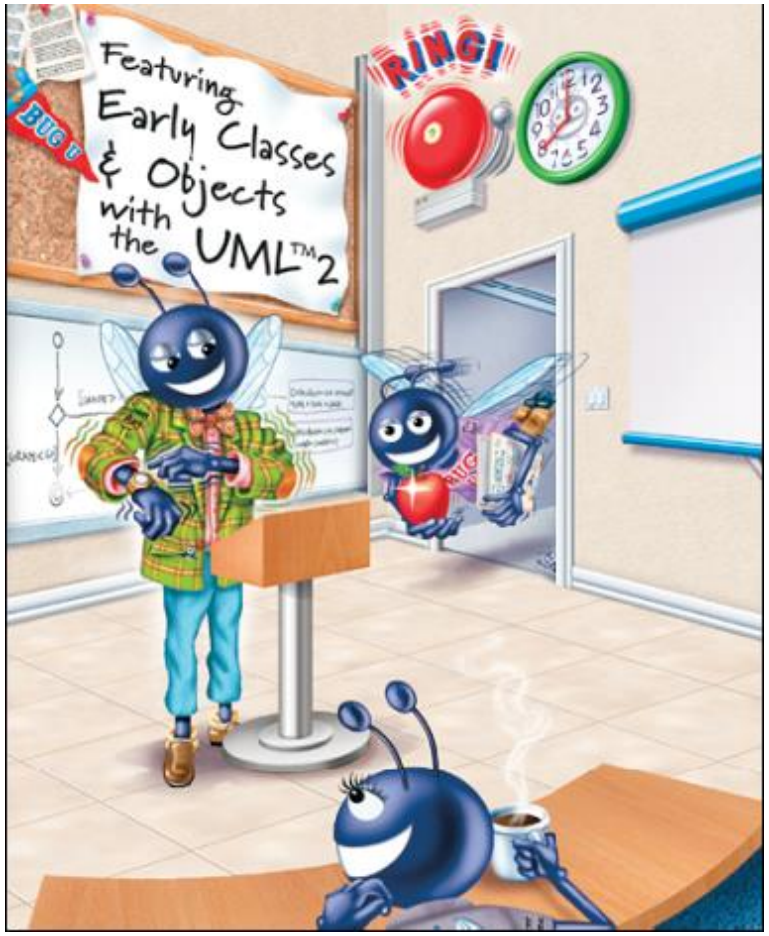
C++程序设计



课程介绍

学习目的：

- ✓ 从“结构化程序设计”转向“面向对象程序设计”
- ✓ 掌握C++程序设计方法；掌握调试程序的基本方法
- ✓ 养成良好的编程习惯

课时安排：

- ✓ 课堂讲授 32学时，实验8学时

考核方法：

- ✓ 期末笔试 (60%)＋实验成绩 (40%)

