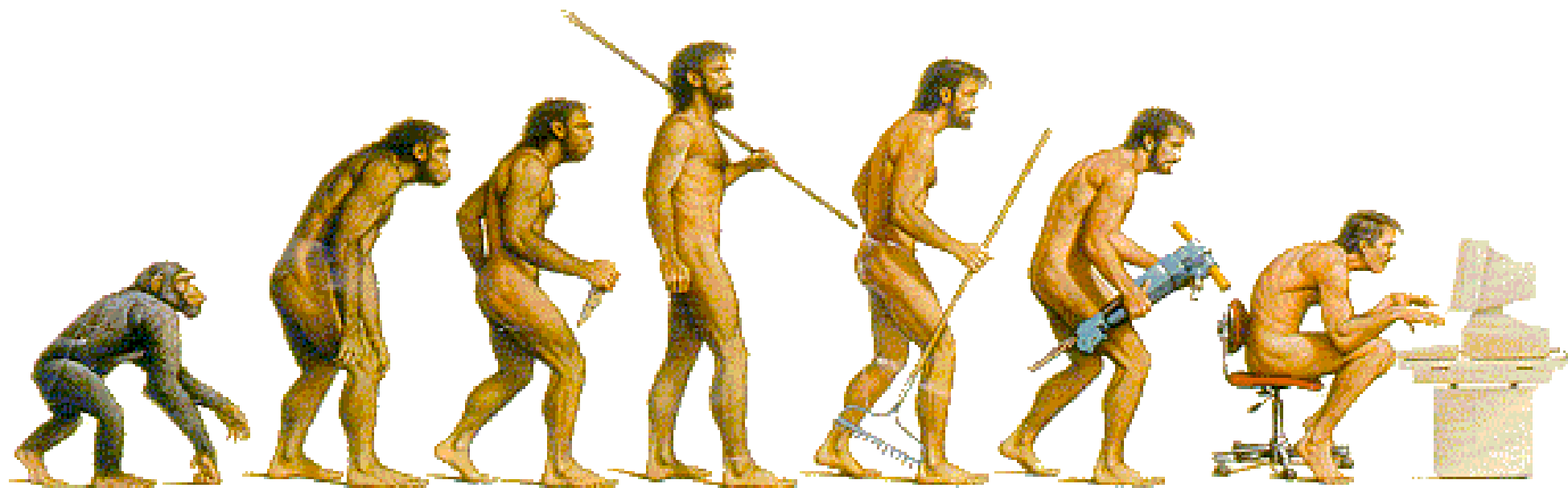


# Object-Oriented Programming

with C++

Zhaopeng Cui

<http://www.cad.zju.edu.cn/home/zhpcui/>



# Course Contents

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

# Object-oriented Programming

computers are not so much machines as they are mind amplification tools and a different kind of expressive medium. As a result, the tools are beginning to look less like machines and more like parts of our minds, and also like other expressive mediums such as writing, painting, sculpture, animation, and filmmaking. Object-oriented programming is part of this movement toward using the computer as an expressive medium.

# Buzzwords

responsibility-driven design

Encapsulation (封装)

Inheritance(继承)

iterators

overriding

coupling

cohesion

interface











template

collection classes

mutator methods

Polymorphism(多态)

