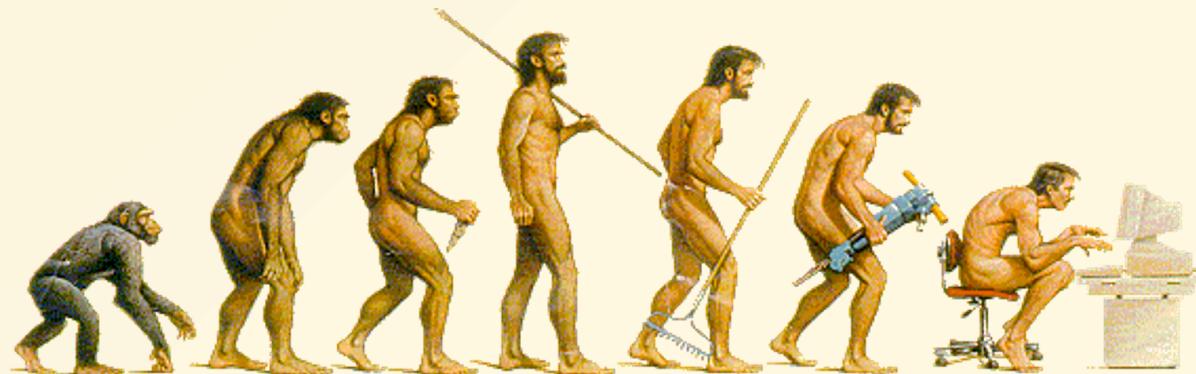


# Object-Oriented Programming with C++

Xiang Chen @ CAD&CG Lab

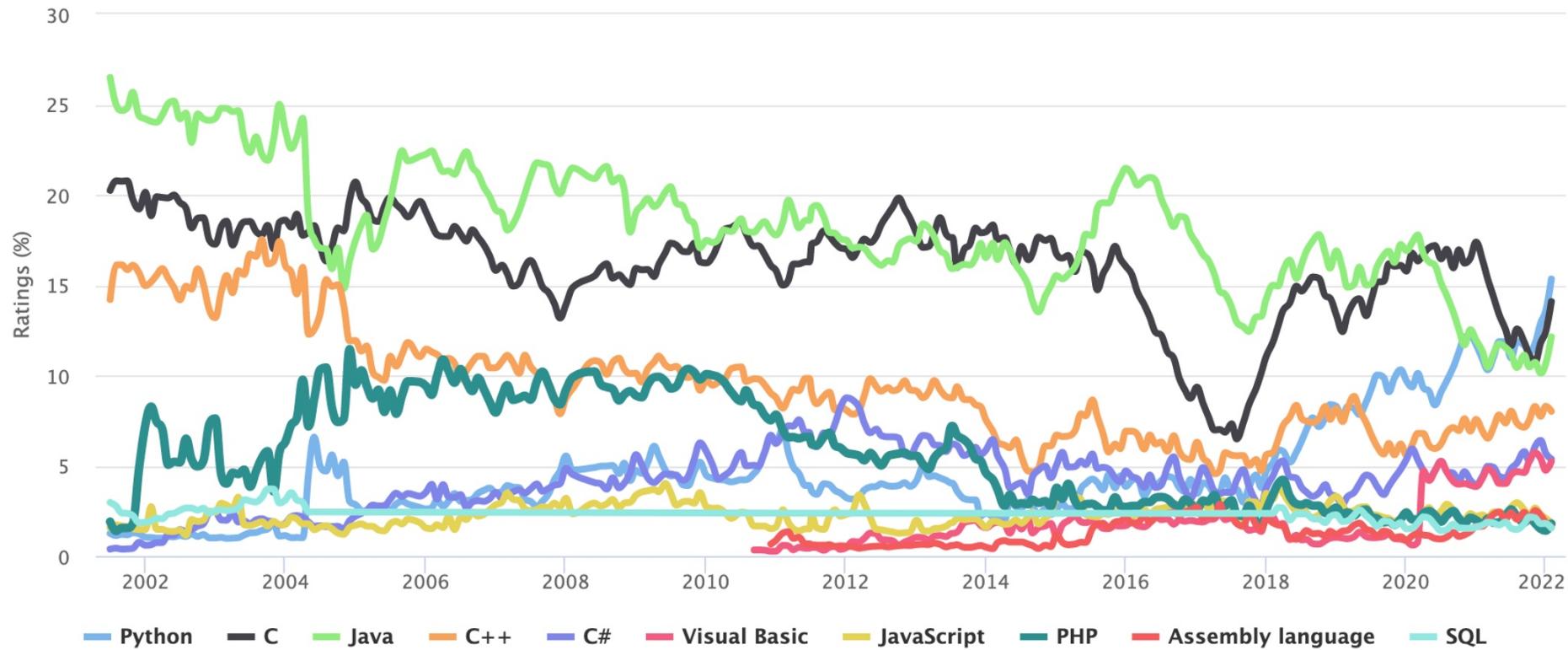
<https://xchen-cs.github.io/>



# Course contents

- Introduction to *Object-Oriented Programming* (OOP)
- ... with a solid software engineering foundation ...
- ... aimed at producing and maintaining large, high-quality, software systems.

# Ranked 4th on TIOBE index



# C++ applications



Adobe  
Illustrator

Google



Windows



Mac OS



mozilla  
Firefox®



MySQL®



Office

# C++ applications



# C++ applications

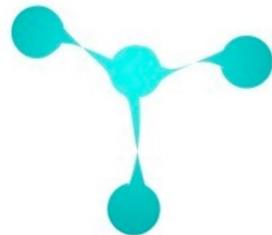
video: ue5 tech demo...