课程介绍

教材：

- ✓ 《Small C++ How To Program》

学习方法：

- ✓ 实践、实践、再实践

联系方式：

- ✓ 丁建睿：dingjianrui@163.com，研究院北505

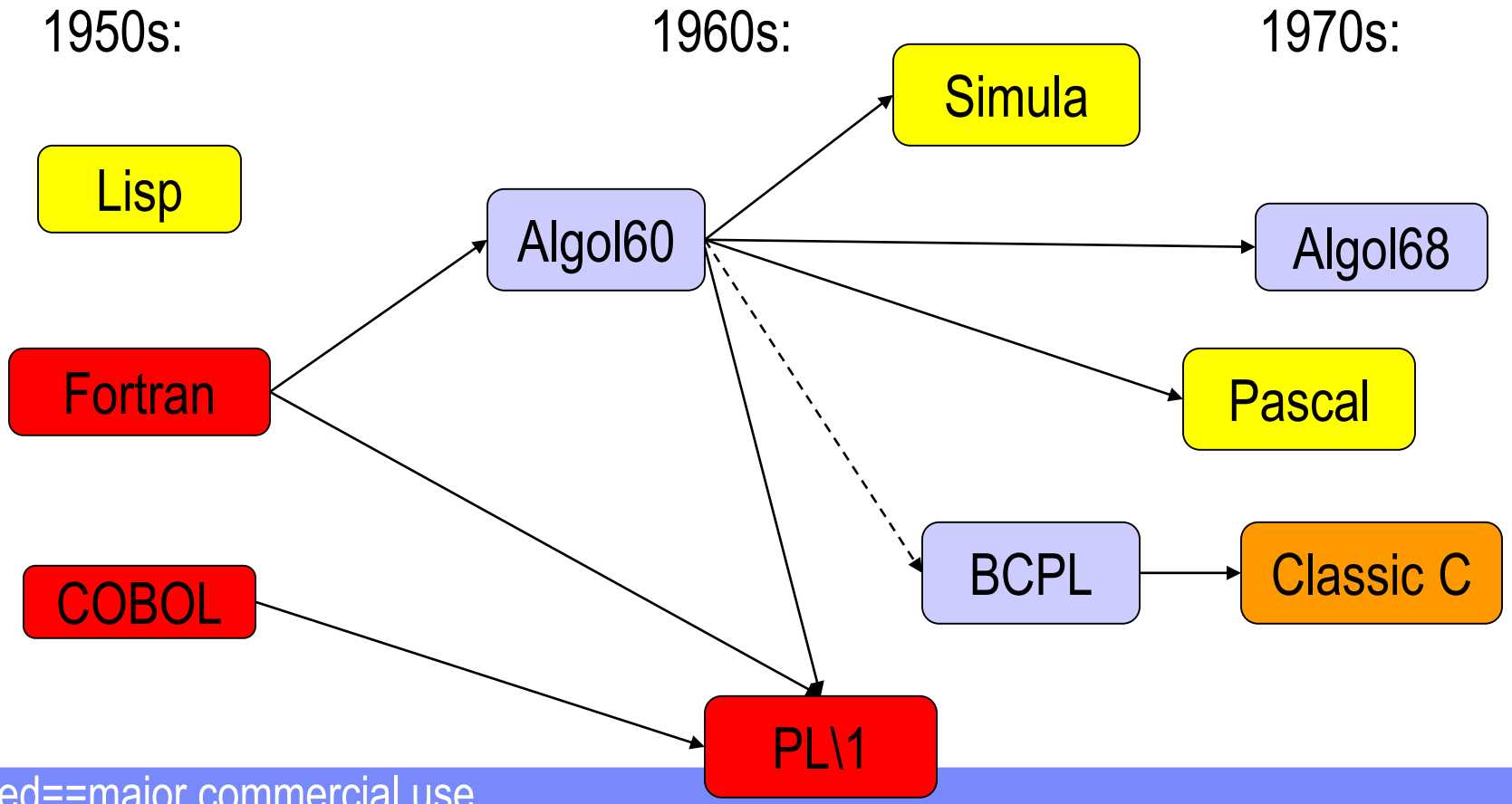
第一讲 C++编程入门

学习目标：

- 程序设计语言
- 理解一个典型的C++程序开发环境
- 使用C++编写简单的计算机程序
- 编写简单的输入输出语句

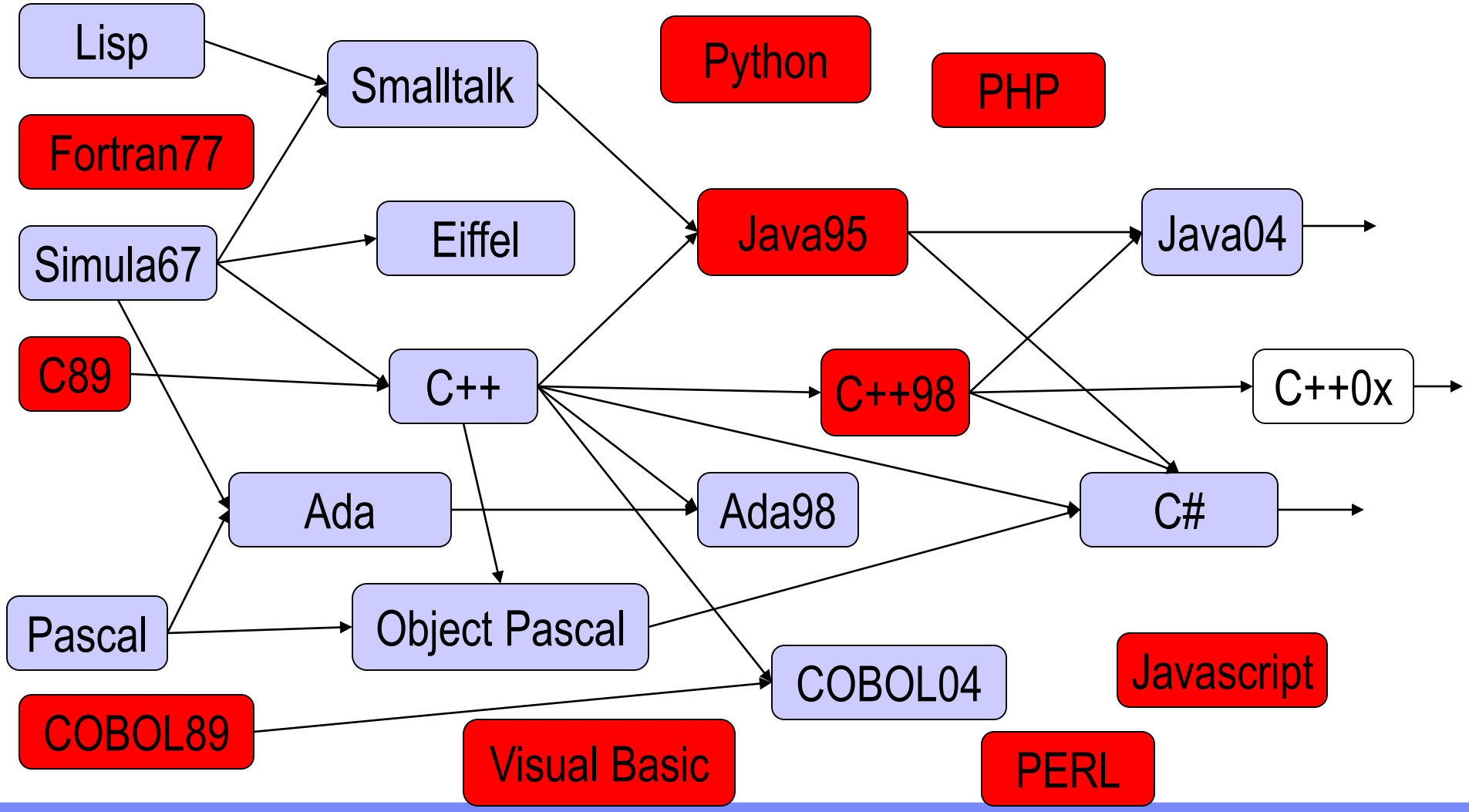
1.1 程序设计语言

◈ 早期程序设计语言



Red==major commercial use
Yellow==will produce important “offspring”

1.1 程序设计语言



1.1 程序设计语言

- 编译、解释、即时编译（Compiled, interpreted, or JIT-compiled）
- 低级、高级（Low or high level）
- 类型系统（Type system）
  - Type Strength: Strong or Weak（不同数据类型之间如何转换）
  - Type Expression: Manifest or Inferred（数据类型如何声明）
  - Type Checking: Static or Dynamic（类型检查在何时进行）
  - Type Safety: Safe or Unsafe（允许在类型变量上进行哪些操作）

1.1 程序设计语言

- 支持的样式（Supported paradigms）
  - 声明式（Declarative）
  - 函数式（Functional）
  - 泛型（Generic）
  - 命令式（Imperative）
  - 结构化（Structured）
  - 过程化（Procedural）
  - 面向对象（Object-Oriented）

1.1 程序设计语言

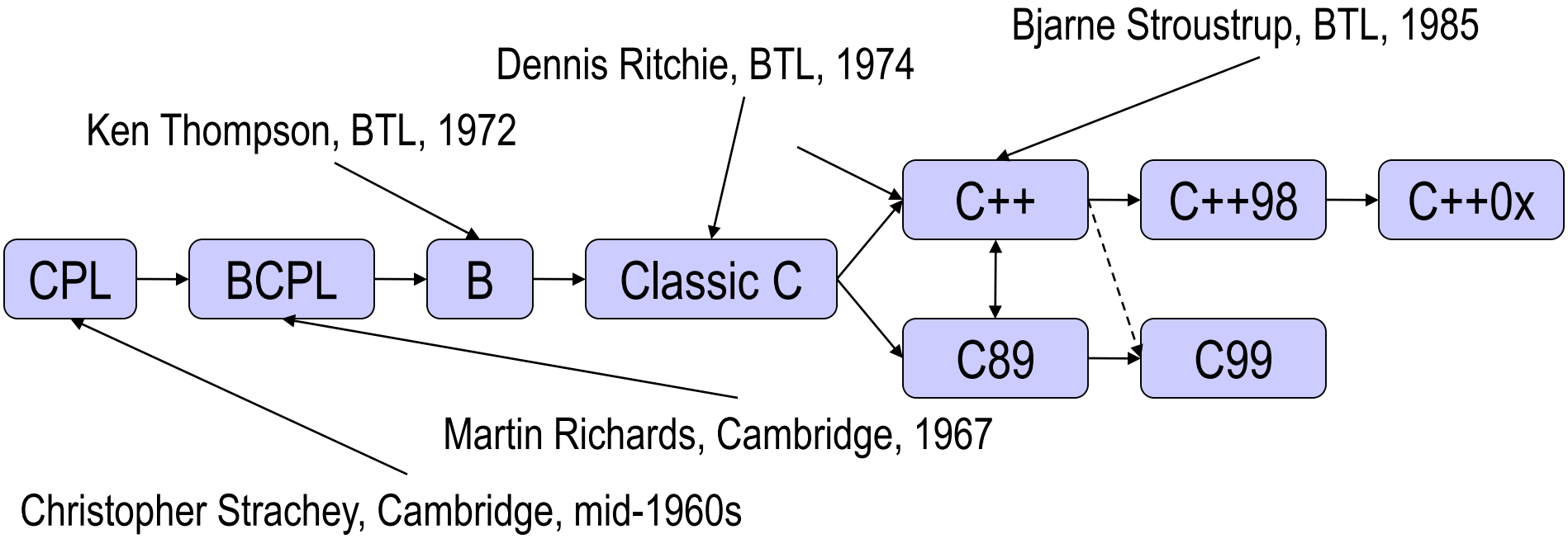
- 标准（Standardization）
  - American National Standards Institute (ANSI)
  - International Organization for Standardization (ISO)
  - 事实上的标准（de-facto standard）
- List of programming languages by type