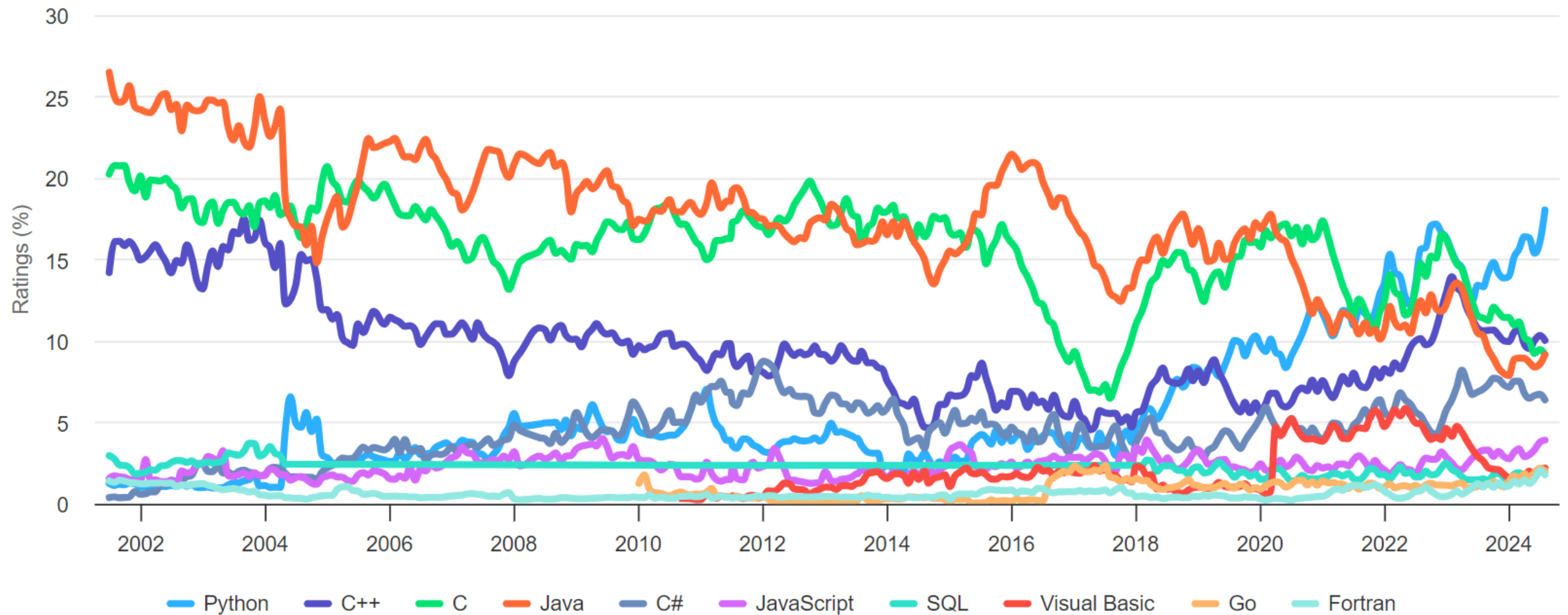
# Ranked 2<sup>nd</sup> on TIOBE Index

Aug 2024	Aug 2023	Change	Programming Language		Ratings	Change
1	1			Python	18.04%	+4.71%
2	3	▲		C++	10.04%	-0.59%
3	2	▼		C	9.17%	-2.24%
4	4			Java	9.16%	-1.16%
5	5			C#	6.39%	-0.65%
6	6			JavaScript	3.91%	+0.62%
7	8	▲		SQL	2.21%	+0.68%
8	7	▼		Visual Basic	2.18%	-0.45%
9	12	▲		Go	2.03%	+0.87%
10	14	▲▲		Fortran	1.79%	+0.75%

