++ - Australia/New Zealand/Asia
FAQ Europe
ACCU - UK
FEATURES
Working Draft of the next standard
github
Current ISO C++ status
Upcoming ISO C++ meetings
Compiler conformance status
NAVIGATION
FAQ Home
FAQ RSS Feed
FAQ Help
SEARCH THIS WIKI
Go
COREHARD


http://cppvap.wikidot.com/


Any questions?