



C++

第一讲 C++概述

基础课教研室C++ 课程组



本讲教学目标

- 了解C++的起源
- 了解C++的应用领域
- 理解对象和类的基本概念
- 理解面向对象的三大特征
- 安装VS集成开发环境

1

课程简介

2

C++概述

3

C与C++的关系

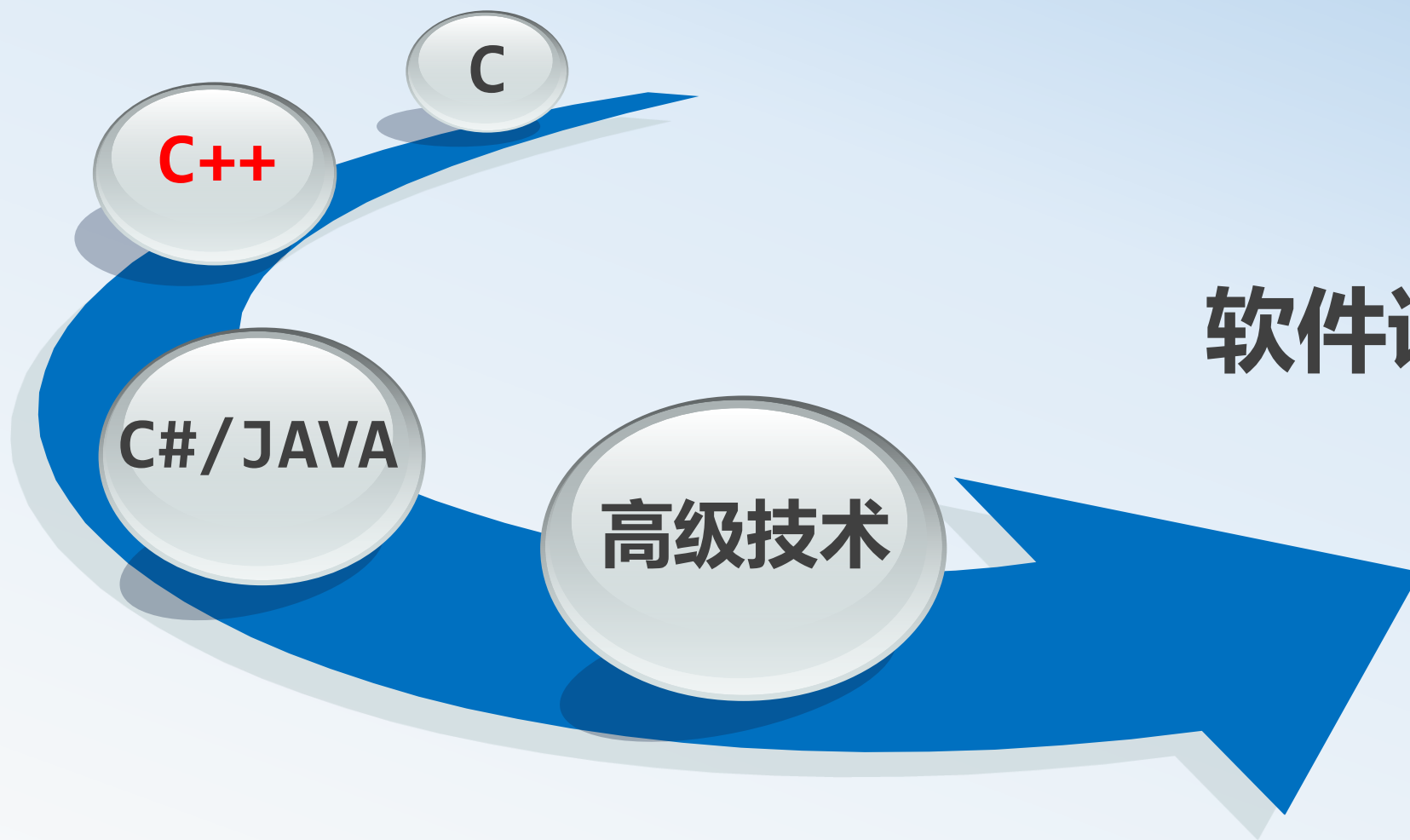
4

面向对象程序设计

5

第一个C++程序

课程简介：课程地位



软件课程体系

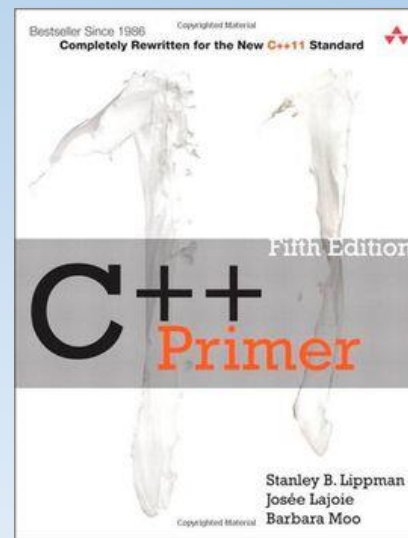
课程简介：参考书籍

课程教材：

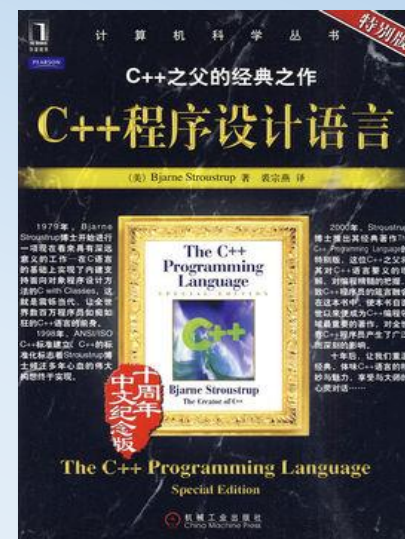


参考资料：

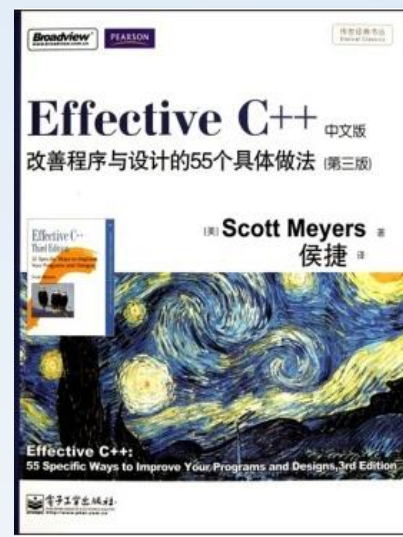
➤ C++ Primer



➤ C++程序设计语言



➤ Effective C++



课程简介

引入

- 面向对象编程基本概念、新的IO、函数、引用

(6)

抽象机制

- **类、继承、多态**

(34)

模板

- 函数模板、类模板

(4)

标准库

- STL

(6)

