C++语言程序设计

C++语言漫谈

中国科大　　黄章进

主要内容

- C++的前世今生
- 为什么学C++
- 如何学好C++

从B到C

C++的前世今生

- BCPL - Basic Combined Programming Language
  - B语言和C语言的祖先，由剑桥大学的Martin Richards在1964年对CPL语言进行简化得到
- 美国AT&T的贝尔实验室
- B语言
  - 1970年，肯·汤普森（Ken Thompson）在BCPL语言的基础上设计

C++的前世今生

从B到C

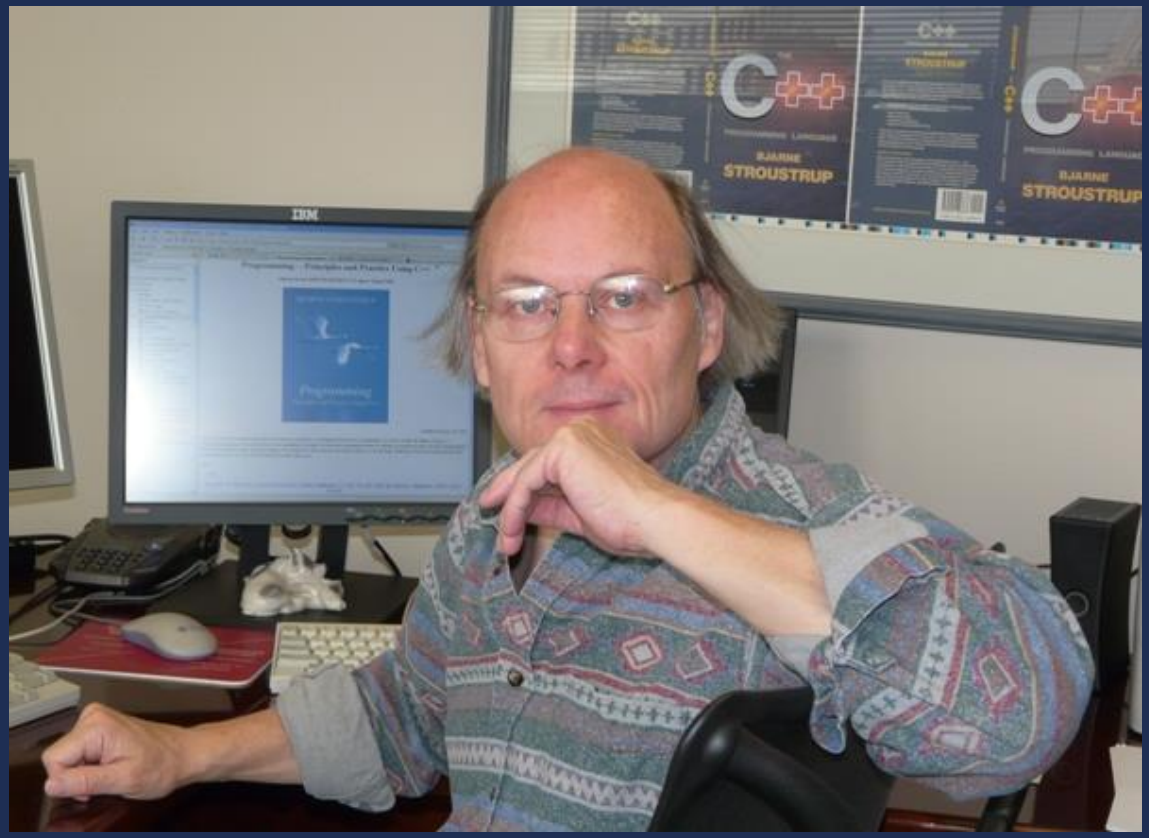
- C语言
  - 1972年，丹尼斯·里奇（Dennis Ritchie）基于B语言设计
  - 1978年，K&R C： Dennis M. Ritchie和Brian W. Kernighan合著《The C Programming Language》
  - C90标准：ISO/IEC 9899-1990
  - C99标准：ISO/IEC9899:1999
  - 2011年12月，ISO正式公布C语言新的国际标准草案：ISO/IEC 9899:2011