  https://en.wikipedia.org/wiki/List_of_programming_languages_by_type

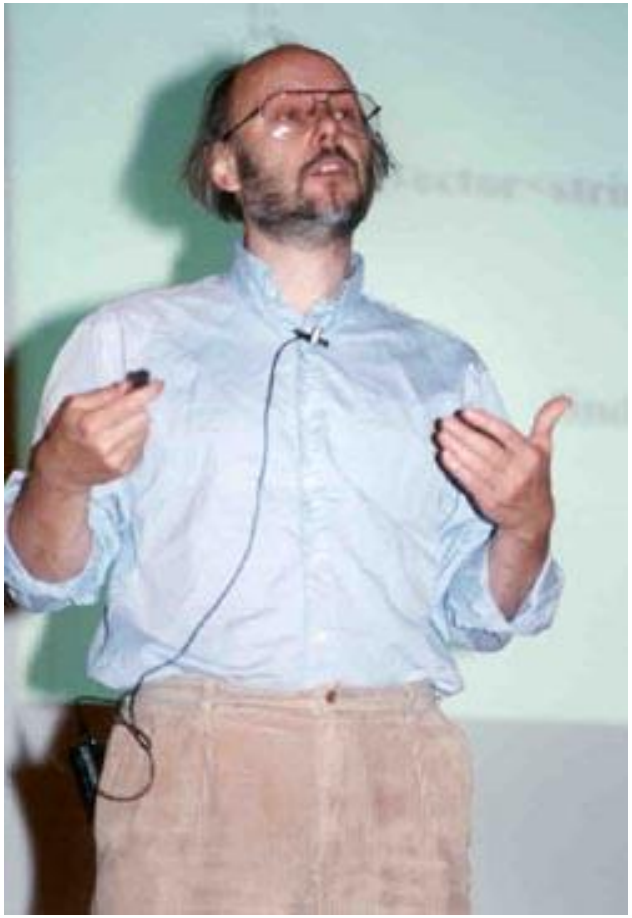
1.1 程序设计语言

- C++的特点
  - an open ISO-standardized language
  - a compiled language
  - a strongly-typed unsafe language
  - supports both manifest and inferred typing
  - supports both static and dynamic type checking
  - offers many paradigm choices
  - is portable
  - is upwards compatible with C
  - has incredible library support