# Ranked 2<sup>nd</sup> on TIOBE Index

## TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



# 其他语言？

- 现代的编程语言在语法上的差异很小
  - 几乎都是C-like语言
- 语言的能力/适用领域主要由
  - 库 and
  - 传统所决定

# C++ Applications



Adobe  
Illustrator

Google



chrome



Windows



Mac OS



mozilla

Firefox®



MySQL®



Office



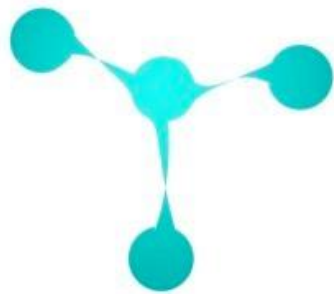
# C++ Applications



# C++ Applications



Caffe



PYTORCH



$\partial y / \text{net}$

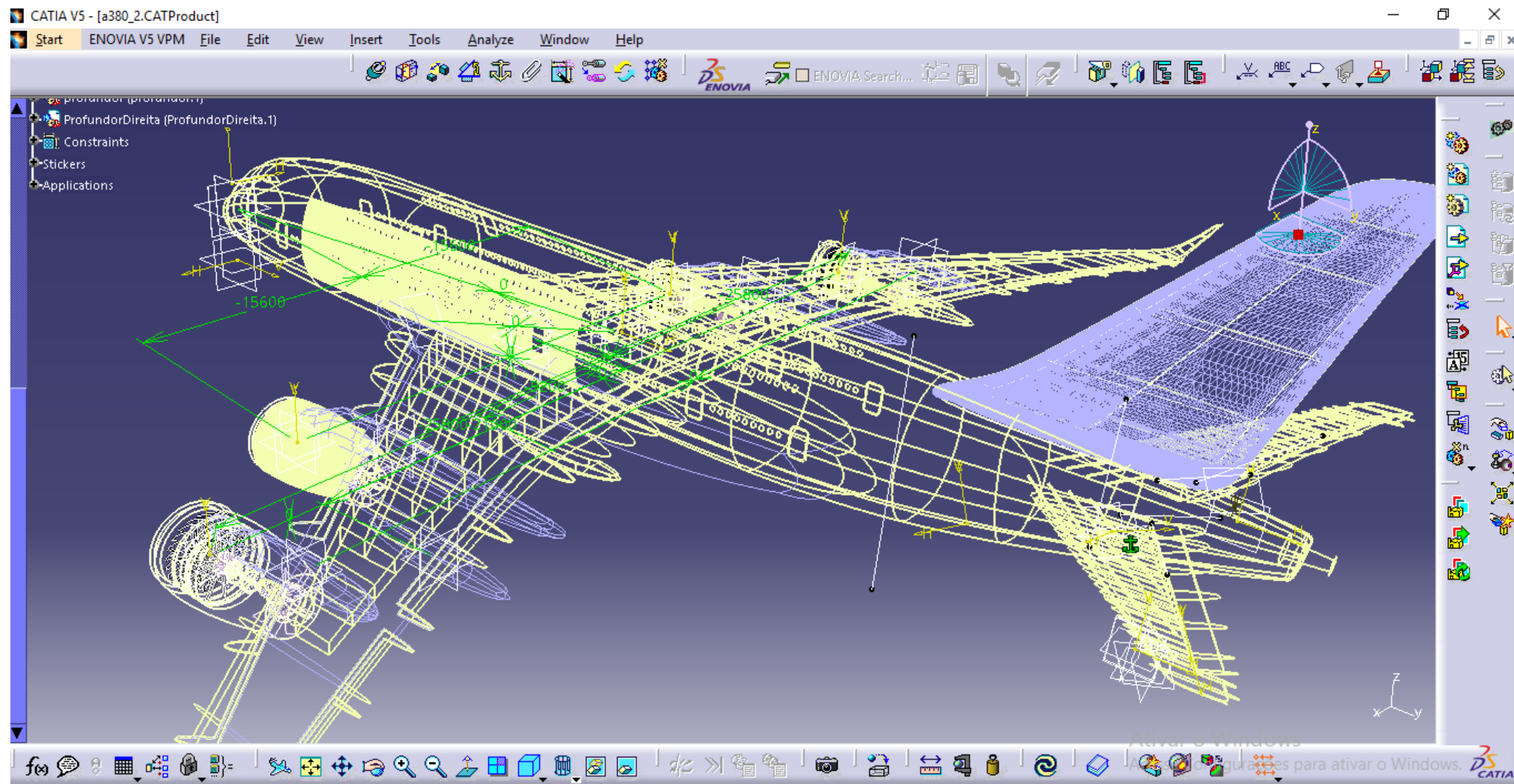


# C++ Applications 「卡脖子问题」

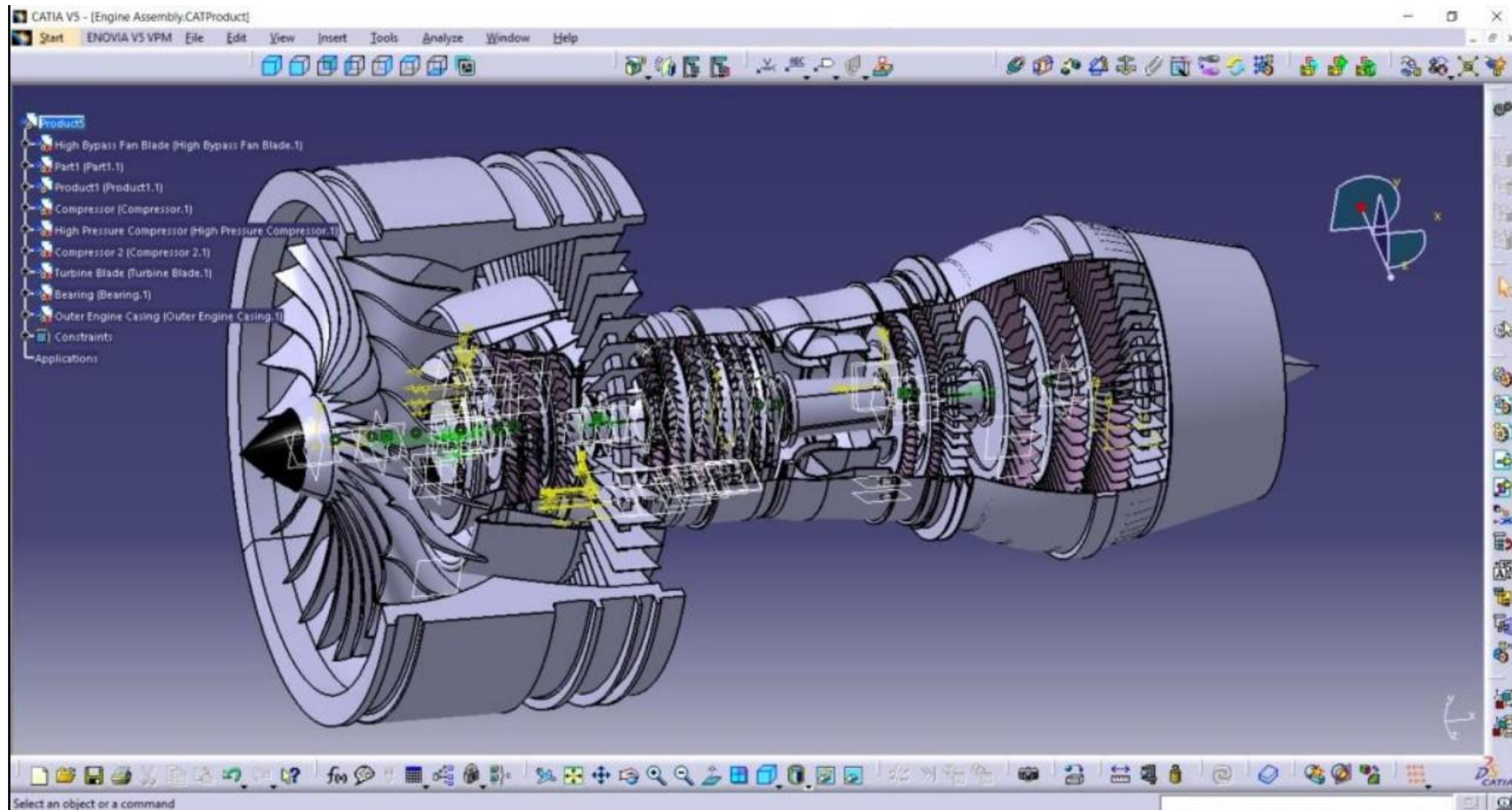




# C++ Applications 「卡脖子问题」



# C++ Applications 「卡脖子问题」





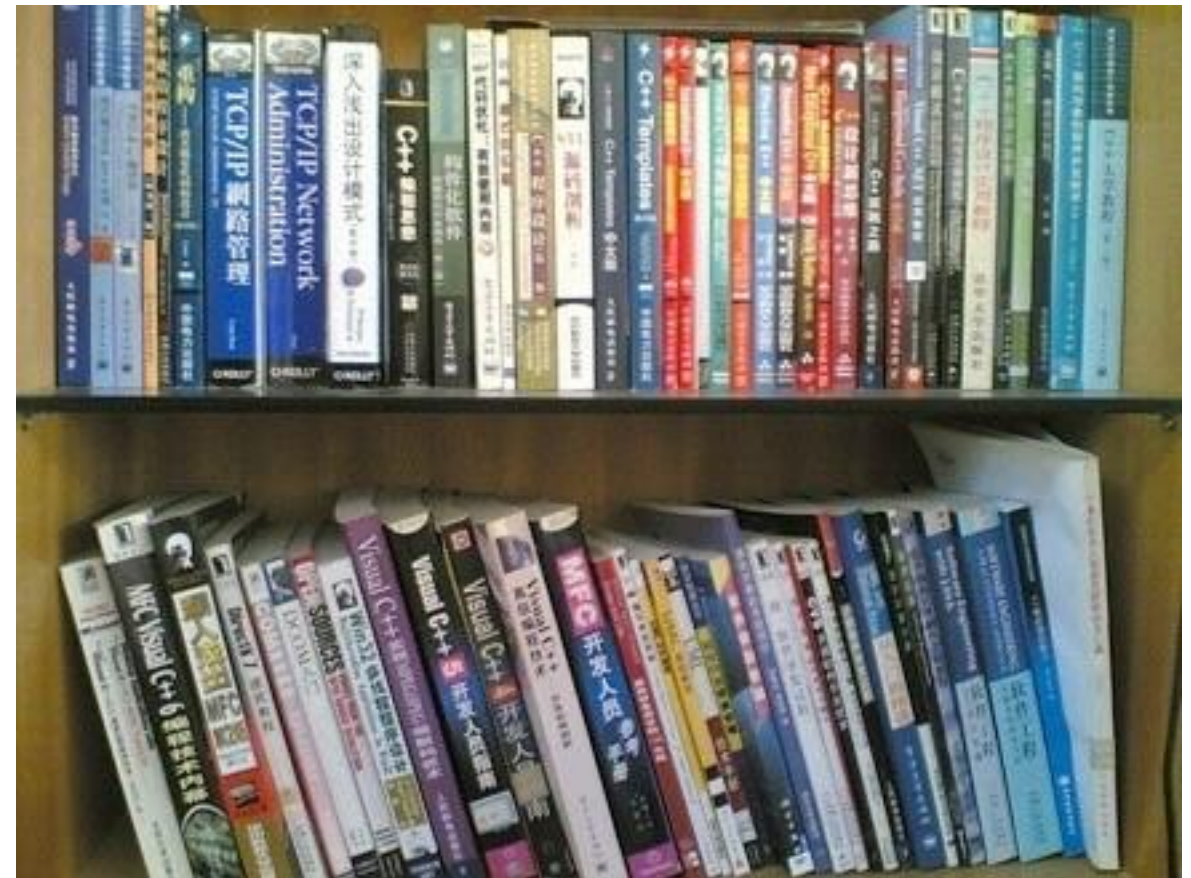
# Textbooks





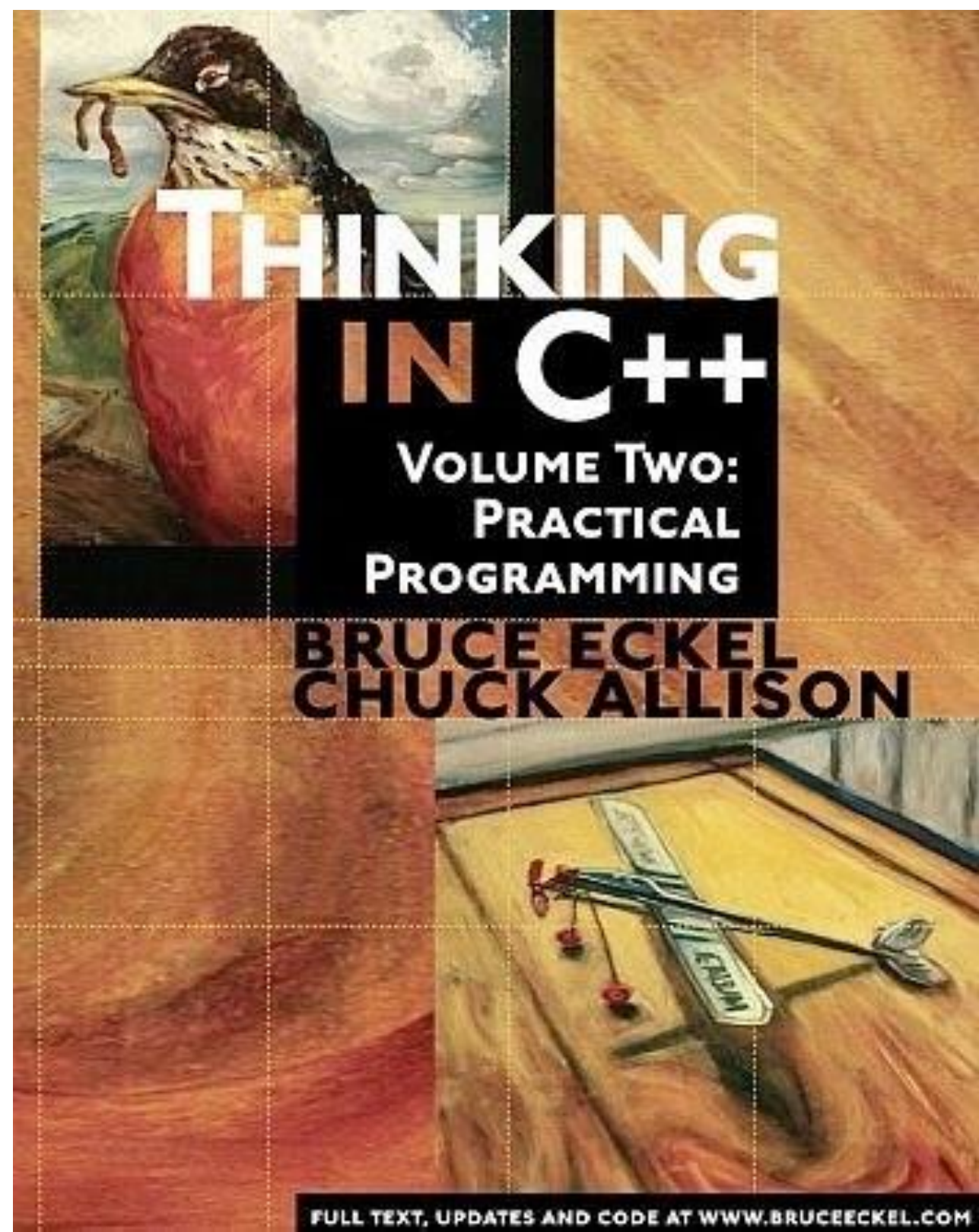
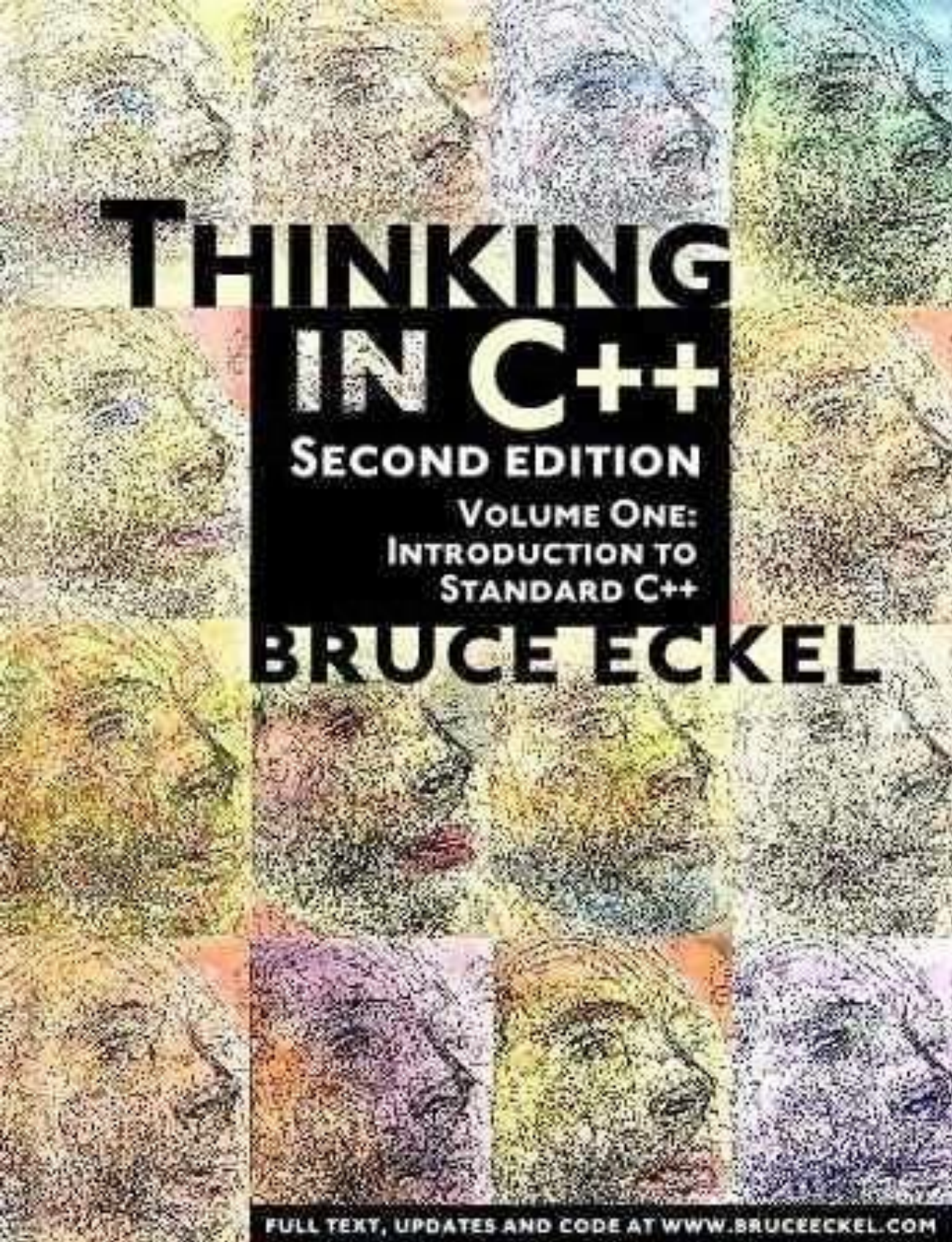
# Textbooks

- Thinking In C++,Ver. 2,Vol. 1 & 2
- C++ Primer, Ver 5
- References:
  - The C++ Programming Language
  - C++:The Core Language
  - Essential C++
  - Effective C++
  - Inside the C++ Object Model
  - C++ Templates





C++编程思想 机械工业出版社





