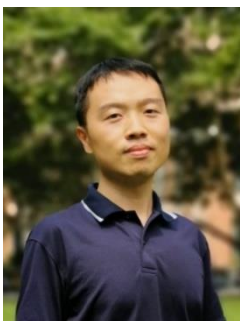




程序设计II



张书航 助理教授

电子邮件: zhangsh52@mail.sysu.edu.cn
个人主页: shuhangz.github.io



李同文 助理教授

电子邮件: litw8@mail.sysu.edu.cn

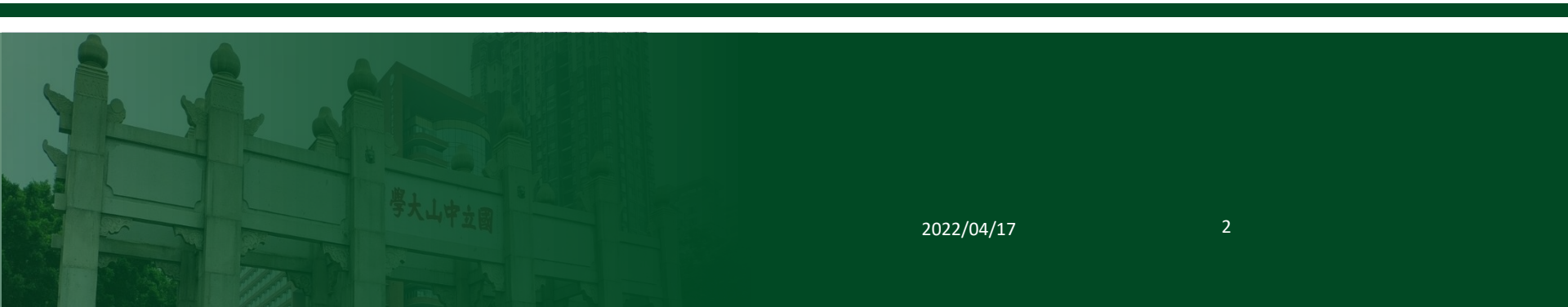


中山大學
SUN YAT-SEN UNIVERSITY



From C to C++ 程序设计方法的发展

程序设计II



• 教材第十五章

15.1 Introduction

15.2 C++

15.3 A Simple Program: Adding Two Integers

15.3.1 Addition Program in C++

15.3.2 `<iostream>` Header

15.3.3 `main` Function

15.3.4 Variable Declarations

15.3.5 Standard Output Stream and Standard Input Stream Objects

15.3.6 `std::endl` Stream Manipulator

15.3.7 `std::` Explained

15.3.8 Concatenated Stream Outputs

15.3.9 `return` Statement Not Required in `main`

15.3.10 Operator Overloading

15.4 C++ Standard Library

15.5 Header Files

15.6 Inline Functions

15.7 C++ Keywords

15.8 References and Reference Parameters

15.8.1 Reference Parameters

15.8.2 Passing Arguments by Value and by Reference

15.8.3 References as Aliases within a Function

15.8.4 Returning a Reference from a Function

15.8.5 Error Messages for Uninitialized References

• 15.1–15.16

15.9 Empty Parameter Lists

15.10 Default Arguments

15.11 Unary Scope Resolution Operator

15.12 Function Overloading

15.13 Function Templates

15.13.1 Defining a Function Template

15.13.2 Using a Function Template

15.14 Introduction to Object Technology and the UML

15.14.1 Basic Object Technology Concepts

15.14.2 Classes, Data Members and Member Functions

15.14.3 Object-Oriented Analysis and Design

15.14.4 The Unified Modeling Language

15.15 Introduction to C++ Standard Library Class Template `vector`

15.15.1 Problems Associated with C-Style Pointer-Based Arrays

15.15.2 Class Template `vector`

15.15.3 Exception Handling: Processing an Out-of-Range Index

15.16 Wrap-Up

• 程序

- 为实现特定目标或解决特定问题而用计算机语言编写的**命令序列的集合**。

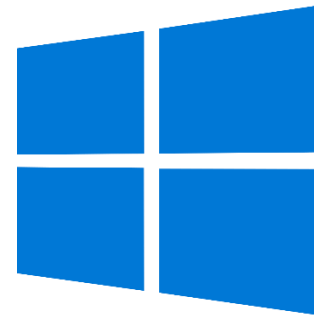
• 程序设计

- 程序设计是指在计算机上使用可执行的程序代码来**有效描述特定问题并执行解决算法的过程**。

程序是人类思维火花的实现定格，呈现出静态特征；而作为产生程序的过程，**程序设计却是动态的，它反映了人类思维的规律和模式。**



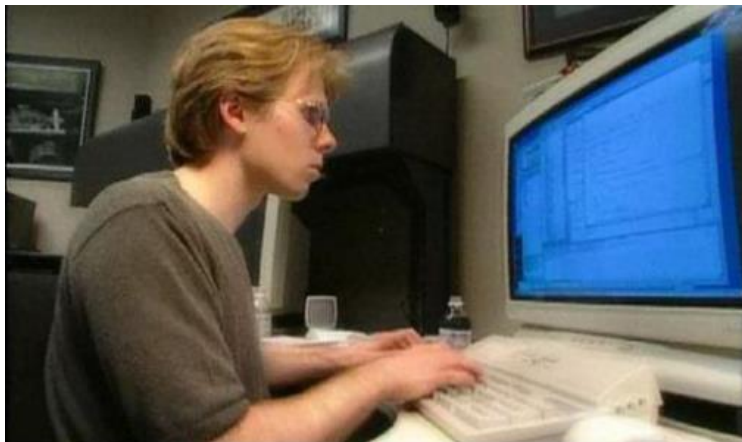
- 软件是计算机程序、所要求的文档资料和在计算机上运行时所必需的数据的总和。
- 软件开发经历的3个主要阶段：
 - 程序设计时期
 - 软件=程序+说明 时期
 - 软件=**程序+文档+数据** 时期
- 程序是软件的重要组成部分



- 程序编写是软件开发工程中的一个关键阶段。软件的质量主要是通过程序的质量来体现的，因此程序**设计在软件开发过程中占有十分重要的地位。**
- 但是，最初的程序设计全凭设计者个人经验和技艺独立进行，是一种典型的**手工艺智力劳动。**

• 程序员

- 从事程序开发、代码维护的专业人员。





Bill Gates



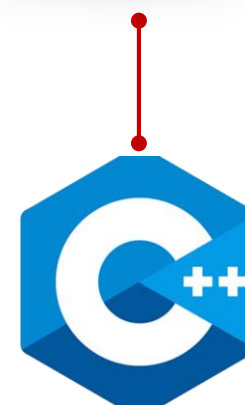
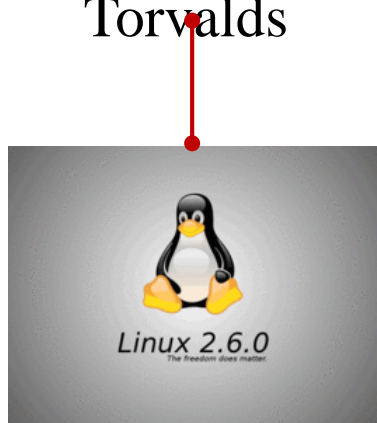
Linus Benedict
Torvalds



James Gosling



Bjarne Stroustrup



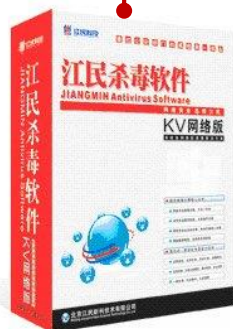
杰出程序员



张小龙



王江民



雷军



马化腾



程序届的精英?