1.2 从 C 到 C++



1.2 从 C 到 C++

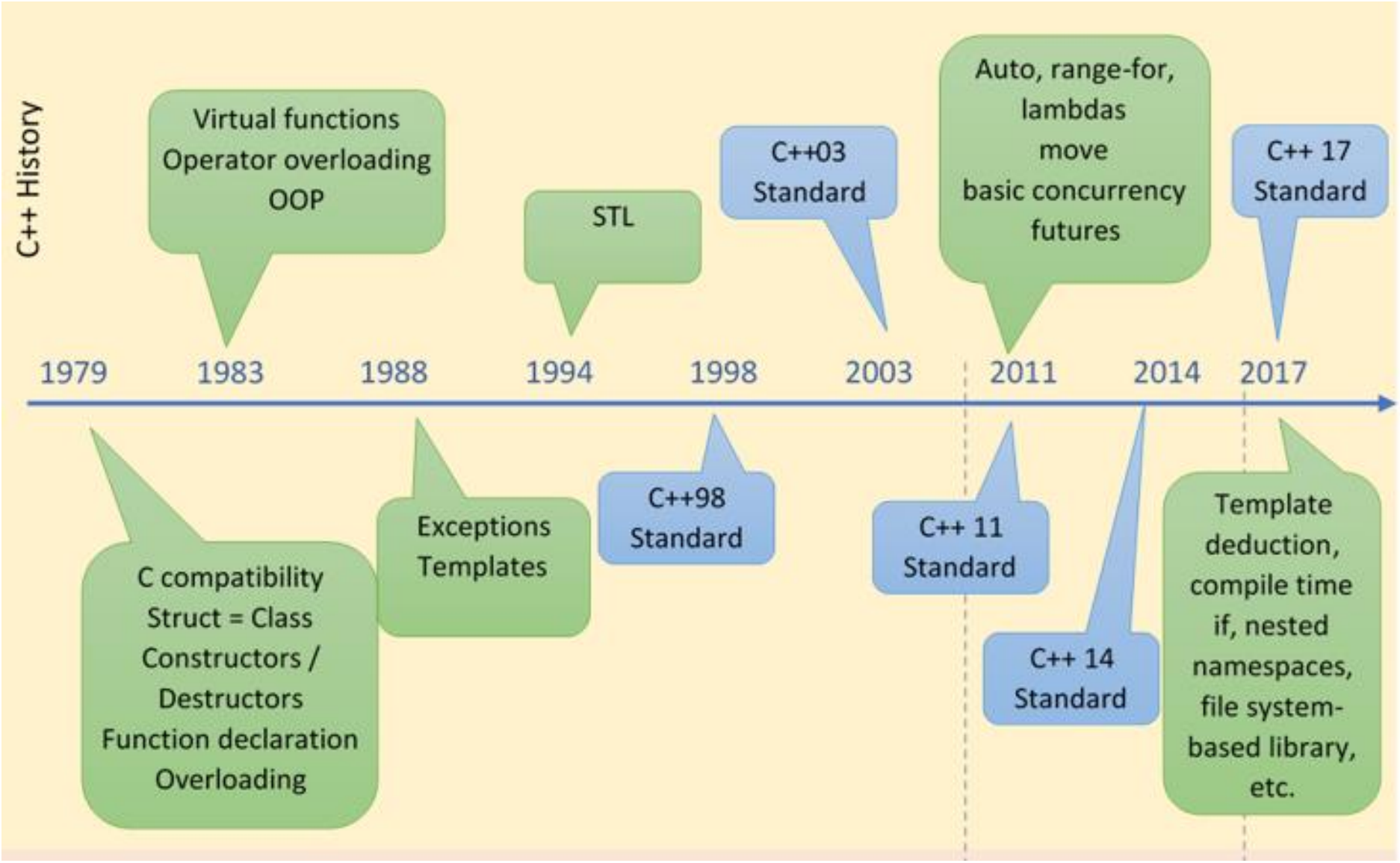


Bjarne Stroustrup（本贾尼·斯特劳斯特卢普）

1.2 从 C 到 C++

History of C++

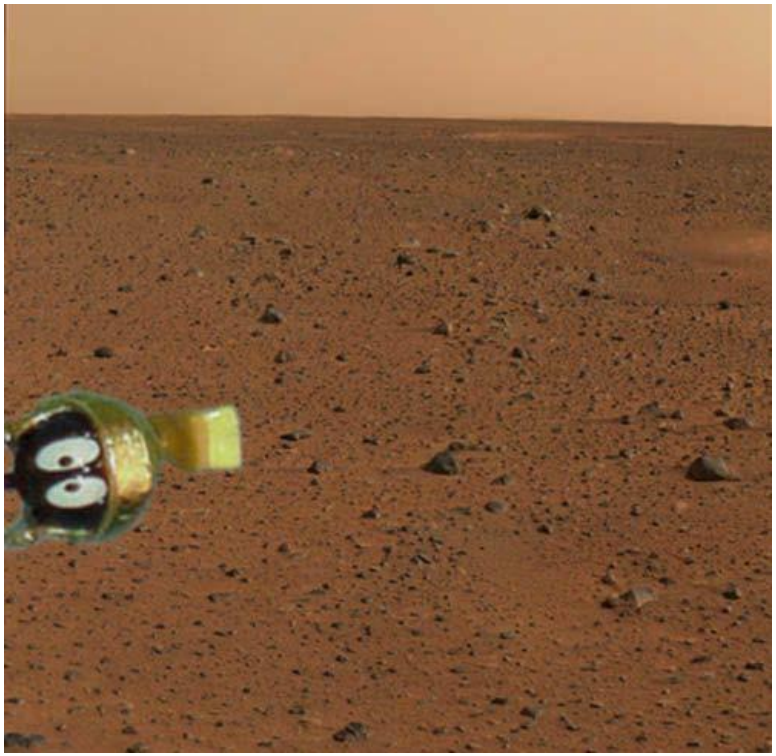
- C++ 由 C 发展而来，并借鉴了 Simula67 and Algol68 的一些特点
- 1980年的第一个版本被命名为 “C with classes”，1983年开始使用 C++
- 1985年发布第一个商用版本
- 1998: ISO/IEC 14882 specifies the standard for C++
- 2005: C++0x
- 2011: C++11



1.2 从 C 到 C++

- C++ is everywhere

  http://www.stroustrup.com/applications.html



1.2 从 C 到 C++

- Uses/Applications of C++ Language
  - Operating Systems
  - Compilers
  - Browsers
  - Libraries
  - Graphics
  - Banking Applications
  - Databases
  - Embedded Systems
  - Telephone Switches

1.2 从 C 到 C++

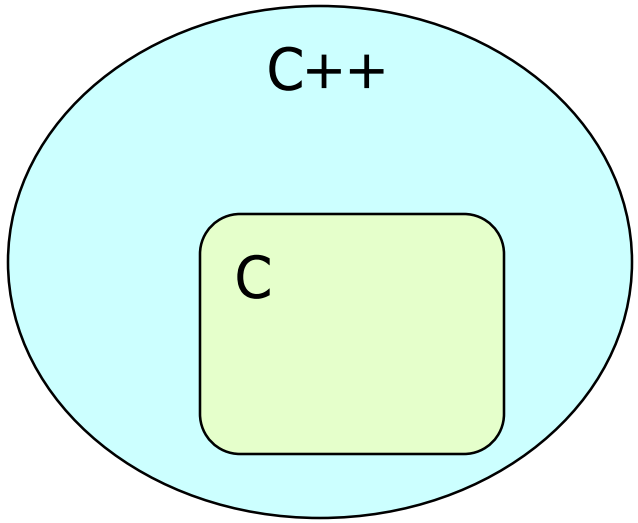
- Top 5 Organizations by traffic

Websites	Popularity (unique visitors per month)[1]	Front-end (Client-side)	Back-end (Server-side)
Google.com[2]	1,600,000,000	JavaScript	C, C++, Go,[3] Java, Python, PHP (HHVM)
Facebook.com	1,100,000,000	JavaScript	Hack, PHP (HHVM), Python, C++, Java, Erlang, D,[6] XHP,[7] Haskell[8]
YouTube.com	1,100,000,000	JavaScript	C, C++, Python, Java,[11] Go[12]
Yahoo	750,000,000	JavaScript	PHP
Amazon.com	500,000,000	JavaScript	Java, C++, Perl[16]

1.2 从 C 到 C++

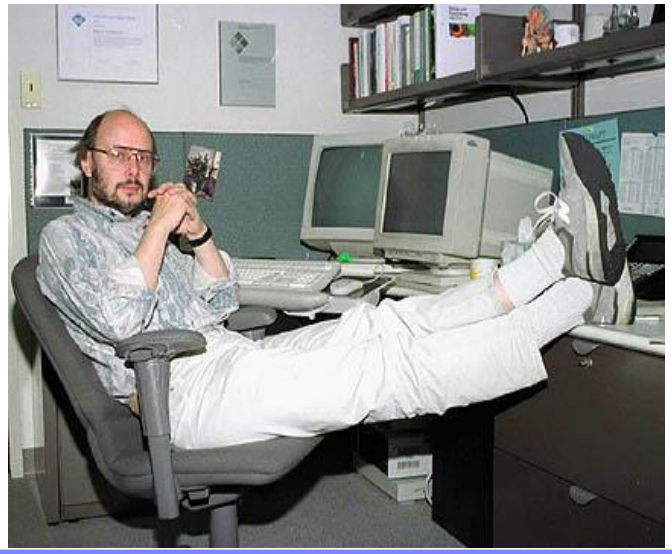
- C++是C的超集(Superset)
  Provides capabilities for object-oriented programming