# Bruce Eckel

- BRUCE ECKEL is the author of “Thinking in C++”, which won the Software Development Jolt Award for best book of 1995. He's been professionally programming for 20 years and has been teaching people throughout the world how to program with objects since 1986. He was a voting member of the C++ Standards Committee.
- <http://mindview.net>




# Online Resources for C++

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

**cppreference.com** [Create account](#)  [Search](#)


Page [Discussion](#) [View](#) [View source](#) [History](#)

 **CppCon 2022**  
It's the annual, week-long gathering for the entire C++ community. [Register now!](#)


## C++ reference


C++98, C++03, C++11, C++14, C++17, C++20, C++23 | [Compiler support C++11, C++14, C++17, C++20, C++23](#)


<b>Freestanding implementations</b> <b>Language</b> Basic concepts Keywords Preprocessor Expressions Declaration Initialization Functions Statements Classes Overloading Templates Exceptions <b>Standard library (headers)</b>	<b>Diagnostics library</b> <a href="#">basic_stacktrace (C++23)</a> <b>Memory management library</b> <a href="#">unique_ptr (C++11)</a> <a href="#">shared_ptr (C++11)</a> <b>General utilities library</b> Function objects — <a href="#">hash (C++11)</a> Utility functions <a href="#">pair — tuple (C++11)</a> <a href="#">optional (C++17)</a> — <a href="#">any (C++17)</a> <a href="#">variant (C++17)</a> <a href="#">String conversions (C++17)</a> <a href="#">Formatting (C++20)</a> <a href="#">Bit manipulation (C++20)</a> <b>Strings library</b>	<b>Iterators library</b> <b>Ranges library (C++20)</b> <b>Algorithms library</b> <a href="#">Constrained algorithms (C++20)</a> <b>Numerics library</b> Common math functions <a href="#">Mathematical special functions (C++17)</a> <a href="#">Mathematical constants (C++20)</a> Numeric algorithms Pseudo-random number generation <a href="#">Floating-point environment (C++11)</a> <a href="#">complex — valarray</a> <b>Date and time library</b> <b>Localizations library</b>
---	---	---