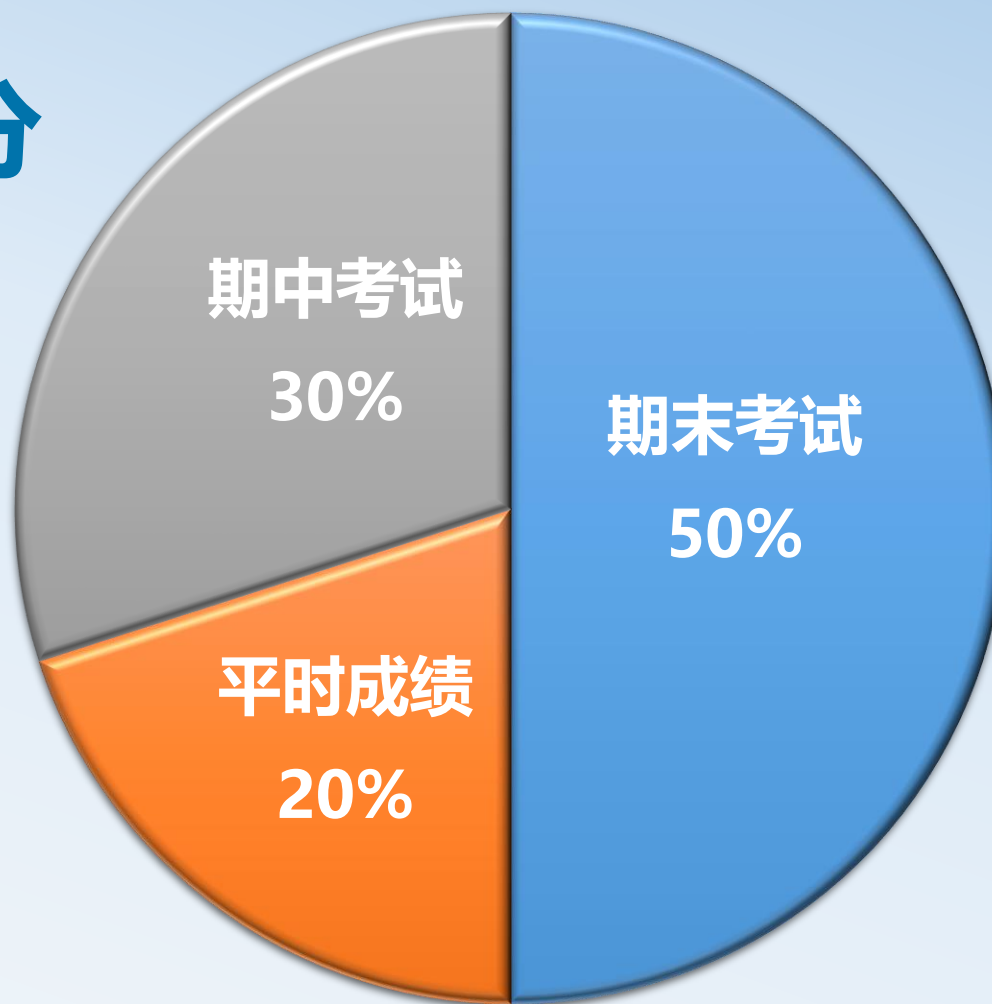
其它

- IO、异常处理、设计模式

(20)

课程简介

必修 72学时 3学分



课程简介：上课要求

- ❖ 讲课的时候不需要电脑
- ❖ 积极听讲，及时回答问题
- ❖ 作业要按时交
- ❖ 有问题要随时问

1

课程简介

2

C++ 概述

3

C与C++的关系

4

面向对象程序设计

5

第一个C++程序

C++概述：C++之父