- 一种混合语言(Hybrid language)
  Can program with, C-like style, Object-oriented style, or both



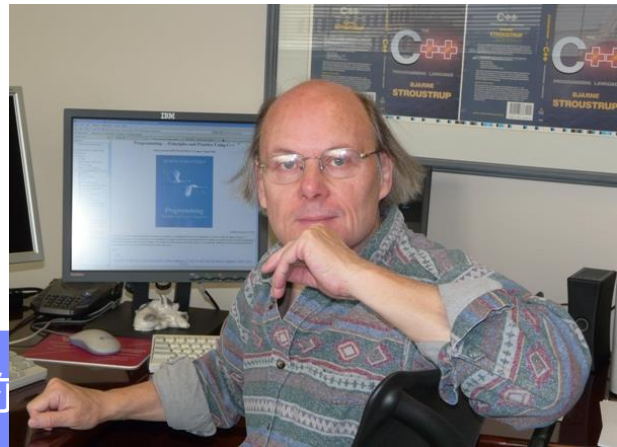
1.2 从 C 到 C++

- C++ programs
  Built from pieces called classes and functions(类和函数)
- C++ standard library
  Rich collections of existing classes and functions



Bjarne Stroustrup 对 C++ 初学者的3条建议

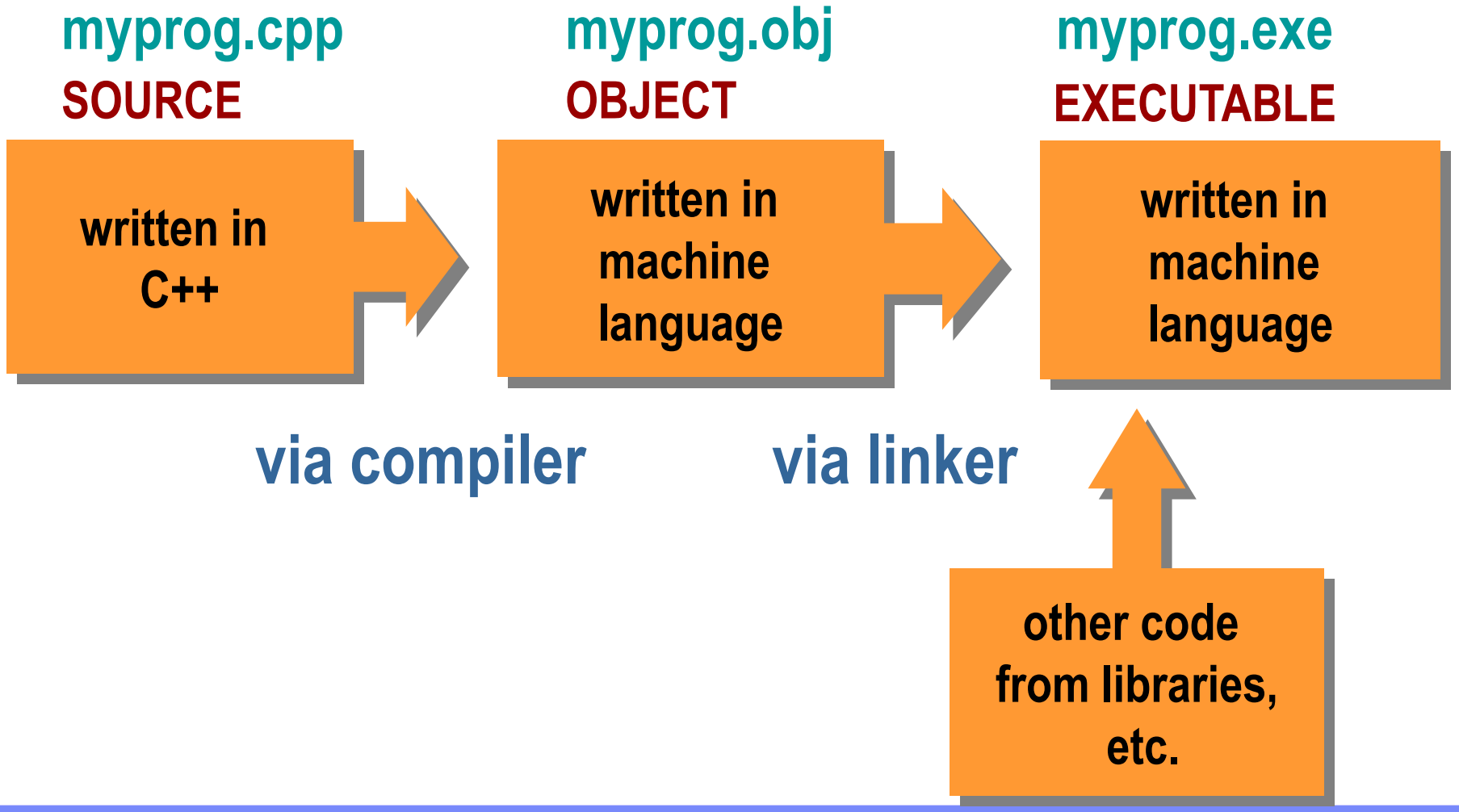
1. Don't panic! All will become clear in time;
2. You don't have to know every detail of C++ to write good programs;
3. Focus on programming techniques, not on language features.



C++ 学习资源

- http://www.cplusplus.com/
- http://www.tutorialspoint.com/cplusplus/cpp_overview.htm