中山大學
SUN YAT-SEN UNIVERSITY



“网络三剑客”

王志东

新浪网创始人

张朝阳

搜狐董事长兼CEO

丁磊

网易公司创始人

Ja!



程序届的精英?



中山大學
SUN YAT-SEN UNIVERSITY



陈天桥

Nein!



程序届的精英?



中山大學
SUN YAT-SEN UNIVERSITY



马云

Nein!



程序届的精英?



中山大學
SUN YAT-SEN UNIVERSITY



刘强东

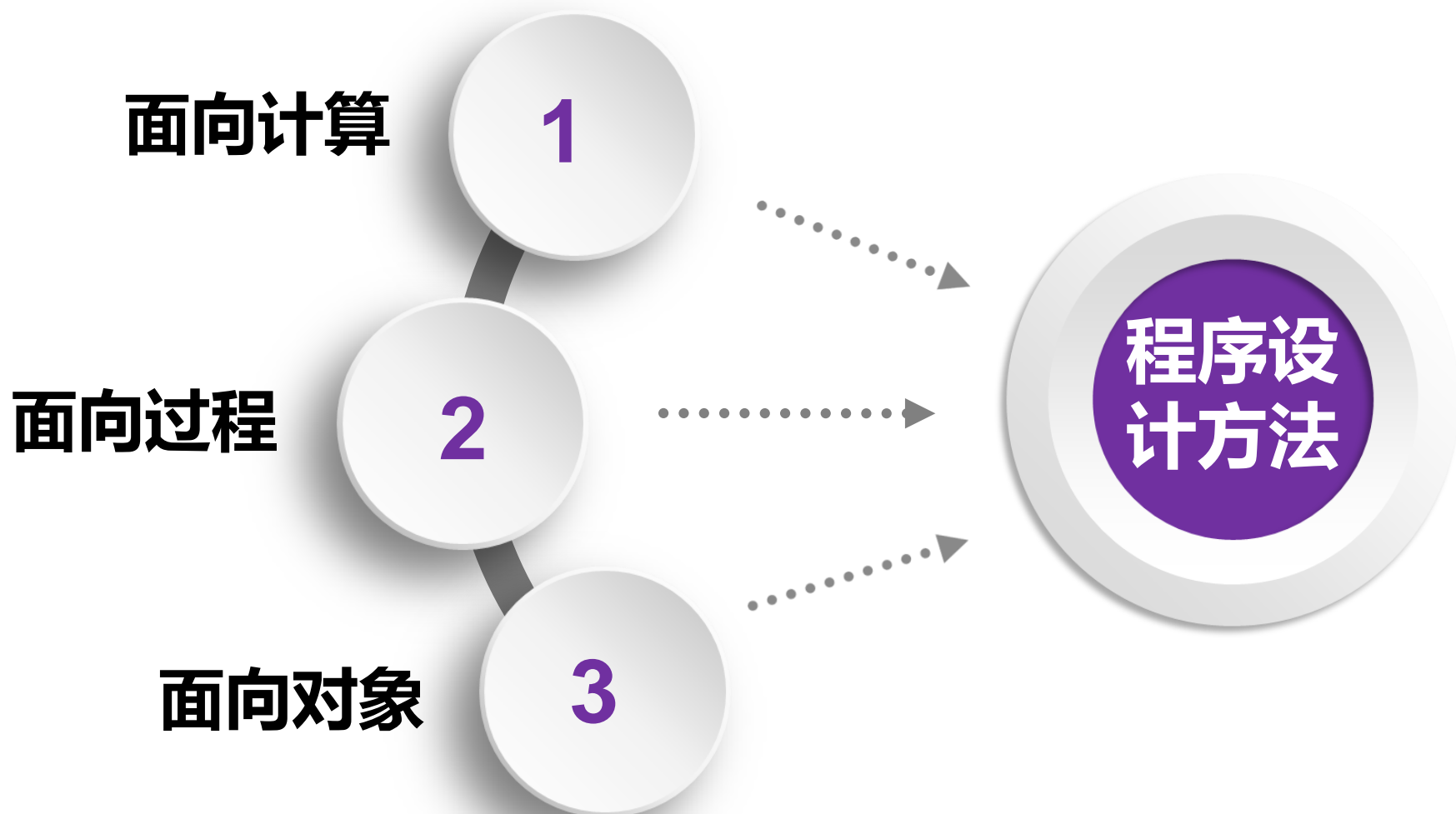
Nein!