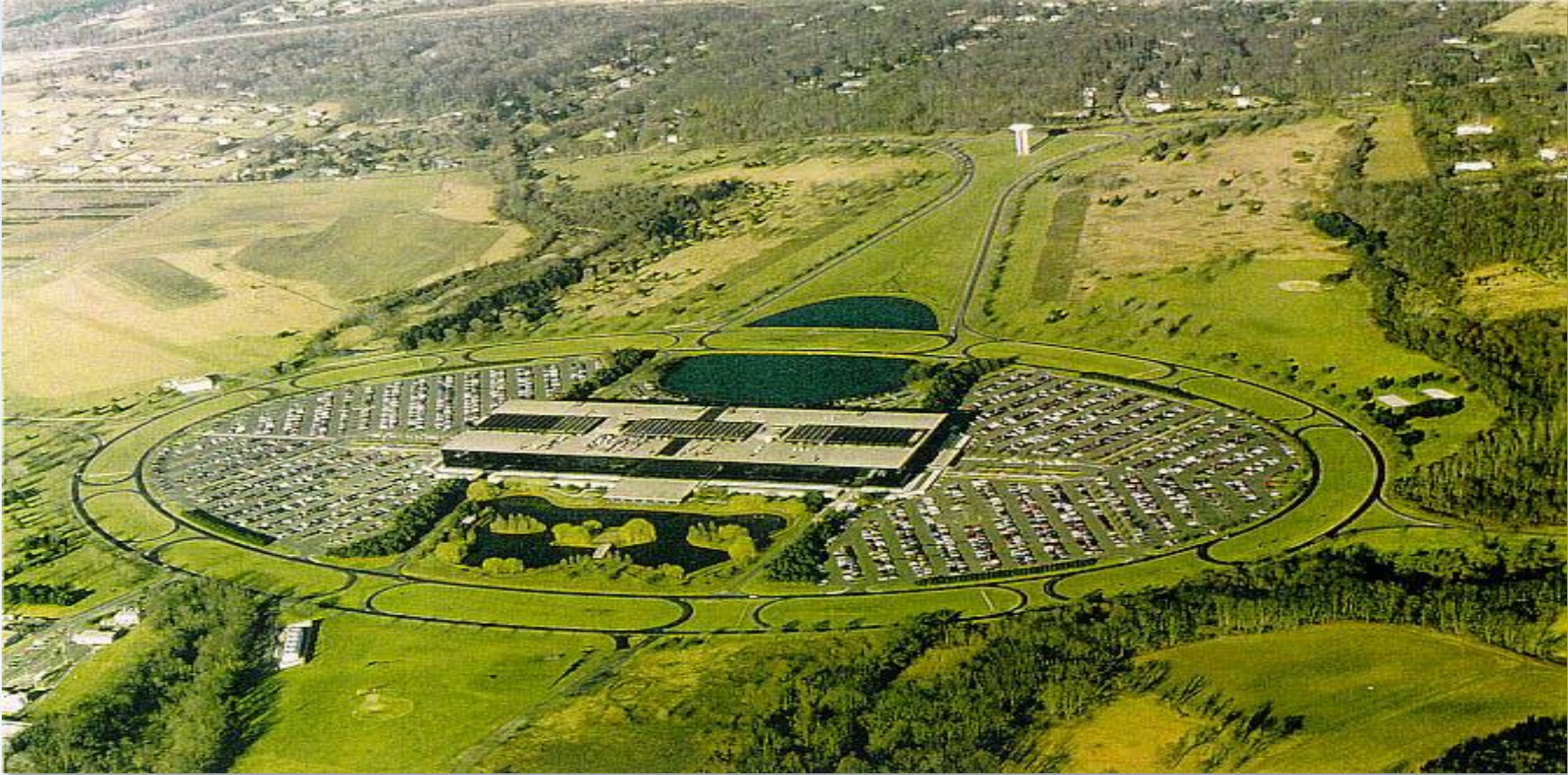
比雅尼·斯特劳斯特鲁普博士，1950年出生于丹麦，先后毕业于丹麦阿鲁斯大学和英国剑桥大学，AT&T大规模程序设计研究部门负责人，AT&T、贝尔实验室和ACM成员。1979年，B. S开始开发一种语言，当时称为“C with Class”，后来演化为C++。1998年，ANSI/ISO C++标准建立，同年，B. S推出了其经典著作The C++ Programming Language的第三版