从C到C++

C++的前世今生

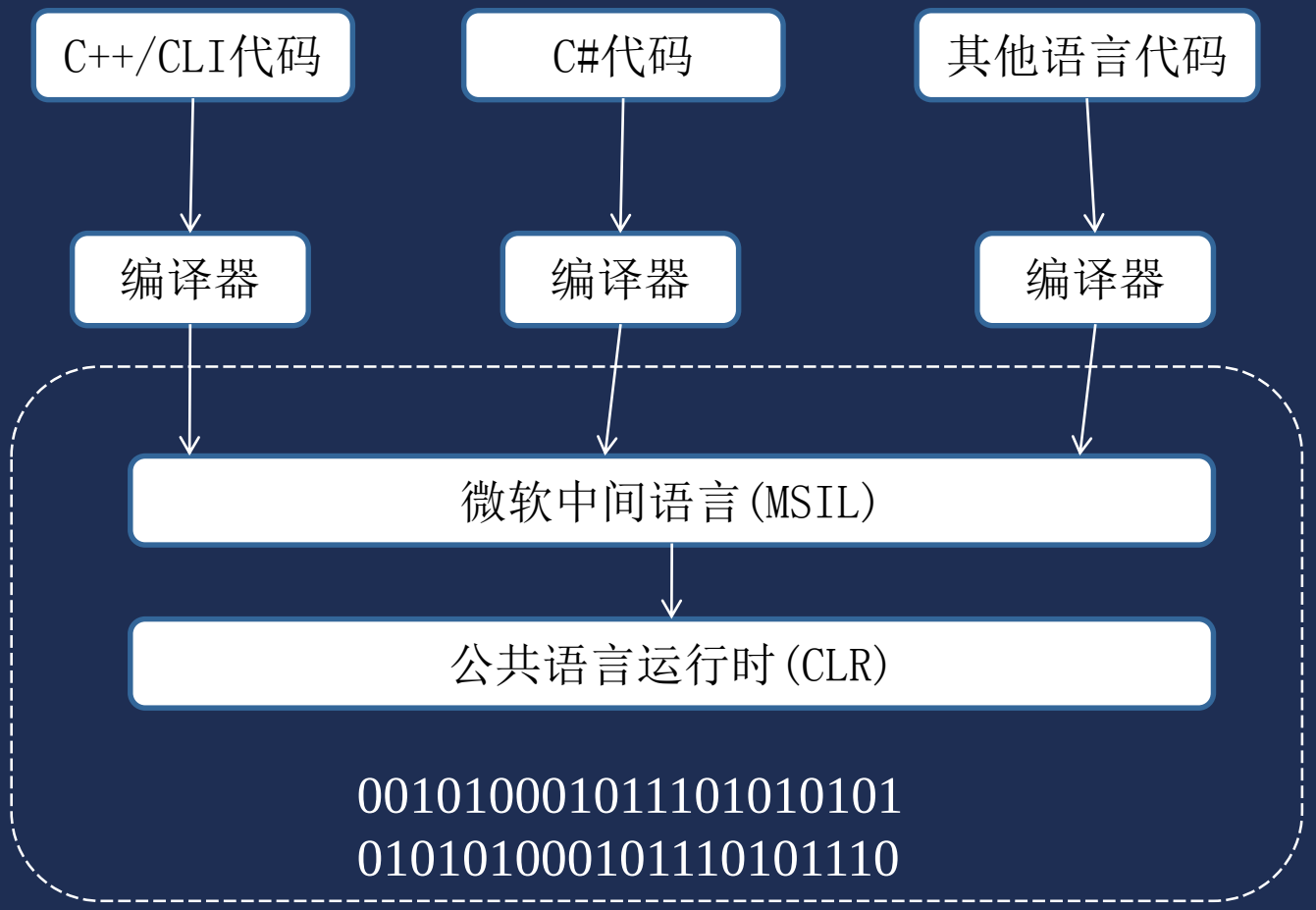
- C with Classes
  - 1979年，本贾尼·斯特劳斯特卢普（Bjarne Stroustrup）博士在C语言中加入类似Simula语言的类机制，
  - 1983投入使用，定名C++
- “拿来主义”
  - 从Simula继承了类的概念
  - 从Algol68继承了操作符重载、引用以及在任何地方声明变量的能力
  - 从BCPL获得”//”注释
  - 从Ada得到了模板、命名空间
  - 从Ada、Clu和ML取来了异常处理等
- C++标准
  - C++98(ISO/IEC 1988-1998)，C++03(ISO/IEC 14882)
  - 2011年9月1日出版发布C++11(ISO/IEC 14882:2011)

Bjarne Stroustrup



.NET Framework的C++/CLI

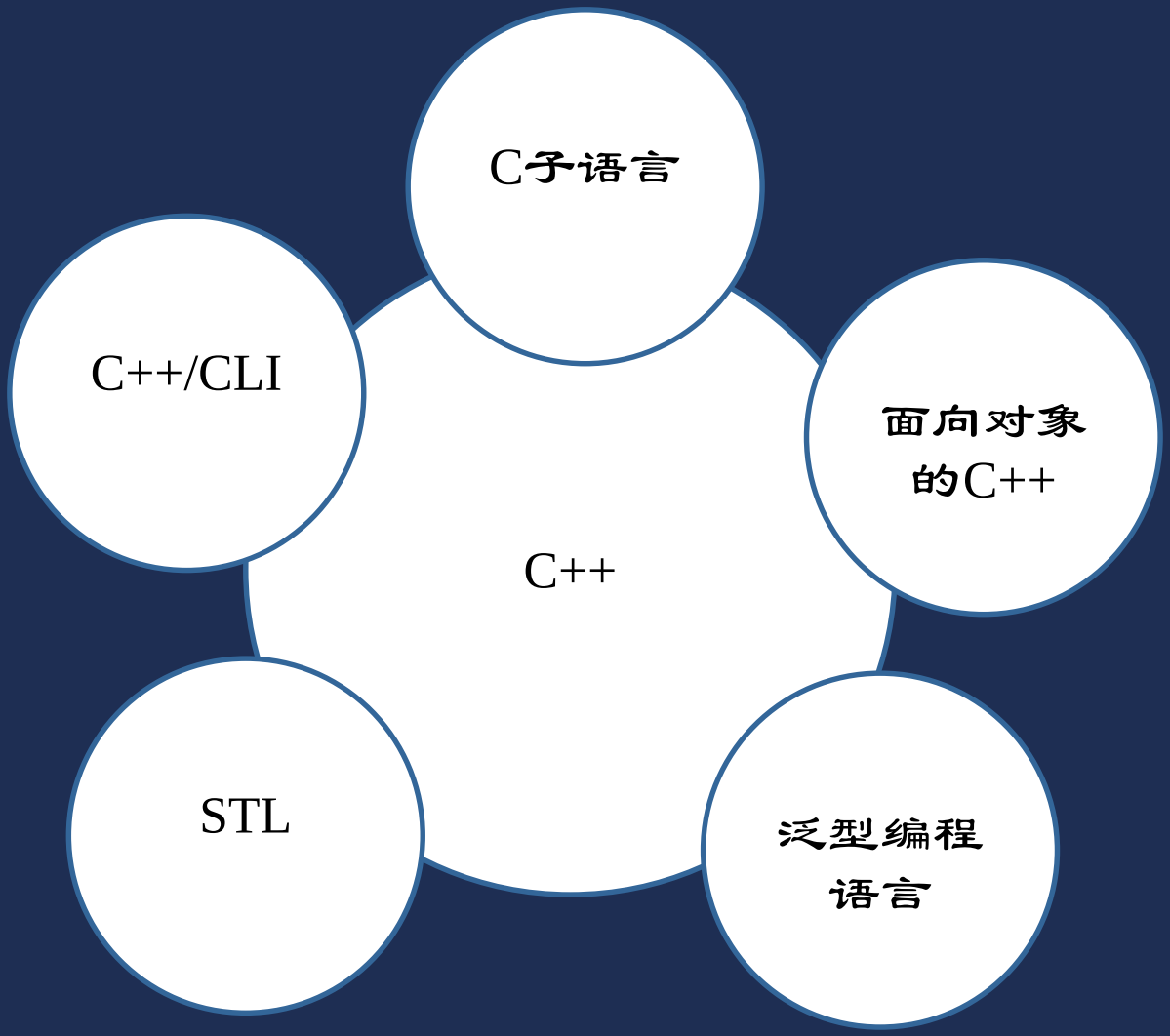
C++的前世今生



公共语言结构(CLI)