# C++ applications



Caffe

 Caffe2



PYTORCH

 Chainer

 Keras

 TensorFlow

theano

$\partial$ y/net

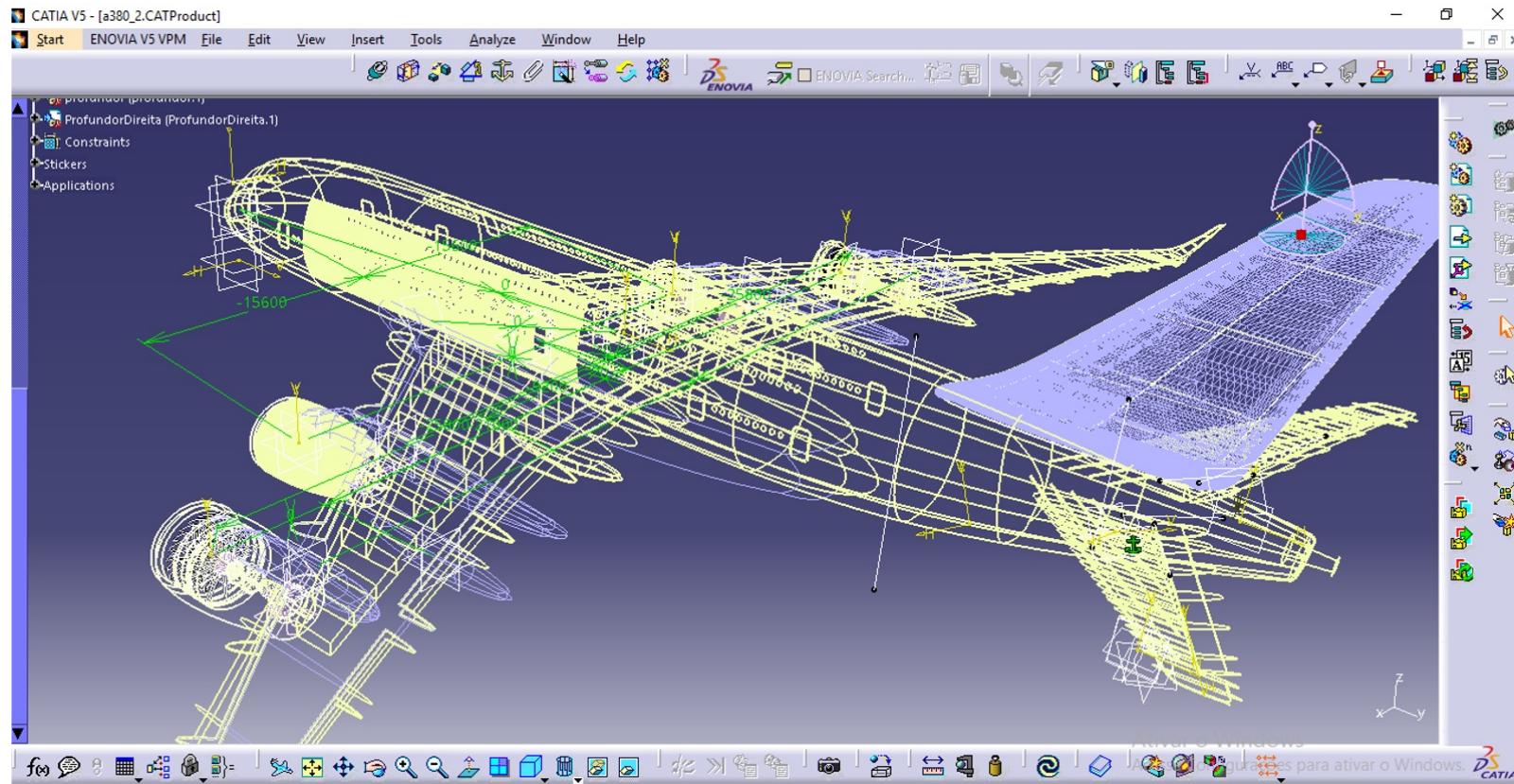
 mxnet

 GLUON

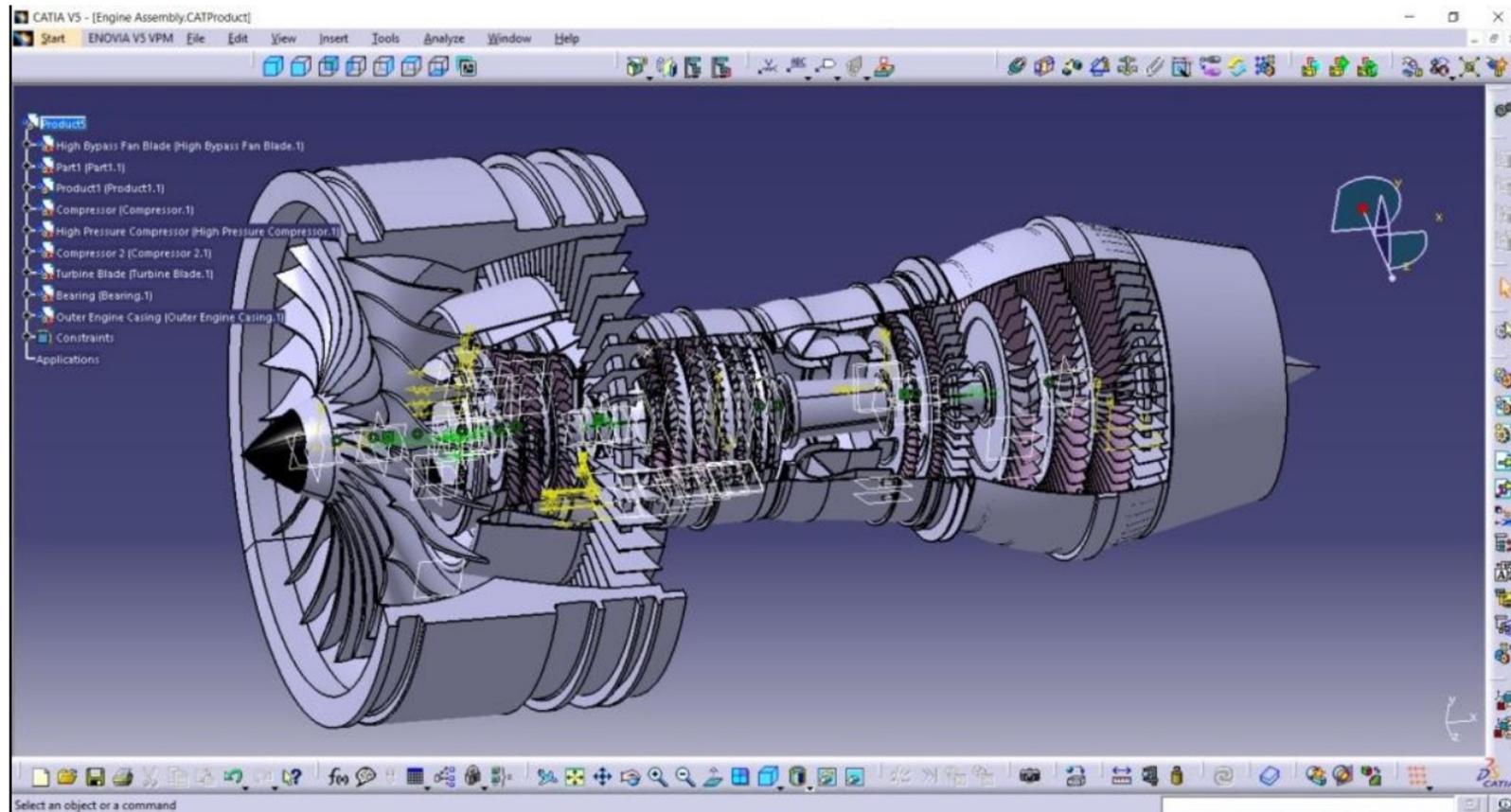
# C++ applications 「卡脖子问题」



# C++ applications 「卡脖子问题」



# C++ applications 「卡脖子问题」



# Buzzwords

responsibility-driven design

inheritance

encapsulation

iterators

overriding

coupling

cohesion

template

interface

collection classes

mutator methods

polymorphic method calls

# **Textbooks**

# Textbooks

The best one I know 😊

***" Teach Yourself C++ in 21 Days! "***

### 第1-10天

学习变量，常量，数组，字符串，表达式，语句，函数……



### 第11-21天

学习程序流程，指针，引用，类，对象，接口，多态……



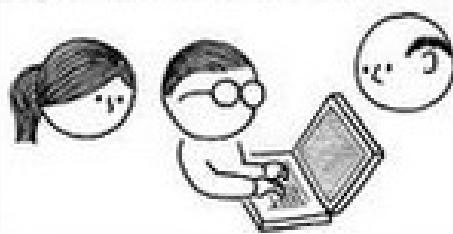
### 第22-697天（两年）

进行大量的“休闲娱乐”方式的编程，并在Hack代码中找到乐趣，从错误中学习总结。



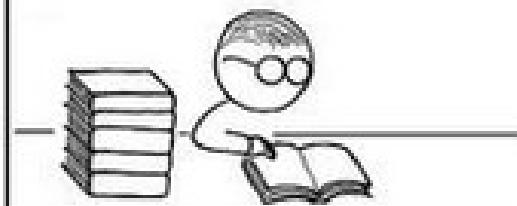
### 第698-3648天（八年）

与其它程序员互相影响，并一同开发项目，并向他们学习。



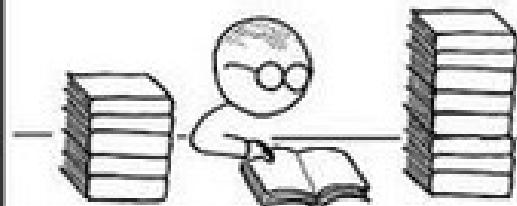
### 第3649-7781天（十年）

学习高等理论物理，使用公式证明量子物理理论。



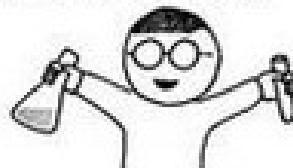
### 第7782-14611天（二十年）

学习生物化学，分子生物学，以及遗传学。



### 第14611天 (自学习编程来的四十年)

使用在生物学方面的知识造一个“返老还童”药剂。



### 第14611天

使用在物理学上的知识造一个通量电容器，把自己传送到学编程的第21天。



### 第21天

把那时的自己给替换了。



www.csdn.net

这是迄今为止  