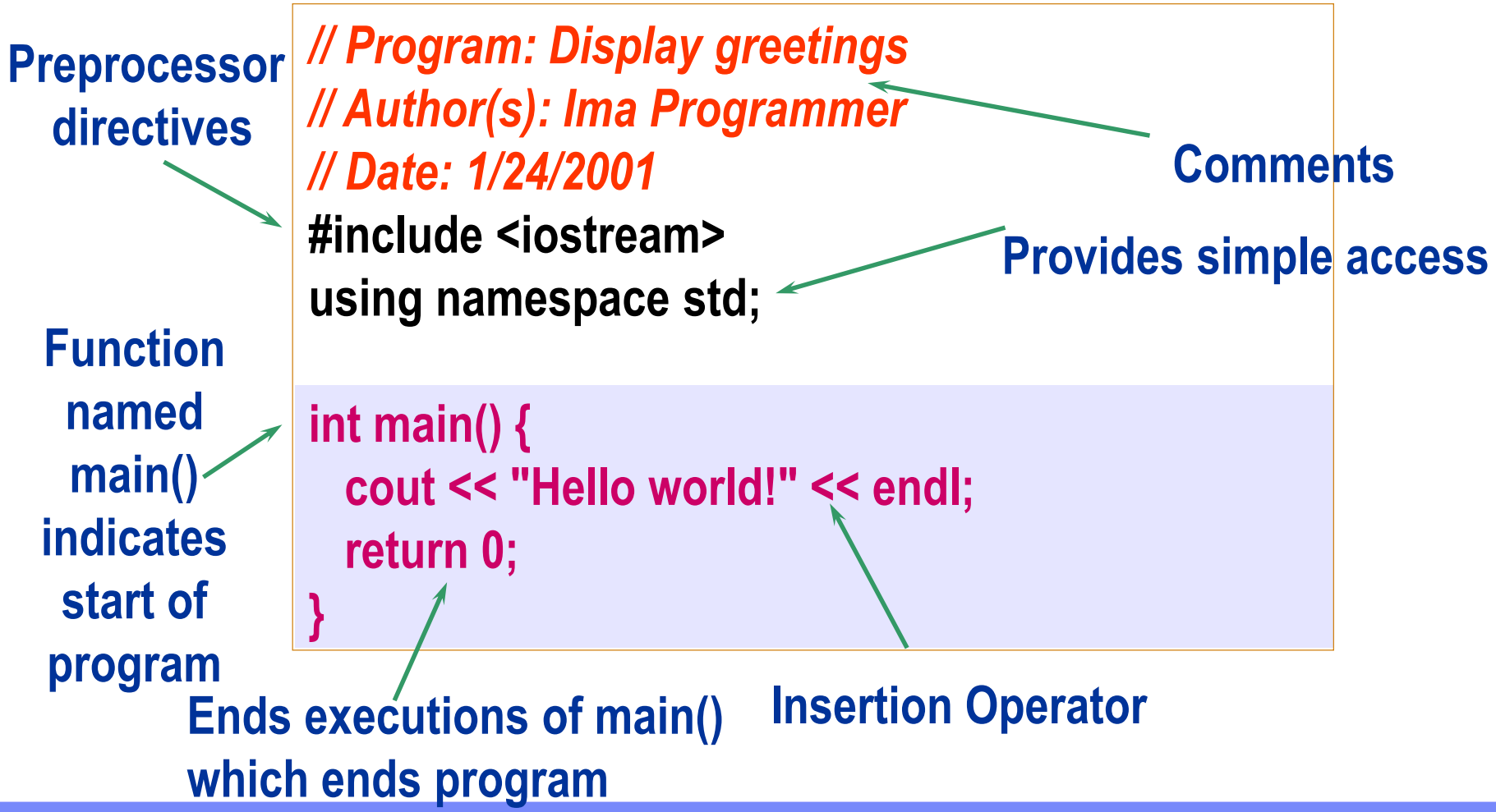
2. C++的基本开发环境



3. A First Program - Greeting.cpp

C++ Implementation	Source Code Extension(s)
Unix	C, cc, cxx, c
GNU C++	C, cc, cxx, cpp, c++
Digital Mars	cpp, cxx
Borland C++	cpp
Watcom	cpp
Microsoft Visual C++	cpp, cxx, cc
Freestyle CodeWarrior	cpp, cp, cc, cxx, c++

3. A First Program - Greeting.cpp



3. A First Program - Greeting.cpp

几种不同的include方法：

1. #include <iostream.h>

2. #include <iostream>
   using namespace std;

3. #include <iostream>
   using std::cout;

4. #include <iostream>
   std::cout <<…;

3. A First Program - Greeting.cpp

Kind of Header	Convention	Example	Comments
C++ old style	Ends in `.h`	`iostream.h`	Usable by C++ programs
C old style	Ends in `.h`	`math.h`	Usable by C and C++ programs
C++ new style	No extension	`iostream`	Usable by C++ programs, uses `namespace std`
Converted C	`c` prefix, no extension	`cmath`	Usable by C++ programs, might use non-C features, such as `namespace std`