程序设计方法的发展

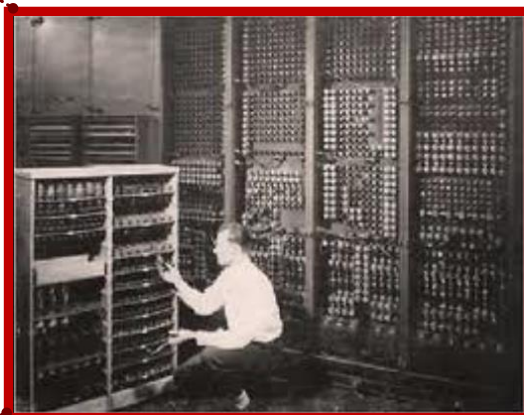




典型语言包括：机
器指令、汇编语言

- 二进制指令组成

- 随后出现汇编语言

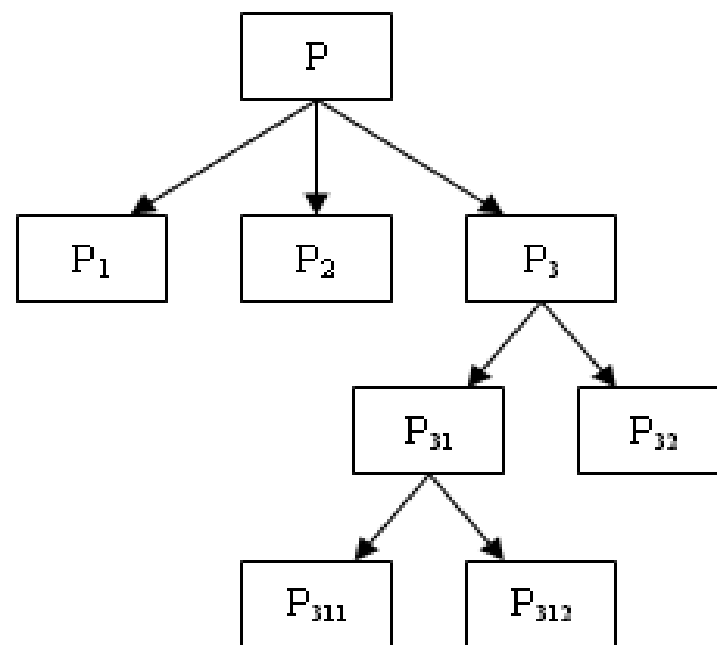


- 不考虑可读性

- 注重机器、逐一执行



典型语言包括：C语言、Pascal语言



结构化程序设计的树状图

问题——软件危机



客户需求与
解决方案不
一致



软件开发进展
情况较难衡量

落后的软件生产
方式无法满足迅
速增长的计算机
软件需求。

软件开发质
量难以评价

软件没有“磨损”
的概念。



软件危机产生的原因



客户是这样描述需求的



项目经理是这么理解的



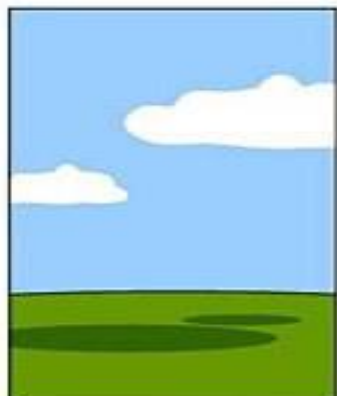
分析师是这么设计的



程序员是这么编写的



商业顾问是这么描绘的



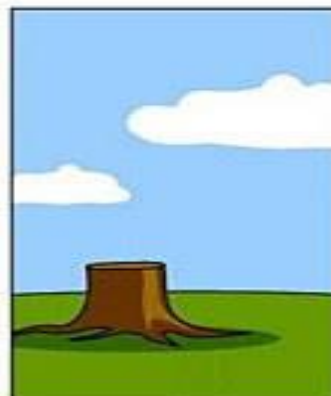
项目书写出来是这样的



操作中用了这样的工具



客户是这么建造的



提供的支持就这个样子



而这才是
客户真正需要的



软件工程的出现带来了编程方法的变革

数据与操作的分离

可重用性差

维护代价高

面向过程

典型语言包括：
VC语言、Java语言、C#语言

面向对象

数据和操作的封装

可重用性极大提高

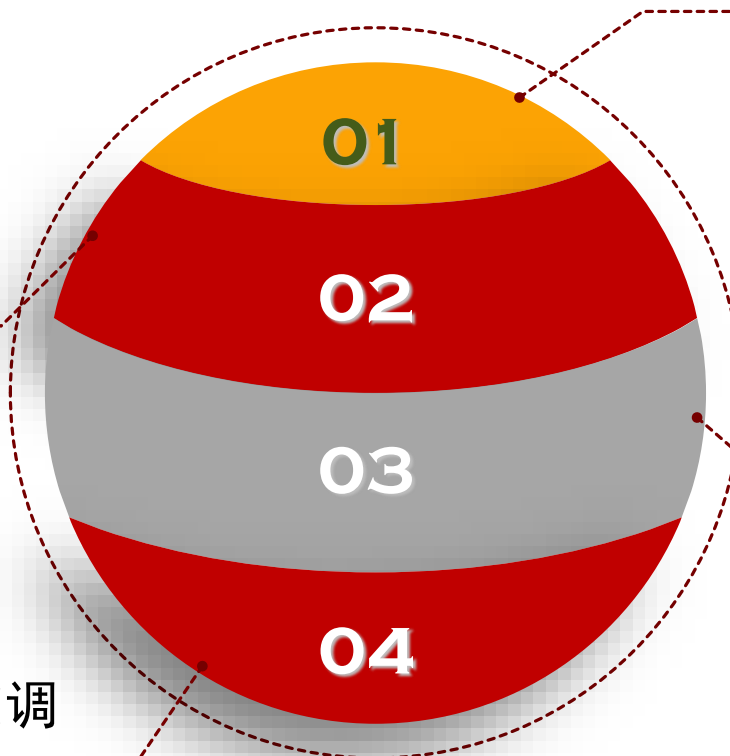
维护代价降低



- 关键是把数据和函数封装成类, 通过对象调用



- 便于程序的修改和调试, 便于团队协作, 便于大型程序维护

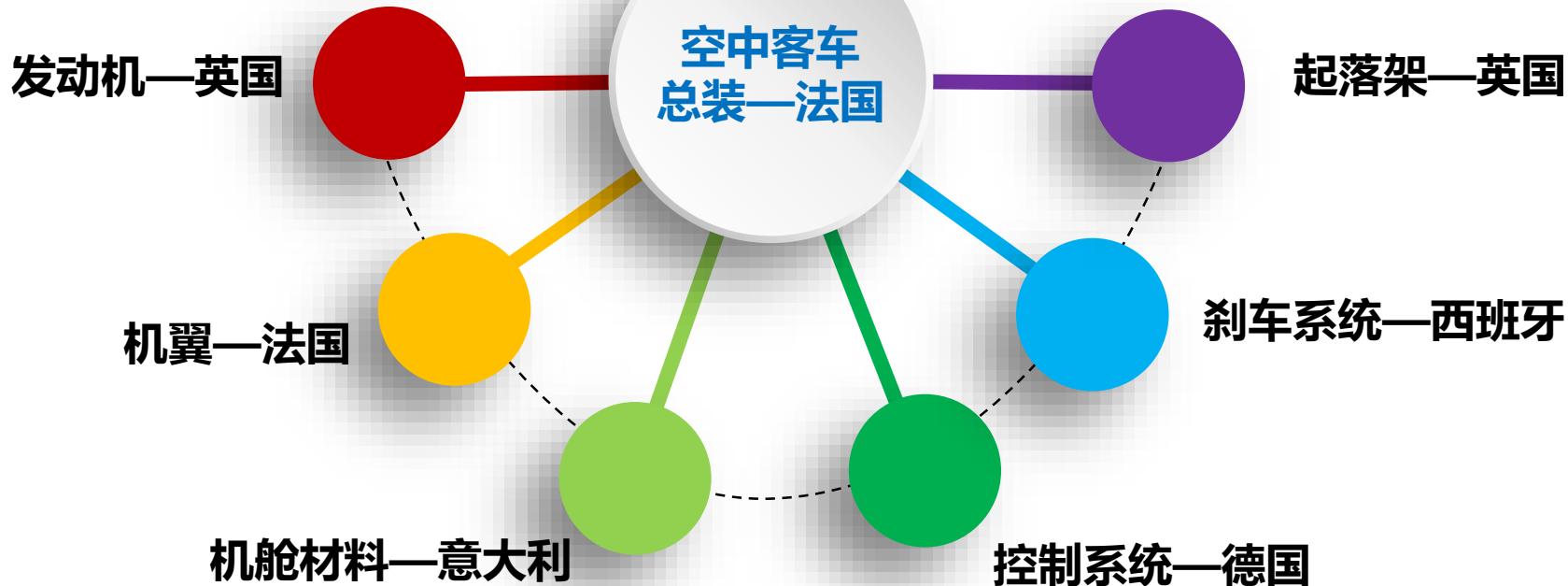
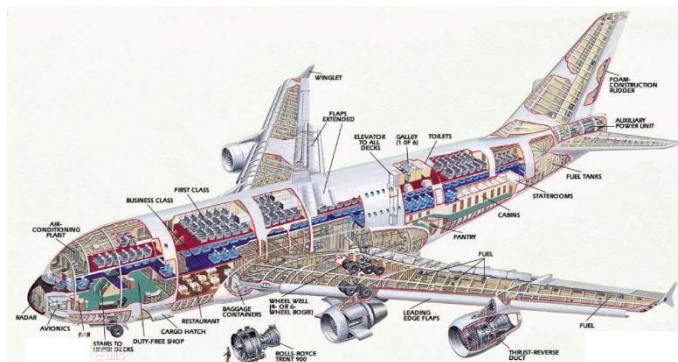