我所知道的最容易的  
《Teach Yourself C++ in 21 Days》

# Textbooks

- Thinking in C++
- C++ Primer
- The C++ Programming Language
- A Tour of C++
- Essential C++
- Effective C++
- Inside the C++ Object Model
- ...



# Online Resources for C++

- <https://cppreference.com>
- <https://isocpp.org/>
- <https://www.youtube.com/user/CppCon>

# Assessment

- Lab (50%, including the in-class quiz)
  - Announced in week 1, 3, 5, 7, 9, 11, and 13
  - Submit on-line (manually judged by TA)
  - The deadline is truly DEAD
  - Per day delay cost: 10% of the full score
- Final exam (50%, on PTA)

# Resources

- Courseware (学在浙大)
- Assignments (PTA)

**PTA绑定码: 915333**

# Academic honesty

- Students are fully responsible for their actions.
- During homework, students can help each other through hints and explanations.
- Copying code from anybody else is strictly forbidden.
- <https://conduct.berkeley.edu/integrity/>



uc berkeley Academic honesty



All

Images

News

Shopping

Videos

More

Tools

About 494,000 results (0.46 seconds)

Ad · <https://www.wholeren.com/> : (412) 756-3137

## 厚仁教育-北美学术紧急应对中心 - 美国考试作弊被抓/作业抄袭...

GPA低/考试作弊/代写被抓,被学校开除/退学,帮您申诉/应对听证会/紧急转学/恢复身份. 考试作弊/GPA低/抄袭剽窃/成绩造假,7000+成功案例,专业团队帮你申诉,摆脱困境. 100%成功率.