C++的五大子语言

C++的前世今生



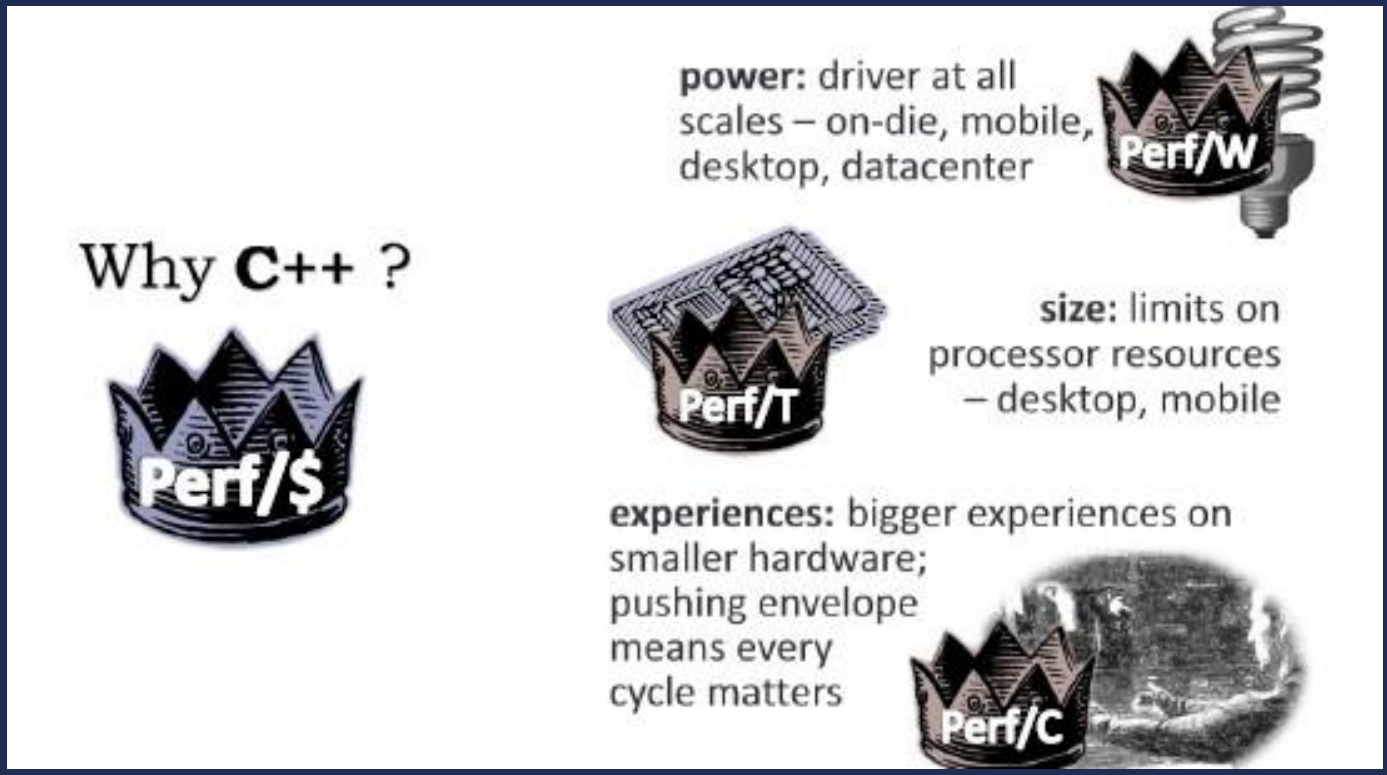
Why C++?

为什么学C++

- 内容来自于C++ and Beyond 2011上的一次公开演讲
- 演讲者：Herb Sutter
  - ISO C++ 委员会的Chair
  - C++/CLI首席架构师
  - Microsoft的软件架构师
  - Exceptional C++ 和 C++ Coding Standards 的作者