- 数据与操作联系在一起, 便于程序的修改和调试



- 编程人员能在可视化的环境中进行组件化编程

面向对象的程序设计的优点



面向对象程序设计的成果



操作系统



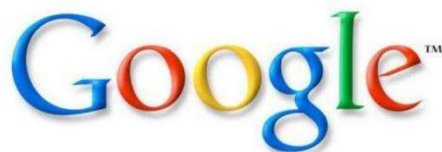
数据库



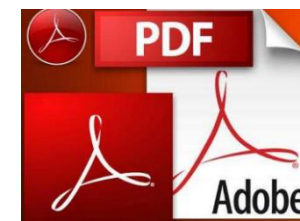
社交平台



文字操作



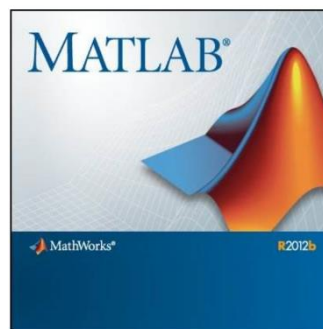
搜索引擎



图文处理



传输工具



数据分析



游戏娱乐

面向过程vs面向对象

•问题： 如何把大象装进冰箱？

面向过程：

为了把大象装进冰箱，需要3个过程。

- 1) 把冰箱门打开（得到打开门的冰箱）
- 2) 把大象装进去（打开门后，得到里面装着大象的冰箱）
- 3) 把冰箱门关上（打开门、装好大象后，获得关好门的冰箱）

每个过程有一个阶段性的目标，依次完成这些过程，就能把大象装进冰箱。

1:
冰箱开门(冰箱)
冰箱装进(冰箱, 大象)
冰箱关门(冰箱)
==换个写法
(冰箱开门 冰箱)
(冰箱装进 冰箱 大象)
(冰箱关门 冰箱)

2:
冰箱关门(冰箱装进(冰箱开门
(冰箱), 大象))
==换个写法
(冰箱关门 (冰箱装进 (冰箱开门
冰箱) 大象))

第一步，把冰箱门打开



第二步，把大象装进去



第三步，把冰箱门关上



•问题： 如何把大象装进冰箱？

面向对象：

为了把大象装进冰箱，需要做三个动作（或者叫行为）。

每个动作有一个执行者，它就是对象。

- 1) 冰箱，你给我把门打开
- 2) 冰箱，你给我把大象装进去（或者说，大象，你给我钻到冰箱里去）
- 3) 冰箱，你给我把门关上

依次做这些动作，就能把大象装进冰箱。

1:

冰箱.开门()

冰箱.装进(大象)

冰箱.关门()

2:

冰箱.开门().装进(大象).关门()

问题：如何把大象装进冰箱？



- Q: 面向对象的话只能是冰箱自己关门吧？大象应该关不了冰箱的门。
 - A: 程序员相当于指挥官级别的，坐阵帐中，只能给某人/某物发命令（获对象，调方法，填参数），而且只能得到命令执行的最终结果（方法返回值）。下面人怎么鼓捣，不太晓得。
 - 就是只能告诉编译器：“传我命令，让冰箱关上门。”
 - 然后编译器出去了。
 - 过了一会.....
 - ”报———“编译器回来了，” 报告！冰箱关门成功！ “（可以是返回了true、1等）
 - 嗯。程序员很满意。
 - 具体怎么关的门，只有冰箱那的人才知道。可能是有人关了（冰箱.关门()可不一定就是冰箱自己关的，可能求助于其他人了什么的，也就是可以调用工具类之类的“外援”），也可能是冰箱很智能，发现有命令来自己关了，比如声控啥的.....
 - 总之，具体过程被封装，指挥官/程序员不晓得。当然，有经验的指挥官/程序员也是会想到大概怎么实现。

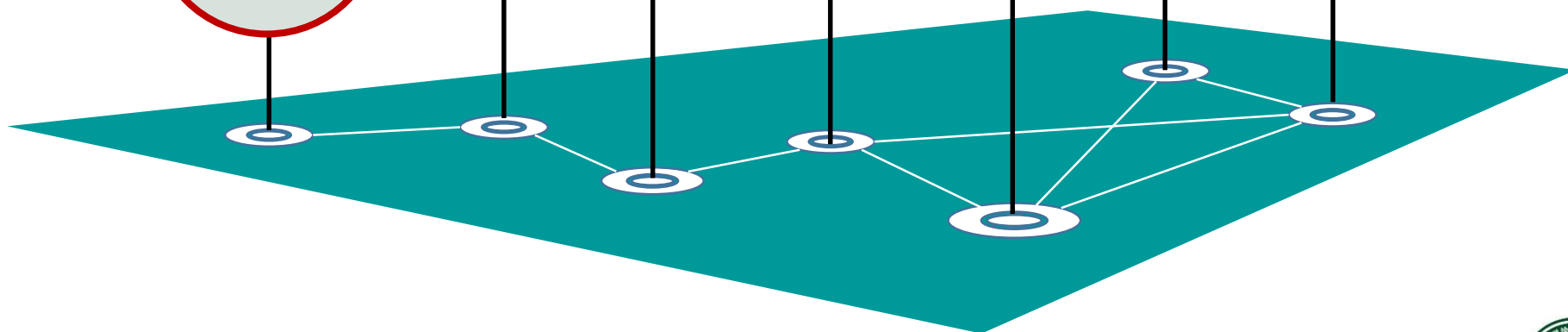
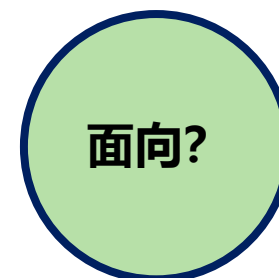
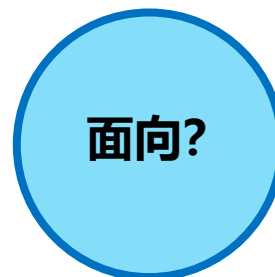
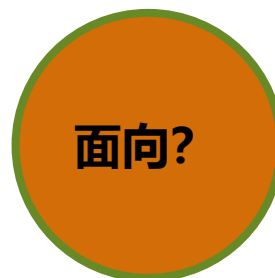


- 面向方面的程序设计 (Aspect-Oriented Programming)
- 面向智能体的程序设计 (Agent-Oriented Programming)
- 面向组件的程序设计 (Component-Oriented Programming)

思考:程序设计方法的未来?