### People also search for

<a href="#">uc berkeley cheating</a>	<a href="#">academic misconduct</a>
<a href="#">uc berkeley cheating reddit</a>	<a href="#">academic interference</a>
<a href="#">uc berkeley rules</a>	<a href="#">academic conduct</a>
<a href="#">uc berkeley cheating policy</a>	<a href="#">student conduct berkeley</a>



<https://conduct.berkeley.edu> › integrity : :

## Academic Integrity - Center for Student Conduct

The high **academic** standard at the **University of California, Berkeley**, is reflected in each degree that is awarded. As a result, every student is expected to ...

### People also search for

<a href="#">uc berkeley cheating</a>	<a href="#">uc berkeley cheating policy</a>
<a href="#">uc berkeley cheating reddit</a>	<a href="#">academic misconduct</a>
<a href="#">uc berkeley rules</a>	<a href="#">academic interference</a>



# E-mail rule

- Contact info:
  - [ TA 陆子仪 ] [ziyilu@zju.edu.cn](mailto:ziyilu@zju.edu.cn)
  - [ me ] [xchenchs@zju.edu.cn](mailto:xchenchs@zju.edu.cn)
- Title starts with **[OOP]**
- State your **name** and **id** in the text

# Introduction to C++

The trip begins...



Let's take off !