 [Get Started!](#) [Tour](#) [Core Guidelines](#) [Super-FAQ](#) [Standardization](#) [About](#)


## News, Status & Discussion about Standard C++


**Follow All Posts**   
*The home of Standard C++ on the web — news, status and discussion about the C++ standard on all compilers and platforms.*


**Recent Highlights**   
CppCon 2023 Lock-free Atomic Shared Pointers Without a Split Reference Count! — Daniel Anderson  
By Blog Staff | Sep 16, 2023 07:58 PM  
CppCon 2023 Why Loops End — Lisa Lippincott  
By Blog Staff | Sep 15, 2023 08:04 PM  
CppCon 2023 Back to Basics: Initialization — Ben Saks  
By Blog Staff | Sep 15, 2023 07:00 PM  
CppCon 2023 Getting Build Tools to Talk to Each Other: Lessons Learned — Diego Rodriguez-Lousada  
By Blog Staff | Sep 15, 2023 02:56 PM  
CppCon 2023 Taro: Task-graph-based Asynchronous Programming Using C++ Coroutines — Dian-Lun Lin  
By Blog Staff | Sep 15, 2023 09:00 AM

**Articles & Books**   
Inside STL: The deque, design — Raymond Chen  
By Blog Staff | Sep 14, 2023 04:17 PM  
C++ Exceptions and Memory Allocation Failure — Wu Yongwei  
By Blog Staff | Sep 12, 2023 03:54 PM  
Five Advanced Initialization Techniques in C++ — Bartłomiej Filipek  
By Blog Staff | Sep 8, 2023 12:28 PM  
Inside STL: The lists — Raymond Chen  
By Blog Staff | Sep 6, 2023 12:23 PM  
Inside STL: The string — Raymond Chen

**Recent CppCast Podcasts**   
[AI and Random Numbers](#)  
Date: Fri, 15 Sep 2023  
[Automatic Static Analysis](#)  
Date: Fri, 1 Sep 2023  
[Soagen](#)  
Date: Fri, 18 Aug 2023  
[How CLion works under the hood](#)  
Date: Fri, 4 Aug 2023  
[Safety, Security and Modern C++, with Bjarne Stroustrup](#)  
Date: Fri, 21 Jul 2023

**Recent C++ Weekly Podcasts**   
[C++23's std::unreachable](#)  
Date: Mon, 11 Sep 2023  
[Google's Bloaty McBlotface](#)  
Date: Mon, 4 Sep 2023  
[Finally! C++23's std::views::enumerate](#)  
Date: Mon, 28 Aug 2023  
[constexpr + mutable?](#)  
Date: Mon, 21 Aug 2023

**Product News**   
No, C++ static analysis does not have to be painful — Geoffrey Adde



# Assessment

- Lab: 50% (including the in-class quiz)
  - ! In week 2,4,6,8,10,12, (14,16)
  - ! To submit on-line, to be judged manually
  - ! Deadline is **DEAD on Midnight of Saturday of (week 3, 5, 7, 9, 11, 13, 16)**
  - ! Per day delay cost: **10% of the full score**
- Final exam: 50%, on PTA

# Resources

- 课件PPT:
  - <https://course.zju.edu.cn> (学在浙大)
- 作业:
  - <https://pintia.cn> (PTA)
  - 面向对象程序设计2024-2025秋冬习题集

## 个人中心

昵称: chenyue

邮箱地址: chenyue@zju.edu.cn

密码管理: [重置密码](#)

中国大学 MOOC: [绑定](#)

应邀做题:  [接受邀请](#)

### 绑定学号

zju - 浙江大学 ▼

姓名

学号

验证码 (通过教师获得)

[绑定](#)

### 已绑学号

暂无学号

# PTA绑定码: **428977**

分享绑定码



该用户组的绑定码为:

428977

已注册的学生可访问以下链接或扫描下方二维码后绑定 浙江大学 校园账户，此前已绑定过的用户无需再次绑定。

<https://pintia.cn/home/bindings?k=428977&c=zju&id=1>



提示: 右键或长按二维码保存图片。

# Course policy

- 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



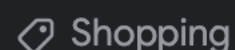
All



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:
  - DingTalk (钉钉) – Preferred
  - oop\_zju@163.com / zhpcui@zju.edu.cn
- Title starts with "[OOP]"
- State your **name** and **id** in the text
- Prefer plain text e-mail
  - NO background, music, color, font styles ...

# Acknowledge



翁恺



陈越



许威威



陈翔

# Introduction to C++

The trip begins...