AlphaGo





From C to C++



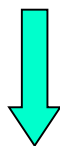
- **1969年**，美国贝尔实验室的Ken Thompson为DEC PDP-7计算机设计了一个操作系统软件，这就是最早的UNIX。
- **1969 - 1971**，Ken Thompson又根据剑桥大学的Martin Richards设计的BCPL语言为UNIX设计了一种便于编写系统软件的语言，命名为B。B语言是一种无类型的语言，直接对机器字操作，这一点和后来的C语言有很大不同。



Ken Thompson

- 1972–1973年，同在贝尔实验室的Denis Ritchie改造了B语言，为其添加了数据类型的概念，并将原来的解释程序改写为可以在直接生成机器代码的编译程序，然后将其命名为C。

C语言的设计者

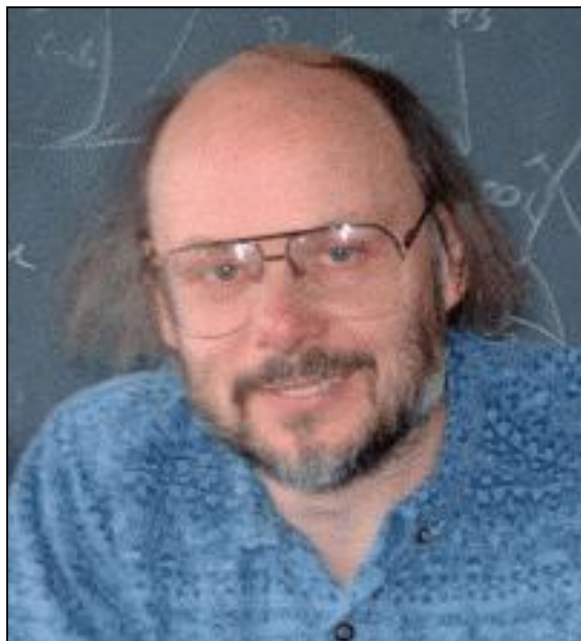


**Dennis
M.Ritchie**



- **1973** 年, Ken Thompson小组在PDP-11机上用C重新改写了UNIX的内核。与此同时, C语言的编译程序也被移植到IBM 360/370、Honeywell 11以及VAX-11/780等多种计算机上, 迅速成为应用最广泛的系统程序设计语言。
- **1983**年, 贝尔实验室的Bjarne Stroustrup博士及其同事开始对C语言进行改进和扩充, 将“类”的概念引入了C语言, 构成了最早的C++语言 (1983)。后来, Stroustrup和他的同事们又为C++引进了运算符重载、引用、虚函数等许多特性, 并使之更加精炼。

C++是一个“更好的C语言” —Stroustrup



C++语言的设计者

Bjarne Stroustrup

- He wanted a language that was:
 - Fast
 - Simple to Use
 - Cross-platform
 - Had high level features

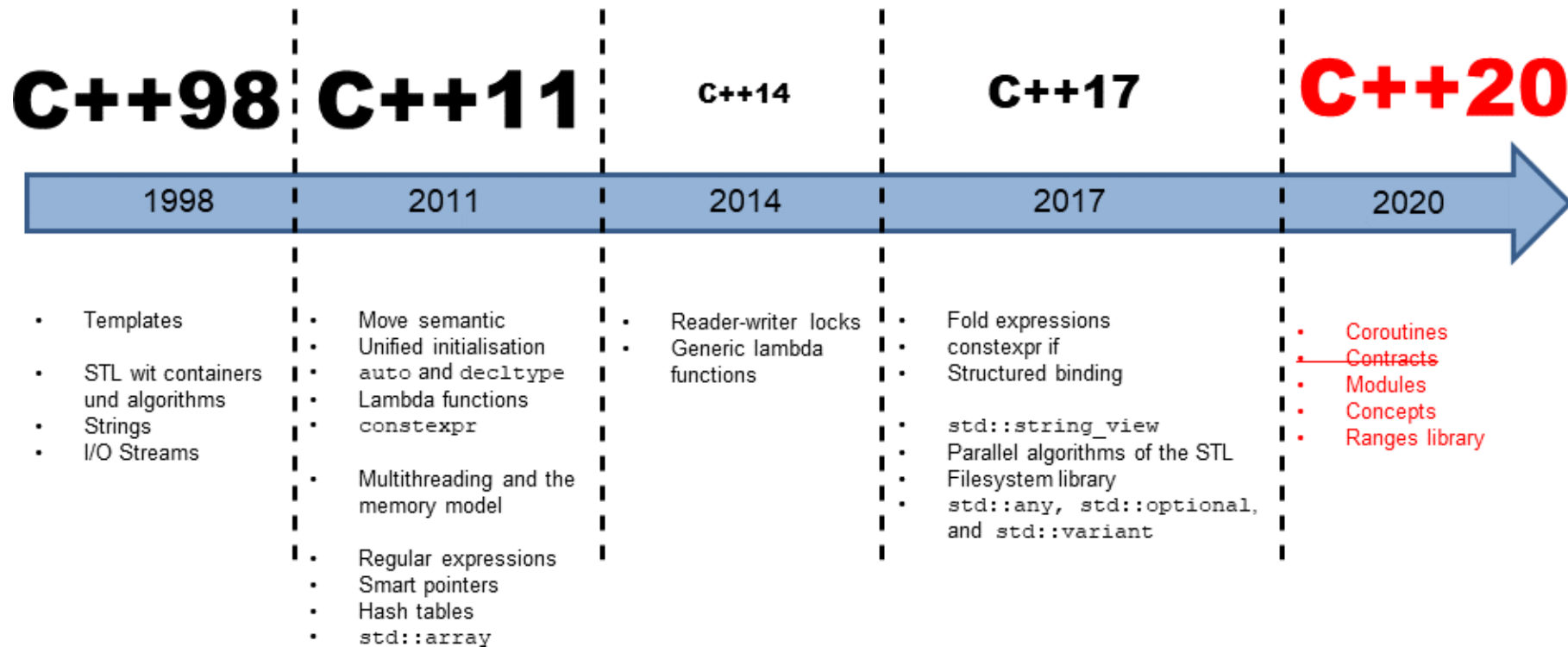


Image taken from <https://www.modernescpp.com/>

- 语言简洁

- 使用方便

- 语法丰富

- 功能强大

- 可移植性好

◆ 全面兼容C

- 它保持了C的简洁、高效和接近汇编语言等特点
- 对C的类型系统进行了改革和扩充
- C++也支持面向过程的程序设计

◆ 支持面向对象的方法

C/C++语言仍是目前应用最为广泛的程序设计语言。

继承--C语言的精髓

高效性、灵活性;

扩充--面向对象机制

适合开发大型软件;

弥补--C语言的缺点

类型更安全(编译器查错能力更强);

支持代码重用;