# The first C++ program

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, World! I am " << 18
        << " Today!" << endl;

    return 0;
}
```

# Development environment

- Windows:
  - visual studio community
- Linux, macOS:
  - g++, clang
  - visual studio code
- Any compiler that supports C++ standard well...

# Read input

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Enter a decimal number: ";

    int number;
    cin >> number;

    cout << "The number you entered is " << number
        << "." << endl;

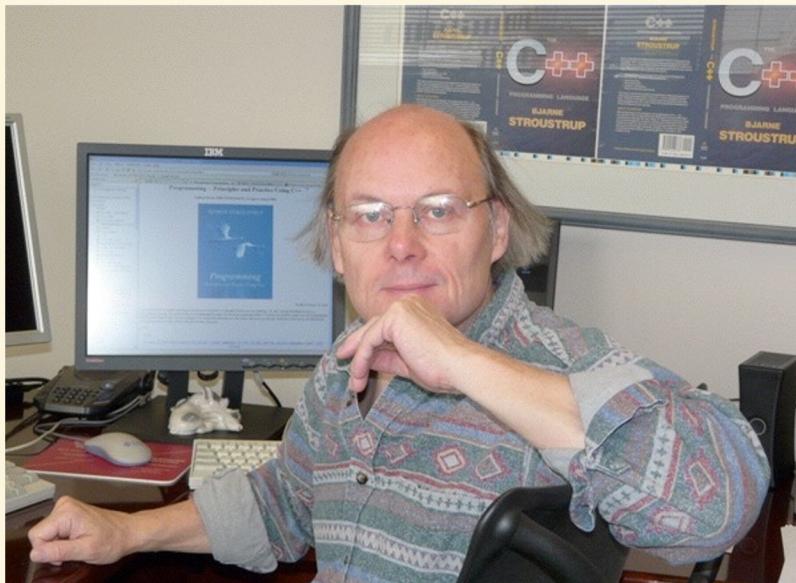
    return 0;
}
```

# The C language

- Strengths
  - Efficient programs
  - Direct access to machine (suitable for OS)
  - Flexible
- Weakness
  - Insufficient type checking
  - Unsuitable for high-level applications
  - No direct support for OOP

# Bjarne Stroustrup

- C++ was first designed and implemented by Bjarne Stroustrup, AT&T, early 1980's
- <http://www.stroustrup.com/>



Bjarne Stroustrup  
visited Zhejiang  
University in  
October 2002.