性价比

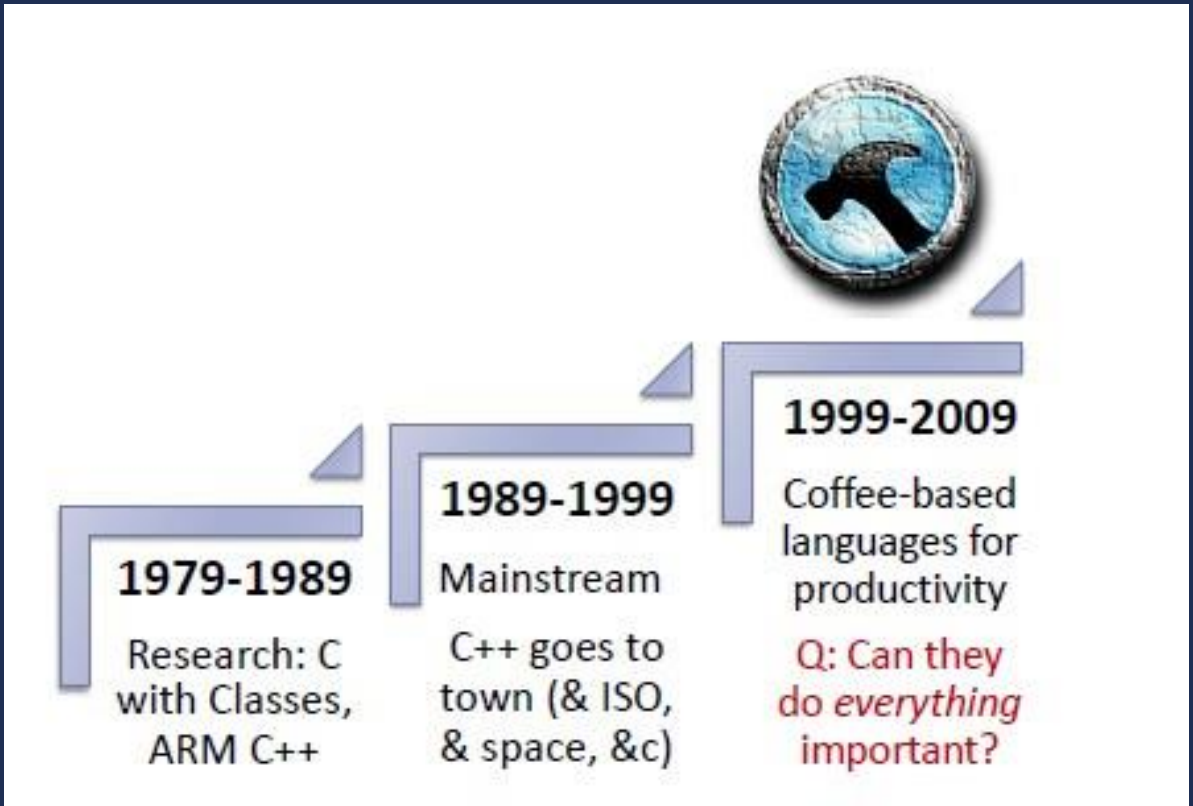
为什么学C++



- 电力、资源和体验

C++进化的三个时期

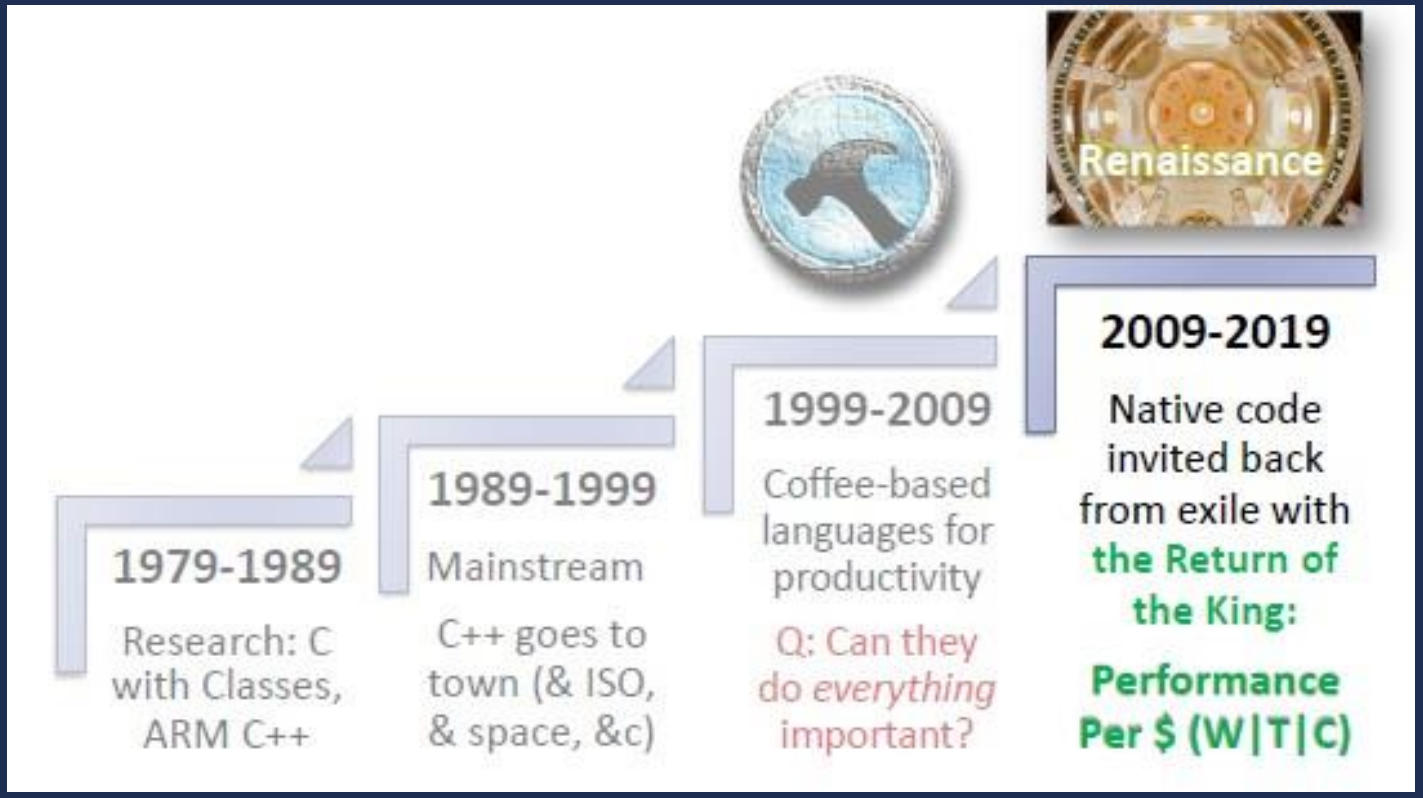
为什么学C++



- 生产力：开发效率

第四个时期

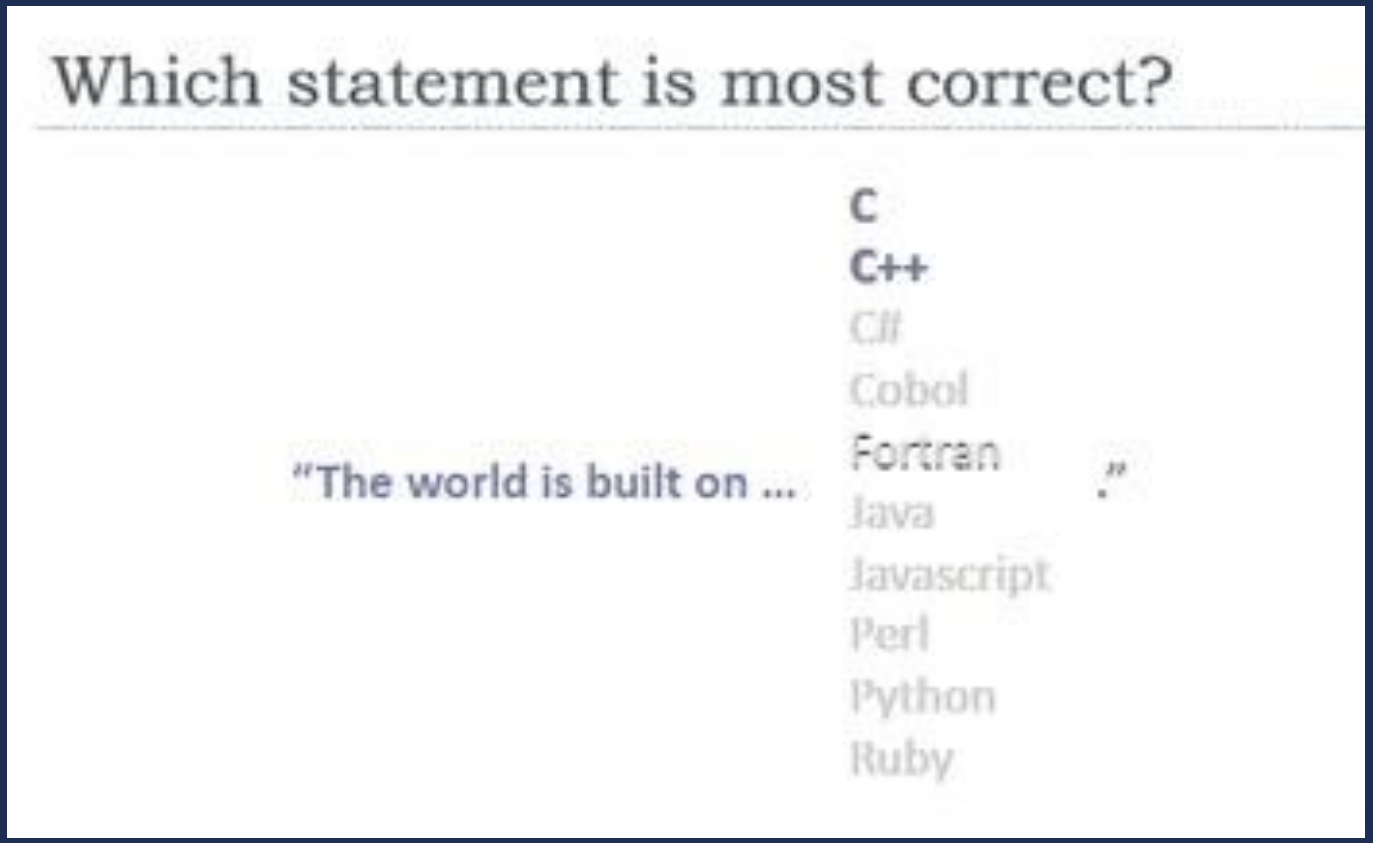
为什么学C++



- 性能

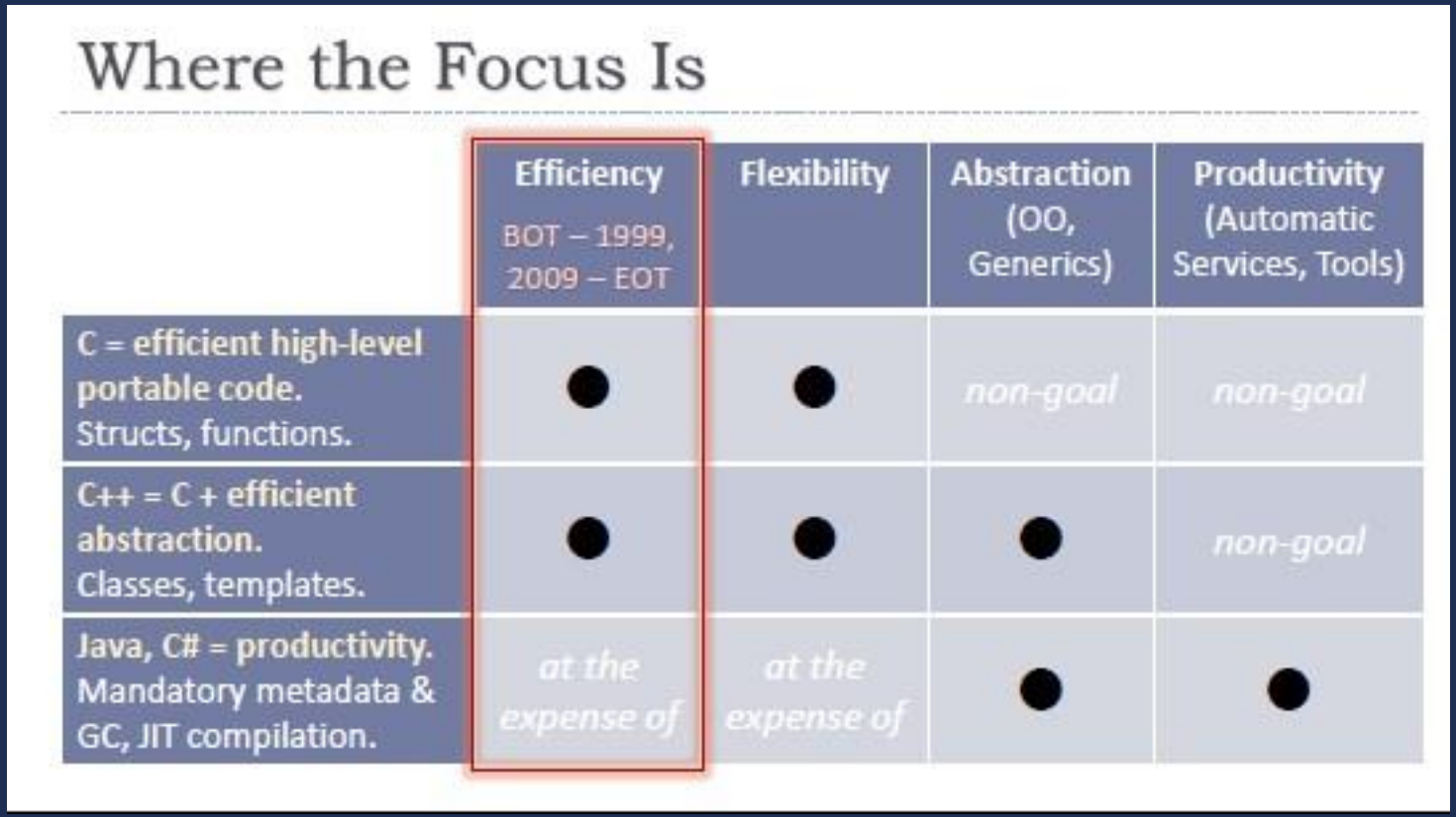
The World is built on

为什么学C++



效率/灵活/抽象/生产力