- 1985年10月, Cfront Release 1.0发布。
- 1987年12月, GNU C++发布。
- 1988年1 月, 第一个Oregon Software C++发布
- 1988年6 月, 第一个Zortech C++发布。
- 1990年5 月, 第一个Borland C++发布。
- 1992年2 月, 第一个Dec C++发布。
- 1992年3 月, 第一个Microsoft C++发布。
- 1992年5月, 第一个IBM C++发布。

2021常用编程语言排行

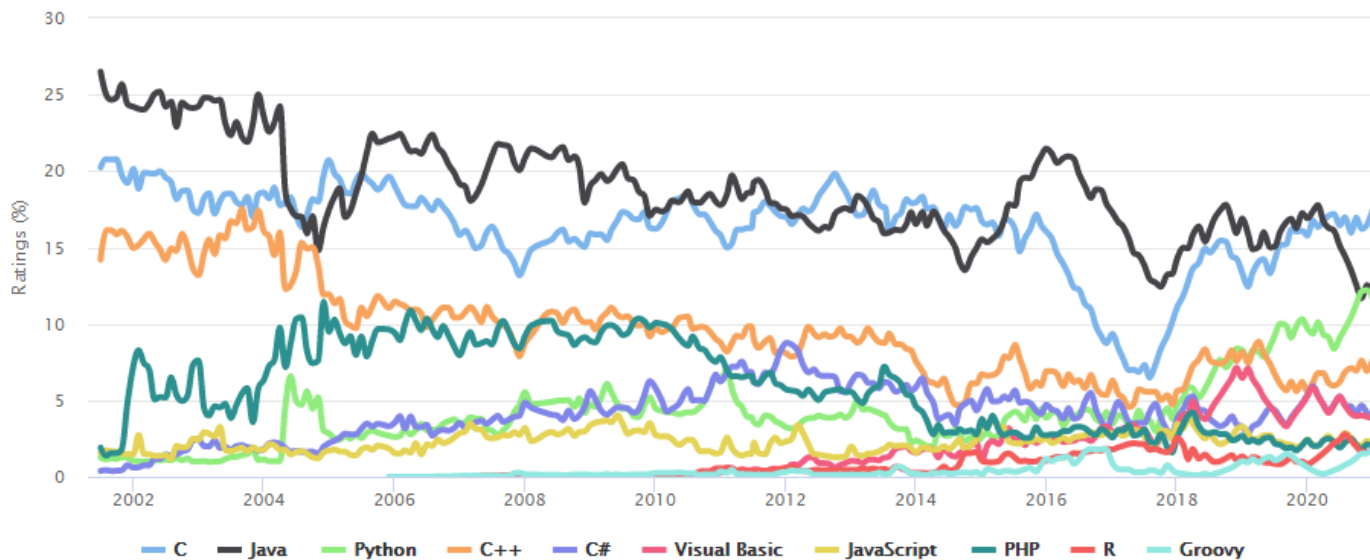


2021年1月

Jan 2021	Jan 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	17.38%	+1.61%
2	1	▼	Java	11.96%	-4.93%
3	3		Python	11.72%	+2.01%
4	4		C++	7.56%	+1.99%
5	5		C#	3.95%	-1.40%
6	6		Visual Basic	3.84%	-1.44%
7	7		JavaScript	2.20%	-0.25%

TIOBE Programming Community Index

Source: www.tiobe.com



2022常用编程语言排行

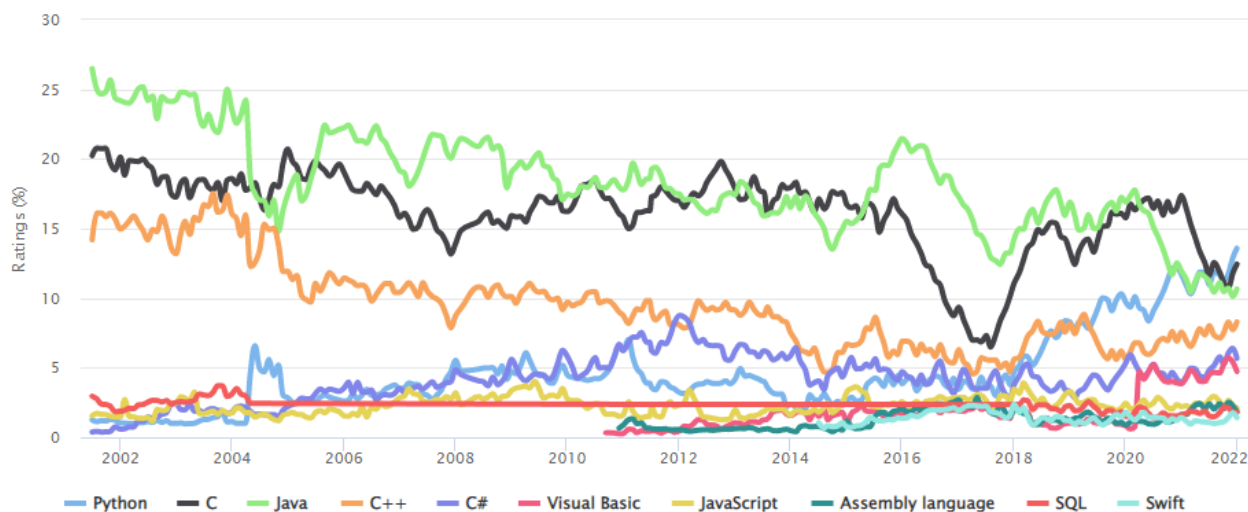


2022年1月

Jan 2022	Jan 2021	Change	Programming Language	Ratings	Change
1	3	▲	Python	13.58%	+1.86%
2	1	▼	C	12.44%	-4.94%
3	2	▼	Java	10.66%	-1.30%
4	4		C++	8.29%	+0.73%
5	5		C#	5.68%	+1.73%
6	6		Visual Basic	4.74%	+0.90%
7	7		JavaScript	2.09%	-0.11%

TIOBE Programming Community Index

Source: www.tiobe.com





C++ 101—简单程序





- 对比课本图2.5 p32, 使用C++重新编写这个程序。看看C++与C的不同之处
- 扩展名
 - C的扩展名.c
 - C++扩展名: .cpp, .cxx, .C (大写)
- C++风格特点
 - 注释
 - 输入输出流





C

```
// Fig. 2.5: fig02_05.c
// Addition program
#include <stdio.h>

// function main begins program execution
int main( void )
{
    int integer1; // first to be entered by user
    number
    int integer2; // second number to be entered by
    user
    printf( "Enter first integer\n" ); // prompt
    scanf( "%d", &integer1 ); // read an integer

    printf( "Enter second integer\n" ); // prompt
    scanf( "%d", &integer2 ); // read an integer

    int sum; // variable in which sum will be stored
    sum = integer1 + integer2; // assign total to
    sum

    printf( "Sum is %d\n", sum ); // print sum
} // end function main
```