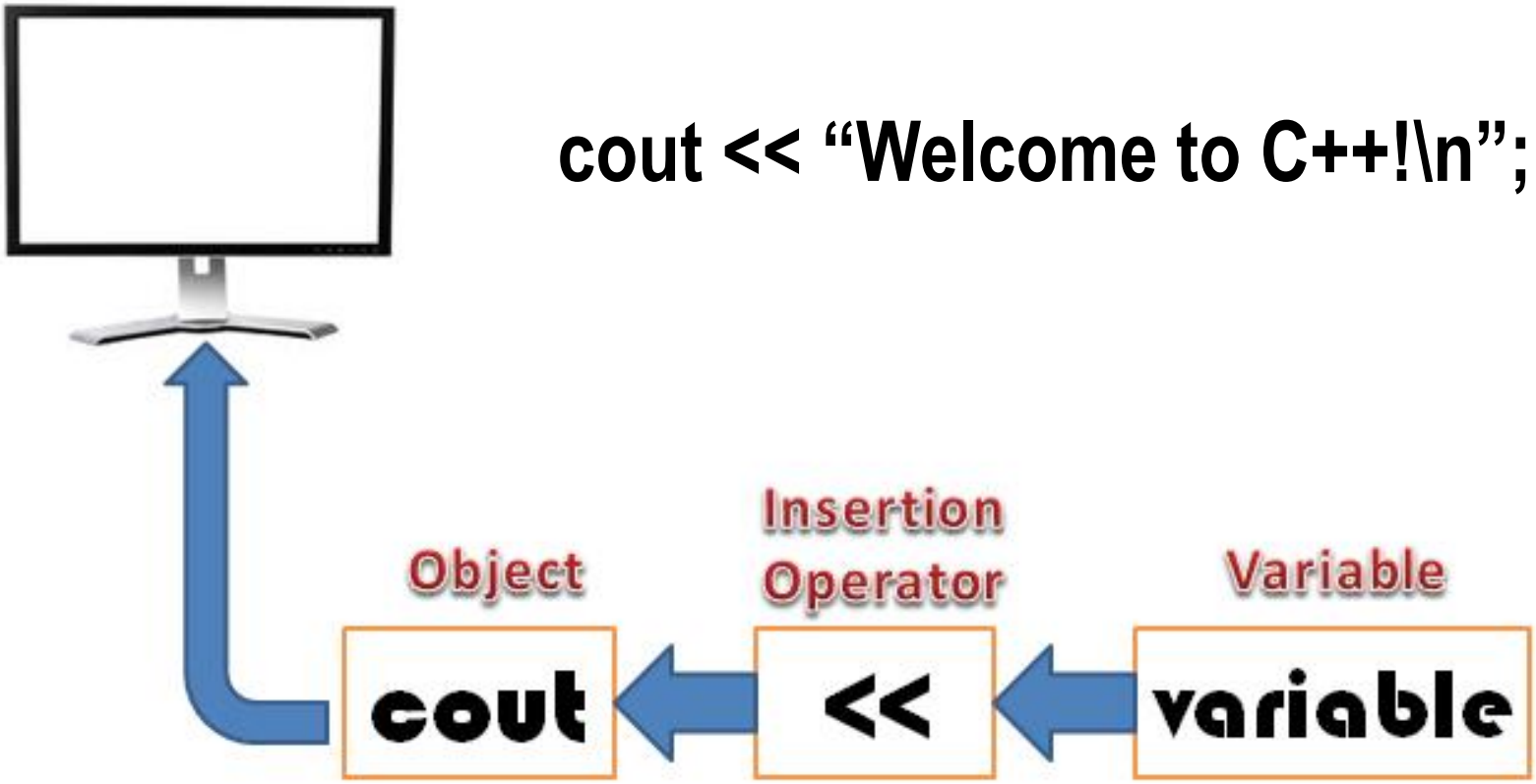
3. A First Program - Greeting.cpp

- std::cout
  - 标准输出流对象
  - 缺省为屏幕
  - std:: 声明 cout 所属的命名空间
    - ✓ 当使用using语句时，std::可以省略

3. A First Program - Greeting.cpp

- <<
  - 流“插入”运算符
  - 将操作符右侧的值“插入”到输出流对象中
  - std::cout << “Welcome to C++!\n”;

3. A First Program - Greeting.cpp



3. A First Program - Greeting.cpp

- \
  - Escape character (转义字符)
  - 指示特殊字符将被输出

3. A First Program - Greeting.cpp

```
#include <iostream>
using namespace std;
int main()
{
    cout << 'A' << ' ' << 'a' << endl;
    cout << "one\ttwo\tthree\n";
    cout << "123\b\b45\n";
    cout << "Alert\a\n";
    return 0;
}
```

4. Another C++ Program: Adding Integers

- 变量
  - 在内存中存储
  - 基本数据类型
    - int – integer numbers
    - char – characters
    - double – floating point numbers
  - 在使用前要为变量命名并声明数据类型
    - int integer1;
    - int integer2;
    - int sum;

```
1   // Fig. 2.5: fig02_05.cpp
2   // Addition program that displays the sum of two numbers.
3   #include <iostream> // allows program to perform input and output
4
5   // function main begins program execution
6   int main()
7   {
8      // variable declarations
9      int number1; // first integer to add
10     int number2; // second integer to add
11     int sum; // sum of num
12
13     std::cout << "Enter fir
14     std::cin >> number1; // read first integer from user into number1
15
16     std::cout << "Enter second integer: "; // prompt user for data
17     std::cin >> number2; // read second integer from user into number2
18
19     sum = number1 + number2; // add the numbers; store resul
20
21     std::cout << "Sum is " << sum << std::endl; // display sum;
22
23     return 0; // indicate that program ended successfully
24
25  } // end function main
```

声明整型变量

使用标准输入对象的流抽取运算符获得用户输入

流操作 std::endl 输出新行, 并清空输出缓冲区

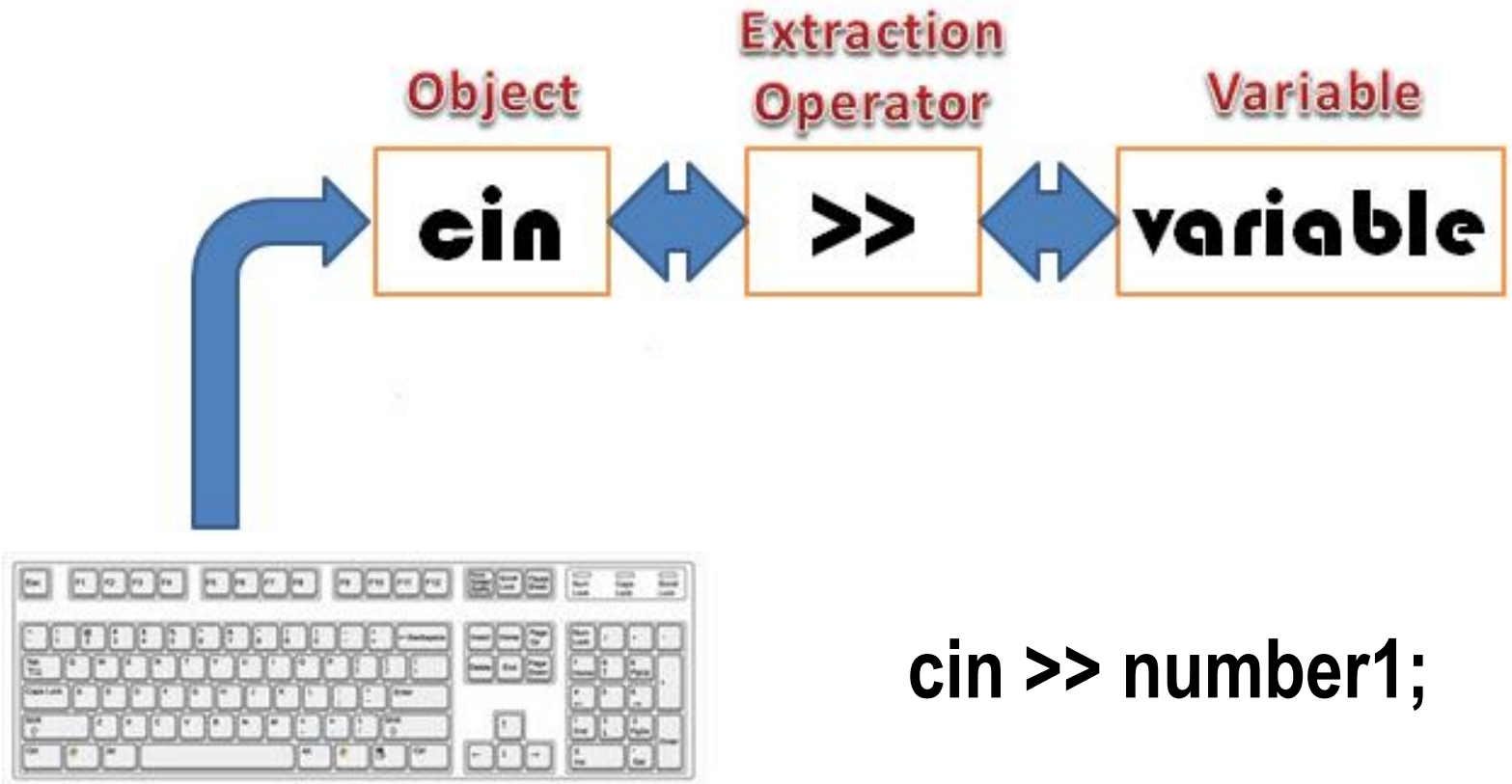
级联的流插入操作

4. Another C++ Program: Adding Integers

良好编程习惯：选择有意义的变量名，将有助于保障程序的“自编档能力”。

良好编程习惯：避免使用以下划线和双下划线开头的标识符，因为C++编译器可能采用这种形式的名称作为内部使用。

4. Another C++ Program: Adding Integers




```
cin >> number1;
```

4. Another C++ Program: Adding Integers

- Concatenating(连接) 流插入运算
  - ◈ 在一条语句中使用多个流插入运算
    - ◈流插入运算能够处理不同的数据类型
  - ◈ 也称为： chaining or cascading

```
std::cout << "Sum is " << number1 + number2 << std::endl;
```