为什么学C++



Where the Focus Is

	Efficiency BOT – 1999, 2009 – EOT	Flexibility	Abstraction (OO, Generics)	Productivity (Automatic Services, Tools)
C = efficient high-level portable code. Structs, functions.	●	●	*non-goal*	*non-goal*
C++ = C + efficient abstraction. Classes, templates.	●	●	●	*non-goal*
Java, C# = productivity. Mandatory metadata & GC, JIT compilation.	*at the expense of*	*at the expense of*	●	●

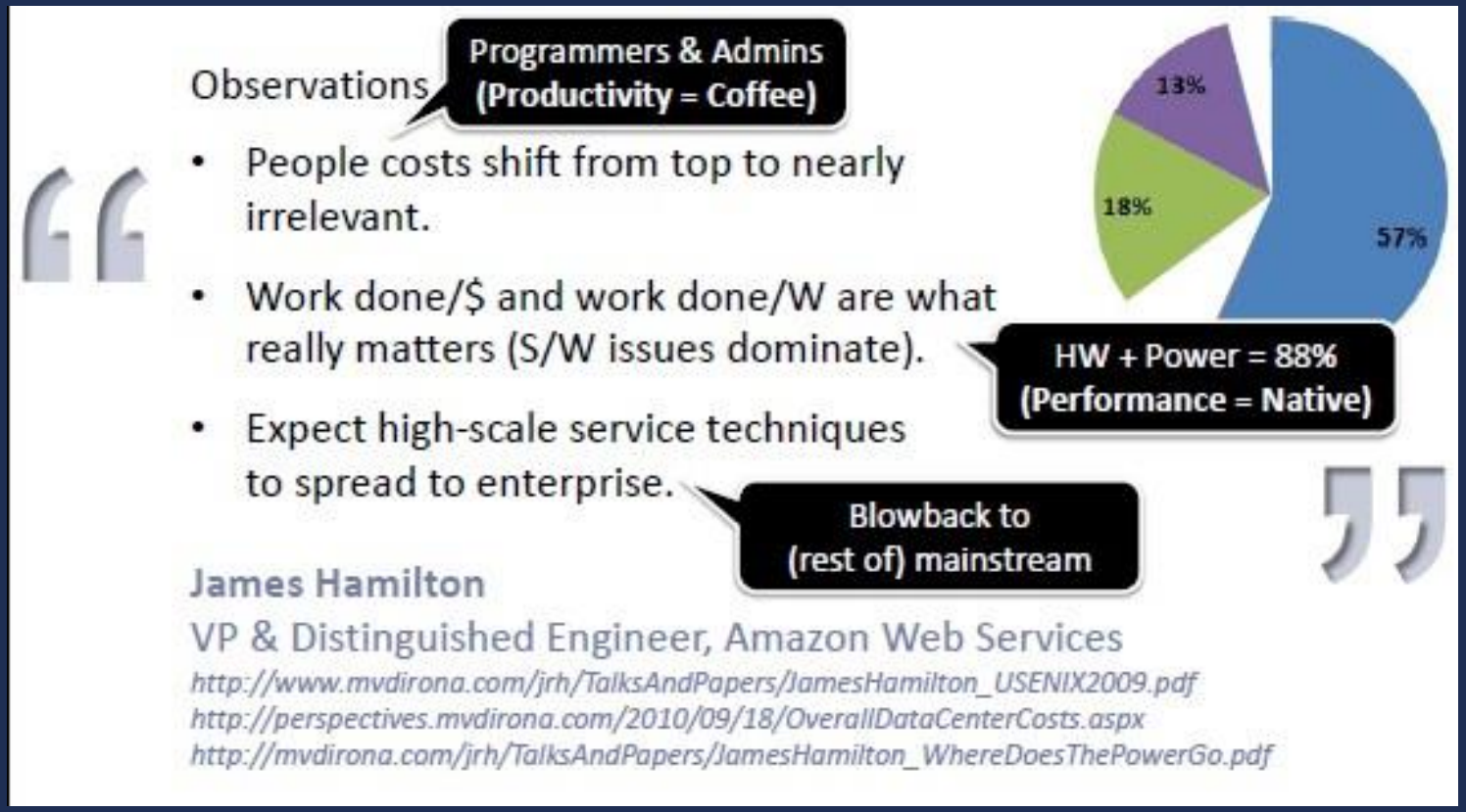
移动设备开发

为什么学C++

Version 1	–	Java	.NET
Version 2+	Objective-C, C & C++ incl. C++ wrappers over Objective-C	Java, C & C++ NDK incl. Java-free C++ apps	?