C++

```
// Fig. 15.1: fig15_01.cpp
// Addition program that displays the sum of two
// numbers.
#include <iostream> // allows program to perform
// input and output

int main()
{
    std::cout << "Enter first integer: "; // prompt
    user for data
    int number1;
    std::cin >> number1; // read first integer from
    user into number1

    std::cout << "Enter second integer: "; // prompt
    user for data
    int number2;
    std::cin >> number2; // read second integer from
    user into number2
    int sum = number1 + number2; // add the numbers;
    store result in sum
    std::cout << "Sum is " << sum << std::endl; //
    display sum; end line
}
```





C

```
1 // Fig. 2.5: fig02_05.c
2 // Addition program.
3 #include <stdio.h>
4
5 // function main begins program execution
6 int main( void )
7 {
8     int integer1; // first number to be entered by user
9     int integer2; // second number to be entered by user
10
11     printf( "Enter first integer\n" ); // prompt
12     scanf( "%d", &integer1 ); // read an integer
13
14     printf( "Enter second integer\n" ); // prompt
15     scanf( "%d", &integer2 ); // read an integer
16
17     int sum; // variable in which sum will be stored
18     sum = integer1 + integer2; // assign total to sum
19
20     printf( "Sum is %d\n", sum ); // print sum
21 }
```

Fig. 2.5 | Addition program. (Part I of 2.)





C++

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end line
16 }
```

um
ne

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.



注释和空格会被完全忽略



- 注释文本定义：
 - On one line that follows `//`
 - Between `/*` and `*/`
- 下面的代码在C++中都是可用的：

```
1 int main() {return 0;} // Ignored comment.
```

```
1 int main()  
2  
3 {    return 0;  
4 }
```

```
1 int main() {  
2     return /* Ignored comment */ 0;  
3 }
```



15.3 简单程序：两个整数相加



• 第三行iostream

- 第3行的C++预处理器指令展示了标准的C++风格，用于包含标准库中的头文件，这一行告诉C++预处理器要包含**输入/输出流头文件** `<iostream>`的内容，没有扩展名。

- 任何使用C++风格的流输入/输出，向屏幕**输出数据**或从**键盘输入数据**的程序都必须包含这个文件。

- 不用**printf**和**scanf**了

- 第13周会详细讲授C++的IO文件读写

- 和C语言一样，每个C++程序都是以函数**main**（第5行）开始执行的。

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end
16 }
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.



include加不加尖括号?

- `#include <file>` — system include files
- `#include "file"` — local include files

将file的内容复制到当前文件中

```
1 #include "some_file.hpp"  
2 // We can use contents of file "some_file.hpp" now.  
3 int main() { return 0; }
```

15.3 简单程序：两个整数相加



- 函数的参数和返回
 - main左边的关键字`int`表示main返回一个整数值。
 - C++要求你为所有函数指定返回类型可能是`void`。
 - 在C++中，用空括号指定参数列表，相当于在C中指定一个`void`参数列表。
- 在C语言中，在函数定义或原型中使用空括号会产生问题。
 - 它禁用了函数调用中的编译时参数检查，允许调用者向函数传递任何参数

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end program
16 }
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.



Common Programming Error 15.1

Omitting the return type in a C++ function definition is a syntax error.



15.3 简单程序：两个整数相加



- 第8行变量声明。
 - 声明几乎可以放在C++程序中的任何地方，但它们**必须在程序中使用相应的变量之前**出现。
 - “用前先声明”

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end program
16 }
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.



15.3 简单程序：两个整数相加

• 流(stream)

- 在C++中，输出和输入是通过字符流(stream)来完成的。
- 第7行使用标准输出流对象-std::cout和流插入操作符<<
- 第9行使用标准的输入流对象-std::cin和流提取运算符, >>, 从键盘上获取一个值。

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end program
16 }
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.



• 流(stream)

- 当计算机执行第9行的语句时，它等待用户为变量number1输入一个值。
- 用户输入一个整数（字符），然后按Enter键。
- 计算机将数字的字符表示转换为整数，并将此值分配给变量number1。

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end program
16 }
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.

15.3 简单程序：两个整数相加



- 第11行在屏幕上显示 "Enter second integer:"，提示用户进行操作。
- 第13行从用户那里获取变量number2的值。
- 第14行的赋值语句计算变量number1和number2的和，并将结果赋值给变量sum。
- 第15行显示字符串"Sum is"与变量sum的数值，后面跟着std::endl，流操纵器 (stream manipulator)。
- endl是 "end of line "的缩写。
- std::endl流操纵器输出一个新行，然后刷新输出缓冲区。
 - 在某些系统中，输出会在机器中累积，直到有足够的输出 "值得"显示在屏幕上为止，std::endl强制在那一刻显示任何累积的输出。

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end program
16 }
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.



15.3 简单程序：两个整数相加



命名空间(namespace)

- 将std::放在cout、cin和endl之前。
- std::cout这个符号表明我们使用的是一个名字，在本例中是cout，它属于"命名空间(namespace)"std。
- **现在请记住：**在程序中每次提到cout、cin和endl之前都要包含std::。
- 可以用using语句来避免重复写std::cout, cin, endl

流插入操作符 <<

- 它“知道”如何输出每种类型的数据，不必用%d, %f等
- 在一条语句中使用多个流插入操作符(<<)被称为连接、链式或级联流插入操作。

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; e
16 }
```

```
1 // Fig. 15.3: fig15_03.cpp
2 // inline function that calculates the volume of a cube.
3 #include <iostream>
4 using std::cout;
5 using std::cin;
6 using std::endl;
7
8 // Definition of inline function cube. Definition of function appears
9 // before function is called, so a function prototype is not required.
10 // First line of function definition acts as the prototype.
11 inline double cube( const double side )
12 {
13     return side * side * side; // calculate the cube of side
14 }
15
16 int main()
17 {
18     double sideValue; // stores value entered by user
19 }
```

Fig. 15.3 | inline function that calculates the volume of a cube. (Part 1 of 2.)