5. Arithmetic

- 算术运算符
  - * - Multiplication
  - / - Division
    - Integer division truncates (截断) remainder (余数)
      - 7 / 5 evaluates to 1
  - % - Modulus operator returns remainder
    - 7 % 5 evaluates to 2

5. Arithmetic

- **算术运算符的优先级**

优先级	运算符	结合性
高 ↑ 低	后置 ++ 后置 --	左→右
	前置 ++ 前置 --	右→左
	* / %	左→右
	+ –	左→右

5. Arithmetic

- 算术运算符的优先级

Algebra: $z = pr\%q + w/x - y$

C++:

```
z = p * r % q + w / x - y;
```

6 1 2 4 3 5

5. Arithmetic

常见编程错误：试图对非整数操作数使用求模运算符%，是一种语法错误。

良好编程习惯：与代数运算一样，可在表达式中加上多余的括号，使其更清晰。这些括号叫做冗余括号。将一条大型语句分割为一系列较短的、较简单的语句，也可以使程序更清晰。

6. Equality and Relational Operators

Standard algebraic equality or relational operator	C++ equality or relational operator	Sample C++ condition	Meaning of C++ condition
Relational operators			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y
Equality operators			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y

6. Equality and Relational Operators

常见编程错误：如果将“==”同赋值运算符“=”混为一谈，会导致逻辑错误或语法错误。

良好编程习惯：较长的语句可分割成几行，如果必须这样分割一条语句，请挑选最合适的断点。比如对一个用逗号分割的列表来说，可选择在某个逗号之后断开；对于较长的表达式，可考虑在一个运算符之后断开，等等。一个语句分割成多行后，除第一行之外，其他所有行都进行缩进处理。

7. Logical Operators

运算符	作用	应用语法
!	逻辑非。如果操作数的值为true，运算结果为false，反之运算结果为true。	!expres
&&	逻辑与。当两个操作数均为true是，运算结果为true。只要有一个操作数为false，运算结果就是false。	expres1 && expres2
||	逻辑或。当两个操作数均为falsee是，运算结果为false。只要有一个操作数为true，运算结果就是true。	expres1 || expres2

7. Logical Operators

执行下列代码段后指定变量的值：

```
bool x = true, y = false, z = false;

a = x && y || z;
b = x || y && z;
c = !(x != y) || (y == z);
```

则 a = ?, b = ?, c = ?

8. sizeof 运算符

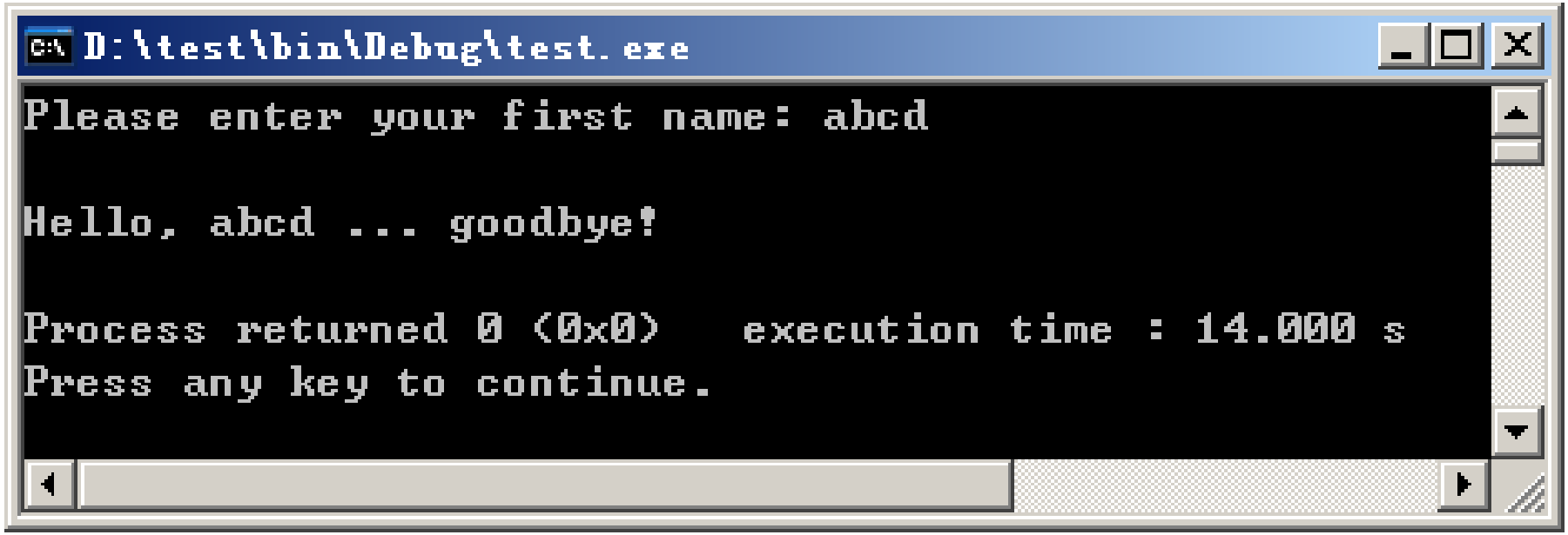
- sizeof 功能
  - 返回一个对象或类型的字节长度。
- 两种用法
  - sizeof (类型名称);
  - sizeof (对象)

9. 位运算

运算符	作用	使用语法
~	按位取非	~expres
<<	左移，将操作数的各位依次左移，右端补0	expres1 << expres2
>>	右移，将操作数的各位依次右移，有符号数的右端补原来的符号位值，无符号数的右端补0	expres1 >> expres2
&	按位与	expres1 & expres2
^	按位异或	expres1 ^ expres2
|	按位或	expres1 | expres2

练习

- 提示用户输入姓名，并在屏幕上打印。



思考题：

P_{54} 页：2.28

要求输入一个5位整数，分解出它的每位数字，每个数字间隔3个空格进行打印。