# 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 
- Linux, macOS:
  - g++, clang, ...(compiler)
  - visual studio code (editor) 
- Any compiler 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 and ES
- Flexible

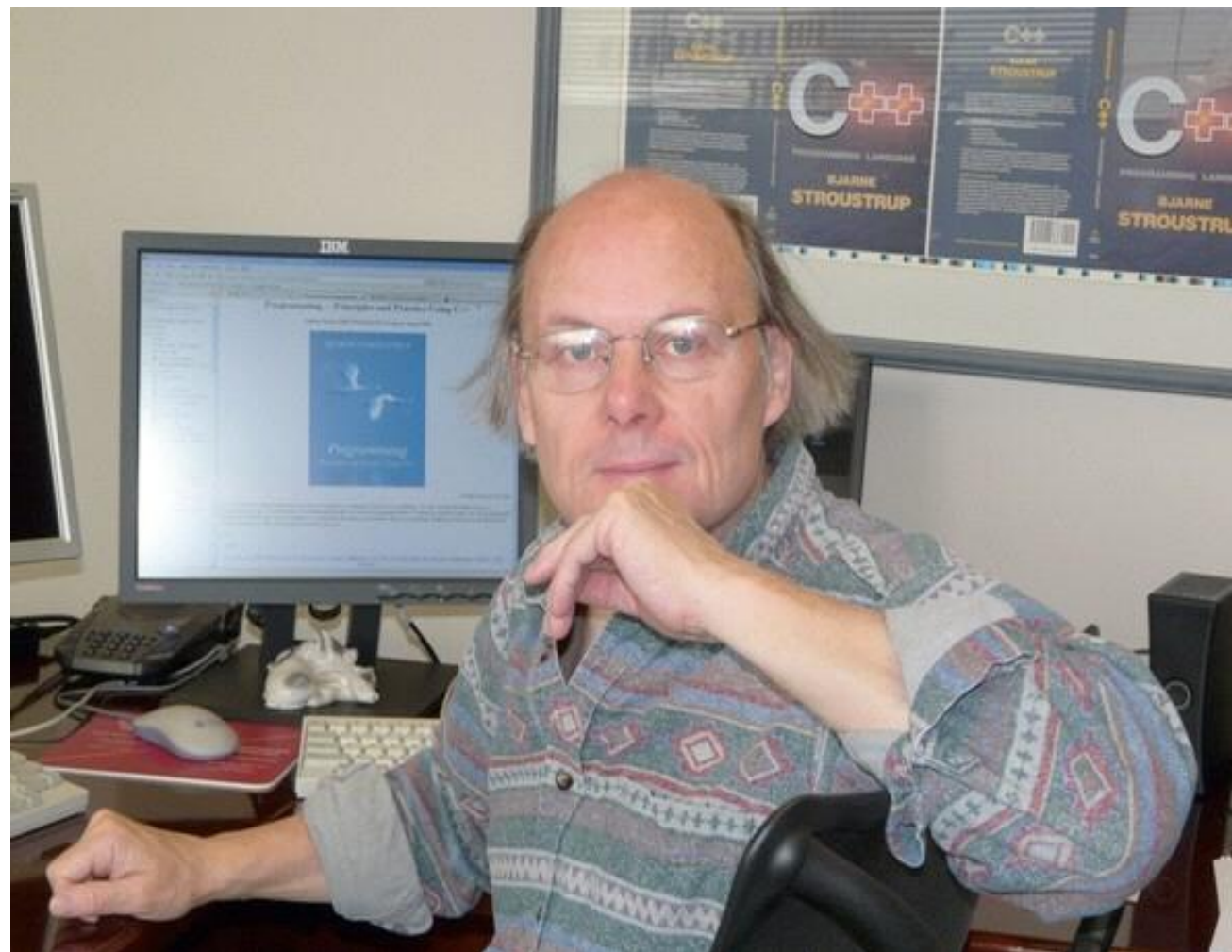
- **Weakness**

- Insufficient type checking
- Poor support for programming-in-the-large
- Procedure-oriented programming



# Bjarne Stroustrup 🔊

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



- Oct. 2002, Stroustrup visited Zhejiang Univ.