15.3 简单程序：两个整数相加



- 如果我要一行输入几个变量呢？
 - 能否连用 >> 符号？

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

int main()
{
    int i,j;
    cin>>i>>j;
    cout<<i<<" "<<j;
}
//输入1 2, 输出1 2
```

输入的时候，变量之间用什么连接？
空格，回车，逗号？



• 使用搜索引擎

- Google, bing

• 写关键词而不是句子

- ✗ C++一行输入几个变量
- ✓ cin 多个变量
- ✓ Cin multiple variable



国内版 国际版

Microsoft Bing cin 多个变量 分隔符

网页 图片 视频 学术 词典 地图

15,100 条结果 时间不限 自适应缩放

[C++中cin的输入分隔符问题及相关 - 人生如梦, 亦如幻 - 博客园](#)
2019-6-10 · 1、C/C++中的类型转换函数(区分分类中的类型转换构造函数): 头文件: C中stdlib.h C++中 cstdlib atof(将字符串转换成浮点型数) atoi(将字符串转换成整型数) atoi(将 https://www.cnblogs.com/loveprogramme/p/11000421.html ▾

[cin输入带有间隔符的多个变量_GuoMengKai-CSDN博客](#)
2019-3-1 · C++中cin的输入分隔符问题及相关 weixin_30693183 的博客 06-10 652 1、C/C++中的类型转换函数(区分分类中的类型转换构造函数): 头文件: C中stdlib.h C++中 cstdlib atof(将字符串转换成浮点型数) atoi(将字符串转换成整型数) atoi(将 https://blog.csdn.net/guomengkai/article/details/87879812 ▾

[在cin输入赋值时, 如何以逗号\(,\)作为分隔符? -CSDN论坛](#)
2005-3-15 · 在 cin >> a >> b >> c; 这样的语句, 为三个变量赋值时, 默认都是以空格为分隔符。请问如何控制, 使得赋值以逗号(,)作为分隔符? 例如赋值形式是: 6,6,8 而不是 6 6 8 https://bbs.csdn.net/topics/70162997 ▾

C语言如何存储输入带逗号的分隔的字符-CSDN论坛 2019-12-8

c - 使用cin读取任意空格的逗号分隔整数对(x,y)-CSDN论坛 2019-9-12

C++输入时以逗号代替空格符-CSDN论坛 2014-3-15

请问C++中怎么实现string的分割 2008-12-1

[查看更多结果](#)

[C++ cin 的详细用法_Dablelv的博客专栏-CSDN博客](#)
2015-9-25 · 代码编译运行环境: VS2012+Win32+Debug。1.cin简介cin是C++编程语言中的标准输入流对象, 即istream类的对象。cin主要用于从标准输入读取数据, 这里的标准输入, 指的是终端的键盘。此外, cout是流的对象, 即ostream类的对象, cerr是标准错误 https://blog.csdn.net/k346k346/article/details/48213811 ▾

[在cin输入赋值时, 如何以逗号\(,\)作为分隔符? -CSDN论坛](#)
2005-3-16 · 在那个C程序中, 变量赋值的语句就是"21cnbao(月影孤鸿)"说的这个方式, 所以我需要在改写的C++语言程序中也控制为以逗号(,)作为分隔符。目前看来"tudou614(魔禁座的SATAN)"提供的方法 https://blog.csdn.net/k346k346/article/details/48213811

15.3 简单程序：两个整数相加



• C++ main函数返回值

- 在这个例子中，在main的结尾没有return 0;语句
- 根据C++标准，如果程序执行到main结束时，没有遇到返回语句，则认为程序终止成功--就像main中最后一条语句是一个值为0的返回语句一样。
- 所以，可以在C++程序中省略main结尾处的返回语句。

```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 int main()
6 {
7     std::cout << "Enter first integer: "; // prompt user for data
8     int number1;
9     std::cin >> number1; // read first integer from user into number1
10
11     std::cout << "Enter second integer: "; // prompt user for data
12     int number2;
13     std::cin >> number2; // read second integer from user into number2
14     int sum = number1 + number2; // add the numbers; store result in sum
15     std::cout << "Sum is " << sum << std::endl; // display sum; end program
16 }
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Fig. 15.1 | Addition program that displays the sum of two numbers.



- 每个C++程序都以 `main` 函数开始
 - `Main` 函数返回错误代码
 - 返回0表示一切OK
 - 如果有错误，可以自定义返回[1, 255]

```
1 int main() {  
2     return 0; // Program finished without errors.  
3 }
```

```
1 int main() {  
2     return 1; // Program finished with error code 1.  
3 }
```

15.3 简单程序：两个整数相加



- 想要C++输出新的东西，或者按自己要求的格式输出？
 - 我们可以“教”C++如何使用>>和<<运算符来输入和输出这些新的数据类型的值（这被称为**运算符重载**--我们将在第7周中探讨这个话题）



程序，搞起来很轻松的
就是头冷



C++ reference

C++98, C++03, C++11, C++14, C++17, C++20		
<ul style="list-style-type: none">Compiler supportFreestanding implementations Language <ul style="list-style-type: none">Basic conceptsC++ keywordsPreprocessorExpressionsDeclarationInitializationFunctionsStatementsClassesTemplatesExceptions Headers <ul style="list-style-type: none">Named requirementsFeature test macros (C++20)Language support library<ul style="list-style-type: none">Type support – traits (C++11)Program utilitiesRelational comparators (C++20)numeric_limits – type_infoinitializer_list (C++11)	<ul style="list-style-type: none">Concepts library (C++20)Diagnostics libraryGeneral utilities library<ul style="list-style-type: none">Smart pointers and allocatorsDate and timeFunction objects – hash (C++11)String conversions (C++17)Utility functionspair – tuple (C++11)optional (C++17) – any (C++17)variant (C++17) – format (C++20)Strings library<ul style="list-style-type: none">basic_stringbasic_string_view (C++17)basic_string_viewNull-terminated strings:<ul style="list-style-type: none">byte – multibyte – wideContainers library<ul style="list-style-type: none">array (C++11) – vectormap – unordered_map (C++11)priority queue – span (C++20)Other containers:<ul style="list-style-type: none">sequence – associativeunordered associative – adaptors	<ul style="list-style-type: none">Iterators libraryRanges library (C++20)Algorithms libraryNumerics library<ul style="list-style-type: none">Common math functionsMathematical special functions (C++17)Numeric algorithmsPseudo-random number generationFloating-point environment (C++11)complex – valarrayInput/output library<ul style="list-style-type: none">Stream-based I/OSynchronized output (C++20)I/O manipulatorsLocalizations libraryRegular expressions library (C++11)<ul style="list-style-type: none">basic_regex – algorithmsAtomic operations library (C++11)<ul style="list-style-type: none">atomic – atomic_flagatomic_ref (C++20)Thread support library (C++11)Filesystem library (C++17)
Technical specifications <ul style="list-style-type: none">Standard library extensions (library fundamentals TS)<ul style="list-style-type: none">resource_adaptor – invocation_typeStandard library extensions v2 (library fundamentals TS v2)<ul style="list-style-type: none">propagate_const – ostream_joiner – randintobserver_ptr – detection idiomStandard library extensions v3 (library fundamentals TS v3)<ul style="list-style-type: none">scope_exit – scope_fail – scope_success – unique_resourceConcurrency library extensions (concurrency TS)Concepts (concepts TS)Ranges (ranges TS)Transactional Memory (TM TS)		
External Links – Non-ANSI/ISO Libraries – Index – std Symbol Index		

<https://en.cppreference.com/w/cpp>



- **C++ Reference:**

- <https://en.cppreference.com/w/cpp>

- **Cpp Core Guidelines:**

- <https://github.com/isocpp/CppCoreGuidelines>

- **Google 开源项目风格指南 (中文版)**

- <https://zh-google-styleguide.readthedocs.io/en/latest/>

- **C++ Tutorial:**

- <http://www.cplusplus.com/doc/tutorial/>





- Python中一个字符串的所有字母用它的后继字母代替（比如，a用b代替，b用c代替），怎么搞？
 - <https://www.zhihu.com/question/491798364>
 - 内有各种“奇葩”的解决方法，包括凯撒加密、神经网络、查表、多项式拟合





感谢



- 本课程PPT部分内容收集自网上公开资料，如有侵权，请邮件联系我们
- 如您是承担了类似课程的教师，需要原始pptx文件，请邮件联系张书航或李同文
- 联系方式在首页