数据中心

为什么学C++


Observations
Programmers & Admins
(Productivity = Coffee)
People costs shift from top to nearly irrelevant.
Work done/$ and work done/W are what really matters (S/W issues dominate).
Expect high-scale service techniques to spread to enterprise.
13%
18%
57%
HW + Power = 88%
(Performance = Native)
Blowback to
(rest of) mainstream
James Hamilton
VP & Distinguished Engineer, Amazon Web Services
http://www.mvdirona.com/jrh/TalksAndPapers/JamesHamilton_USENIX2009.pdf
http://perspectives.mvdirona.com/2010/09/18/OverallDataCenterCosts.aspx
http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_WhereDoesThePowerGo.pdf

减轻全球变暖

为什么学C++

减轻全球变暖

为什么学C++

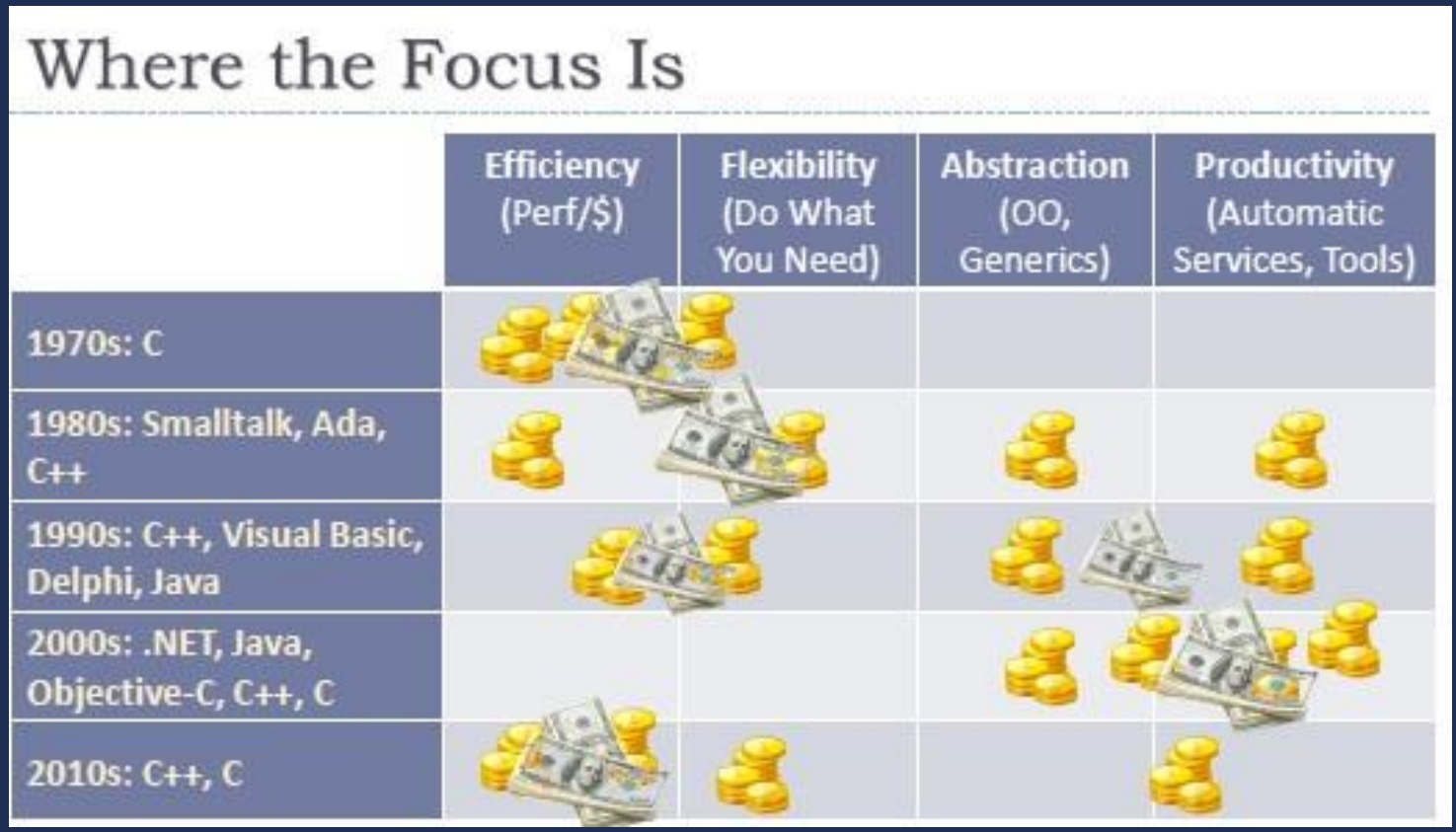
My contribution to the fight against global warming is C++'s efficiency: Just think if Google had to have twice as many server farms! Each uses as much energy as a small town. And it's not just a factor of two…

Efficiency is not just running fast or running bigger programs, it's also running using less resources.