Stroustrup visited zju in 2005

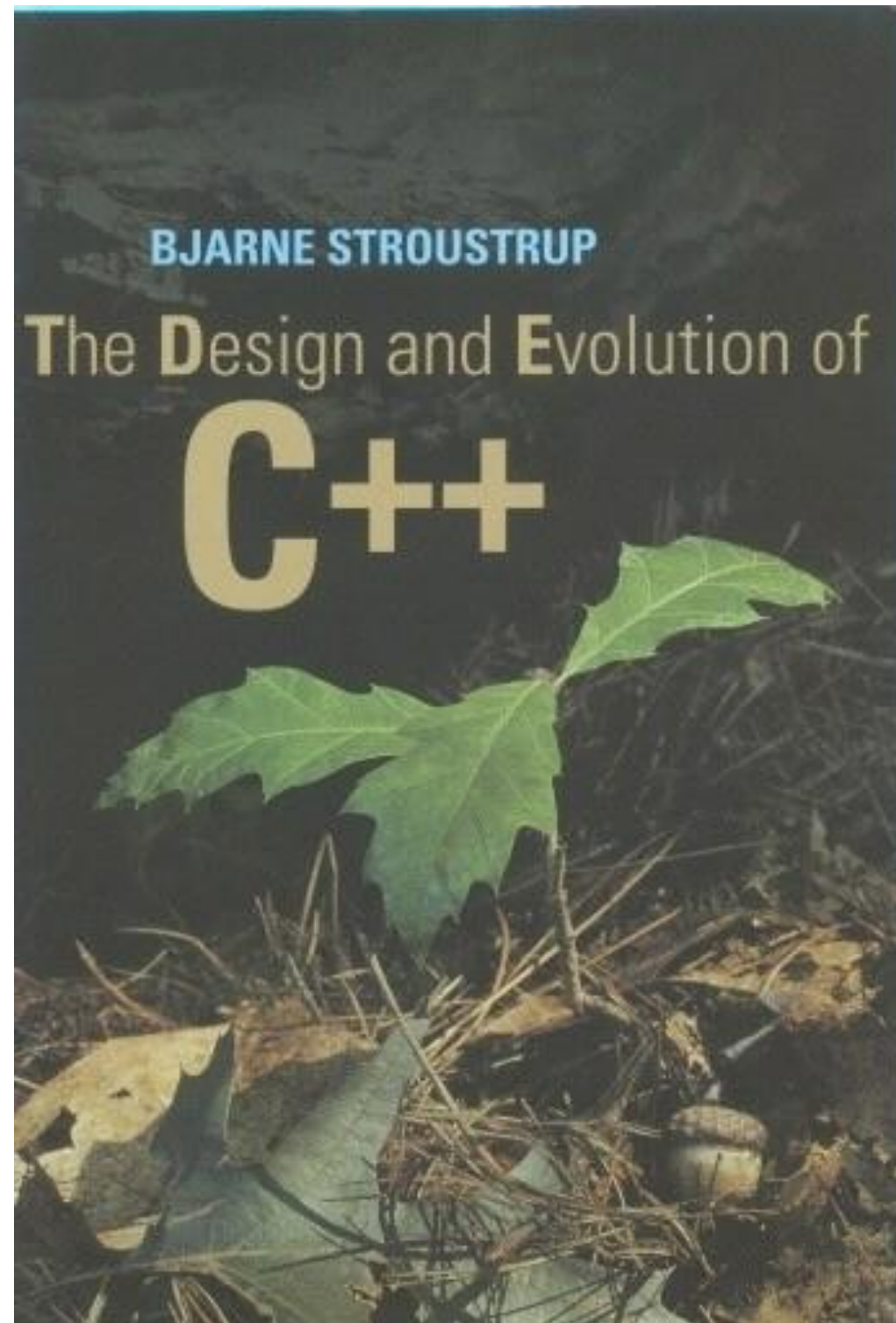




C++ 之父：爱吃辣子鸡， C++20 会非常出色  
<https://www.infoq.cn/article/rv3SX2V8rtRaJj9B17xZ>



# The Design and Evolution of C++



Bjarne Stroustrup, Addison-Wesley, ISBN 0-201-54330-3

# Brief History of C++ (1)

- 1978: BS at Cambridge, UK. Simulation program in *Simula*
  - Supports classes, inheritance, and type check
  - Poor performance
  - <http://www.engin.umd.umich.edu/CIS/course.des/cis400/simula/simula.html>



# Brief History of C++ (2)

- 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
- Standard C++ in 1998: ISO/IEC 14882  
(<http://www.open-std.org/jtc1/sc22/wg21/>)

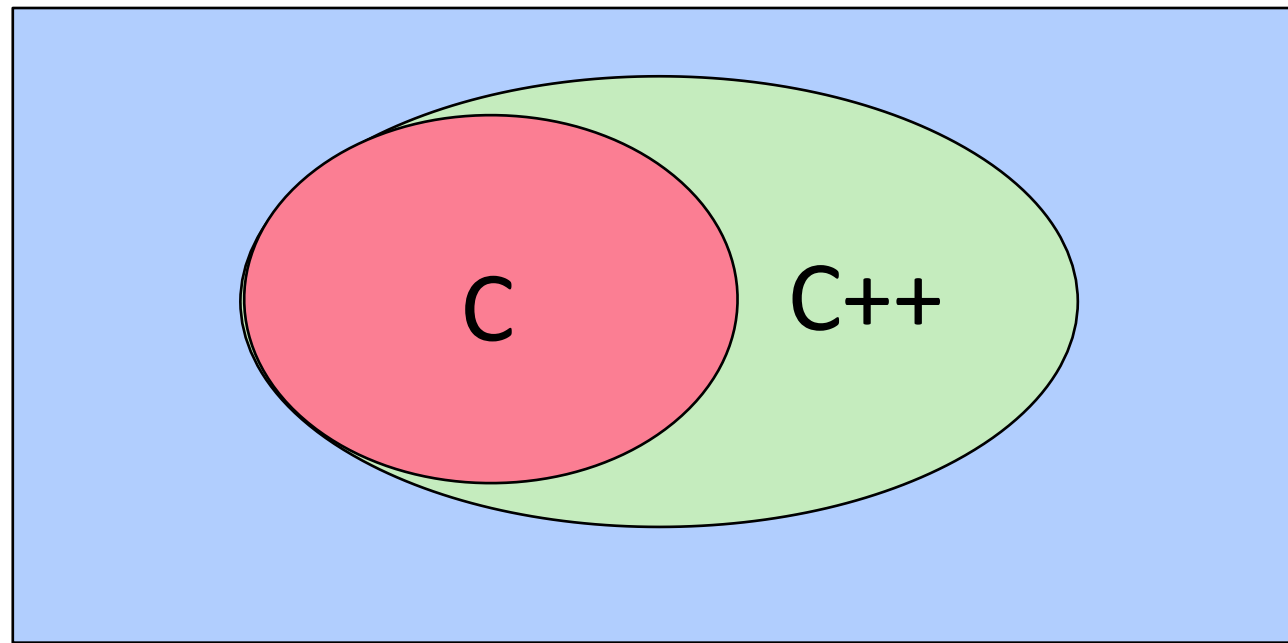
# Brief History of C++ (3)

- 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 object-oriented programming (from *Simula* and *SmallTalk*)

# C and C++



- C++ builds on C
- Knowledge of C helps you in C++
- C++ support 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
- More safe and powerful memory management
- Support for OOP
- Templates
- Exception handling
- More extensive libraries, STL

# C++

- C++ can be viewed as a “better” C
  - $C++ \Rightarrow C=C+1$
- but ...
  - C++ is not C
  - Focus on C++ as a language in its own right
- C++ is a hybrid language, supports
  - Procedure-oriented programming
  - Object-oriented programming
  - Generic programming

# Criticism of ...

- 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)