Bjarne Stroustrup visited zju in 2005



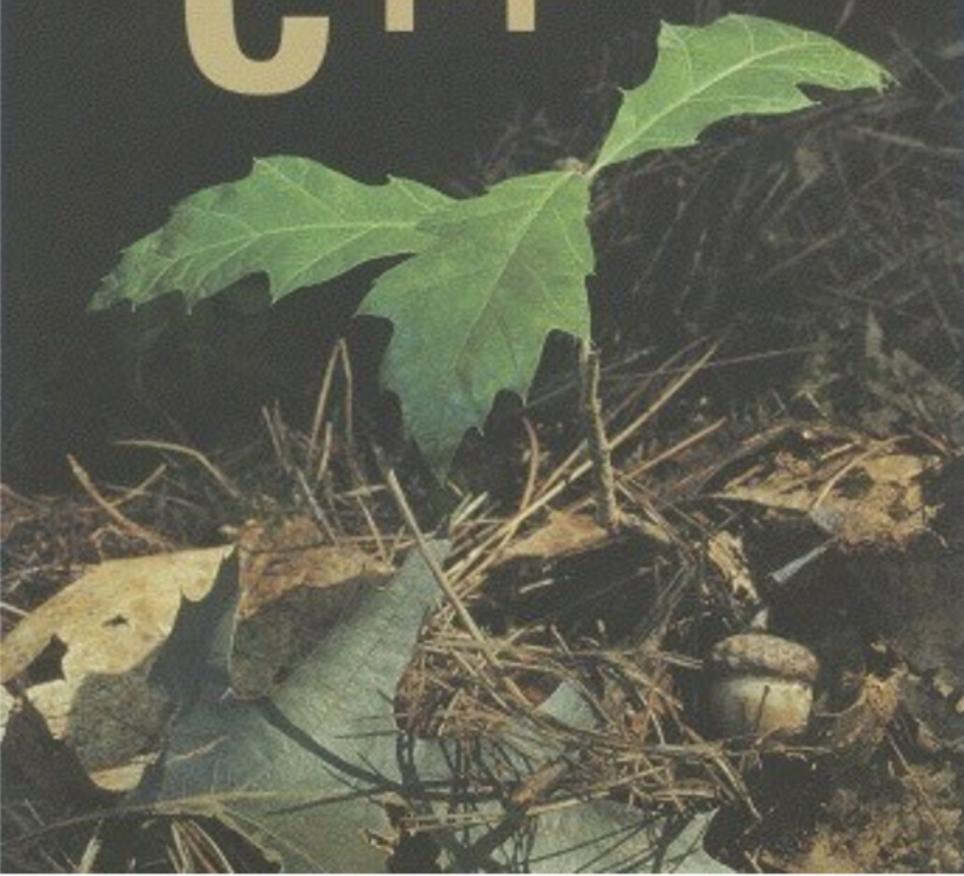
C++ 之父：爱吃辣子鸡，C++20 会非常出色

<https://www.infoq.cn/article/rv3SX2V8rtRaJj9B17xZ>



BJARNE STROUSTRUP

The Design and Evolution of  
**C++**



# Brief history of C++

- 1978: BS at Cambridge, UK. Simulation program in Simula:
  - Supports classes, inheritance, and type check
  - Poor performance
  - <https://en.wikipedia.org/wiki/Simula>

# Brief history of C++

- 1979: BS at AT&T Labs, Cpre, C w/classes
- 1980: Most C++ features but virtual functions
- 1983: C++ w/virtual functions, named C++ by Rick Mascitti
- 1985: "The C++ Programming Language"
- 1990: ANSI C++ Committee ISO/ANSI
- 1998: C++ Standard, ISO/IEC 14882

<http://www.open-std.org/jtc1/sc22/wg21/>

# Brief history of C++

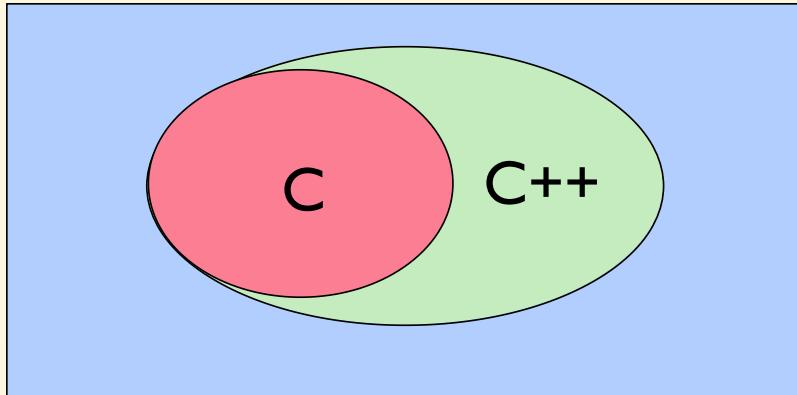
- C++03, C++11, C++14, C++17, C++20
- To be continued ...

# Goal for C++

To combine:

- Flexibility and efficiency of C
- Support for OOP (as Simula and SmallTalk)

# C and C++



- C++ builds on C
- Knowledge of C helps you understand C++
- C++ supports more styles of programming
- C++ provides more features

# C++ improvements

- Data abstraction
- Access control
- Initialization & cleanup
- Function overloading
- Streams for I/O
- Constants (C99)
- Name control
- Inline functions(C99)
- References
- Operator overloading
- Memory management
- Support for OOP
- Templates
- Exception handling
- Extensive libraries
- STL

# C++

- C++ can be viewed as a "better" C
  - C++ → C=C+I
- But C++ is not C, we should
  - Learn C++ as a brand new language
- A multi-paradigm language, which supports
  - Procedure-oriented programming
  - Object-oriented programming
  - Generic programming

# Criticism

- C++

<https://en.wikipedia.org/wiki/C%2B%2B#Criticism>

- OOP

[https://en.wikipedia.org/wiki/Object-oriented\\_programming#Criticism](https://en.wikipedia.org/wiki/Object-oriented_programming#Criticism)