Bjarne Stroustrup, June 2011

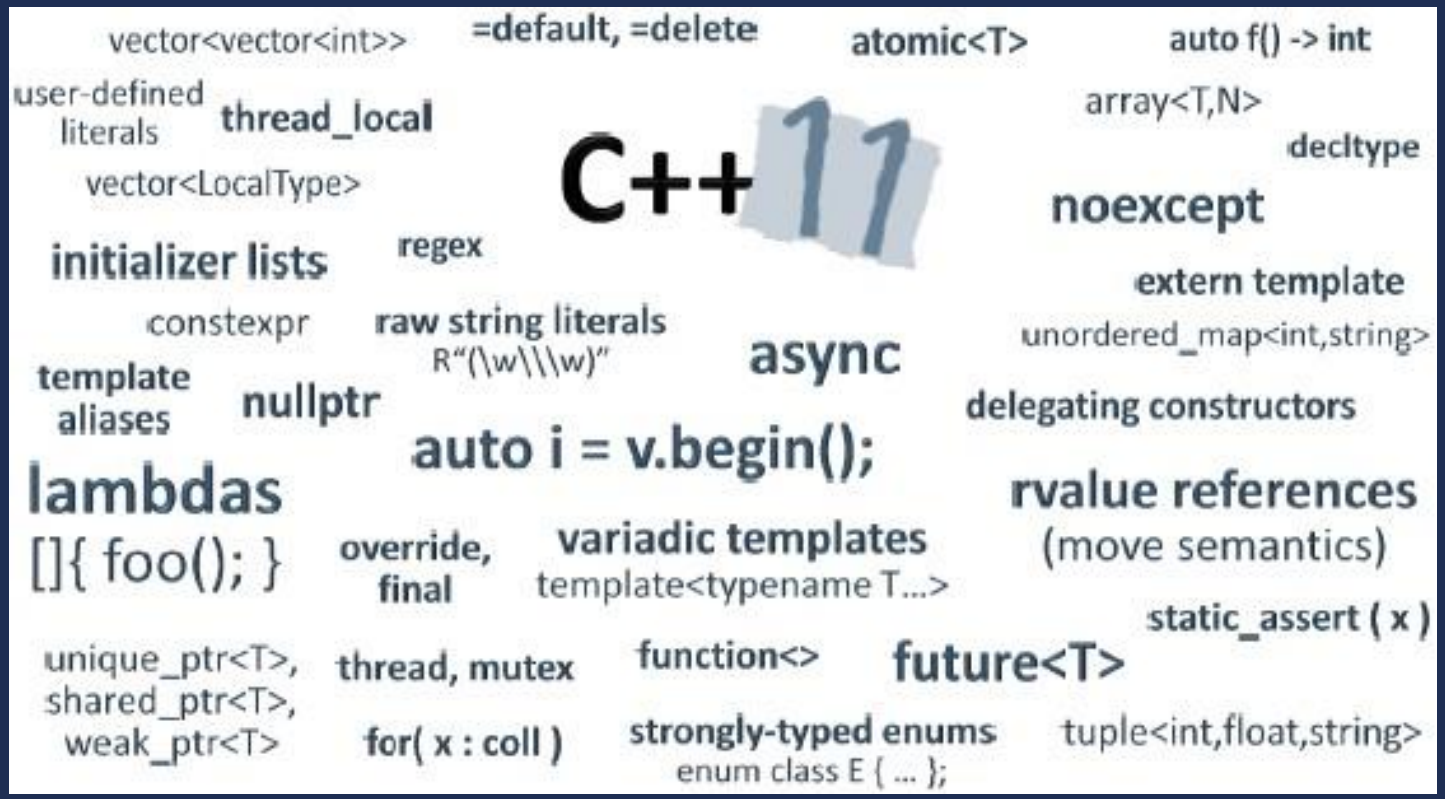
钱投在了哪里

为什么学C++



C++11

为什么学C++



- 性能和抽象的平衡

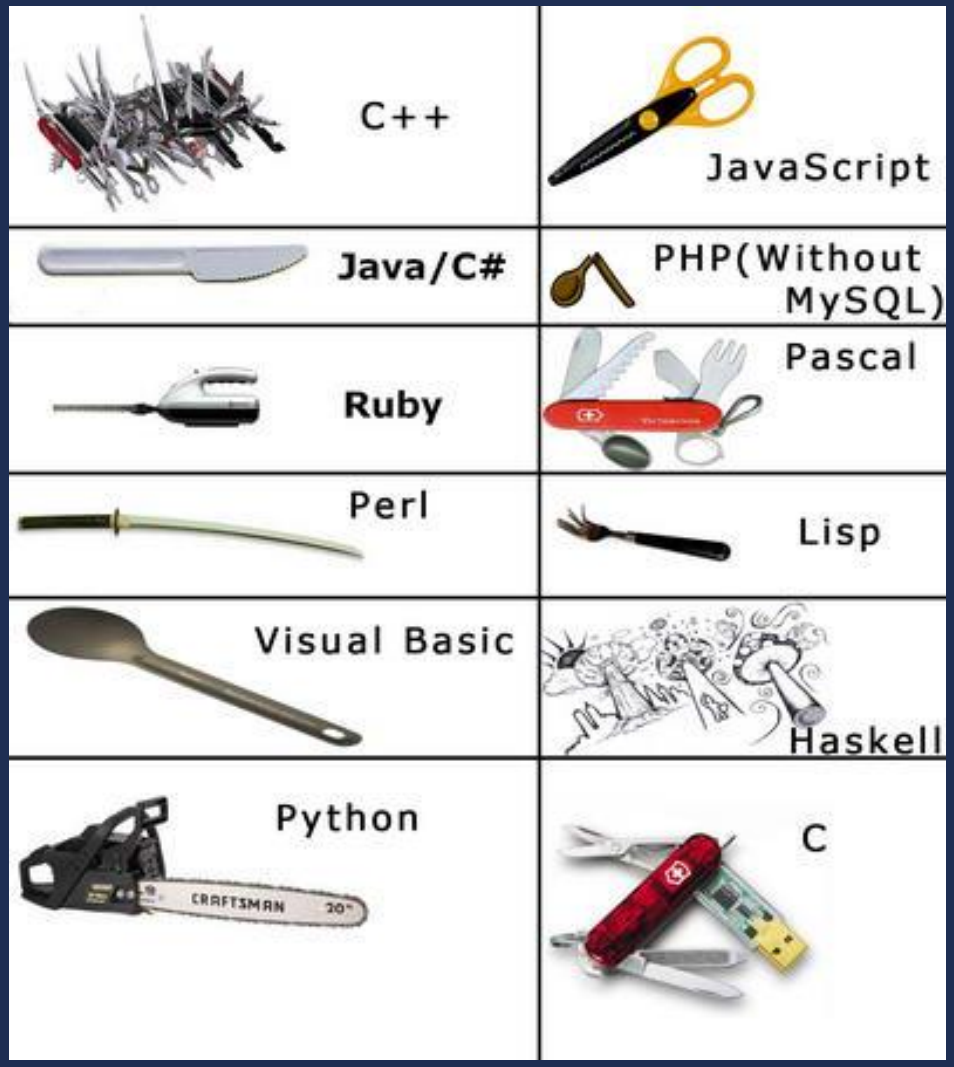
文无第一

为什么学C++

- 数学
  - Mathematica, Maple
- 工程
  - Matlab
- 思想和方法

如果编程语言是一种刀

为什么学C++



多读多写

如何学好C++

- 听说读写
- 多读
  - 阅读优秀源代码，学习设计思想、编程风格
  - 熟读唐诗三百首，不会作诗也会吟
- 多写
  - 北冥神功，化为己用

善用网络

如何学好C++

- Google
  - 教程
  - 文档
  - 源代码
- 维基百科