C++概述：What is C++?

❖ C++是一种通用程序设计语言，特别是面向系统程序设计。

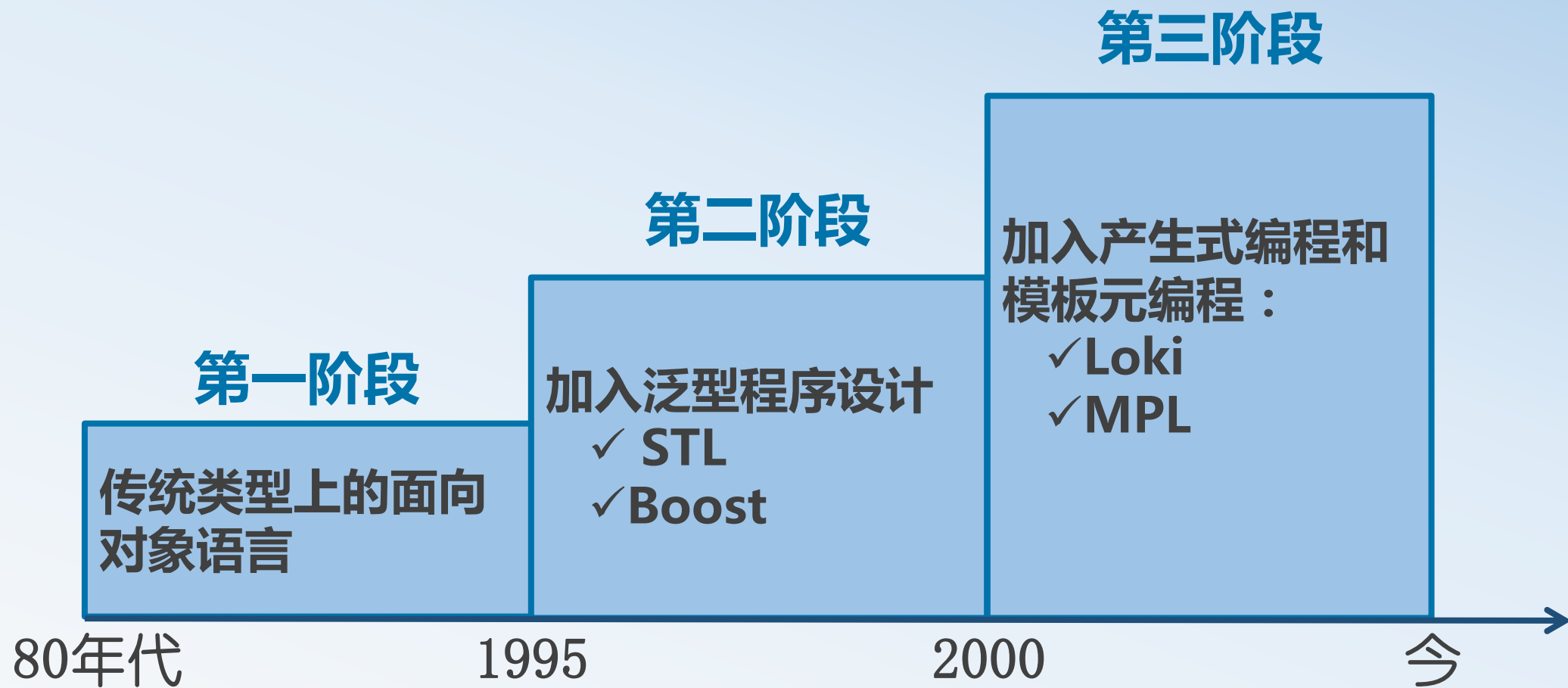
- 是一个更好的C
- 支持数据抽象
- 支持面向对象程序设计
- 支持泛型程序设计

C++概述：Where is C++ from?



C++诞生地----- 贝尔实验室 (Bell Lab)

C++概述：发展史——三个阶段

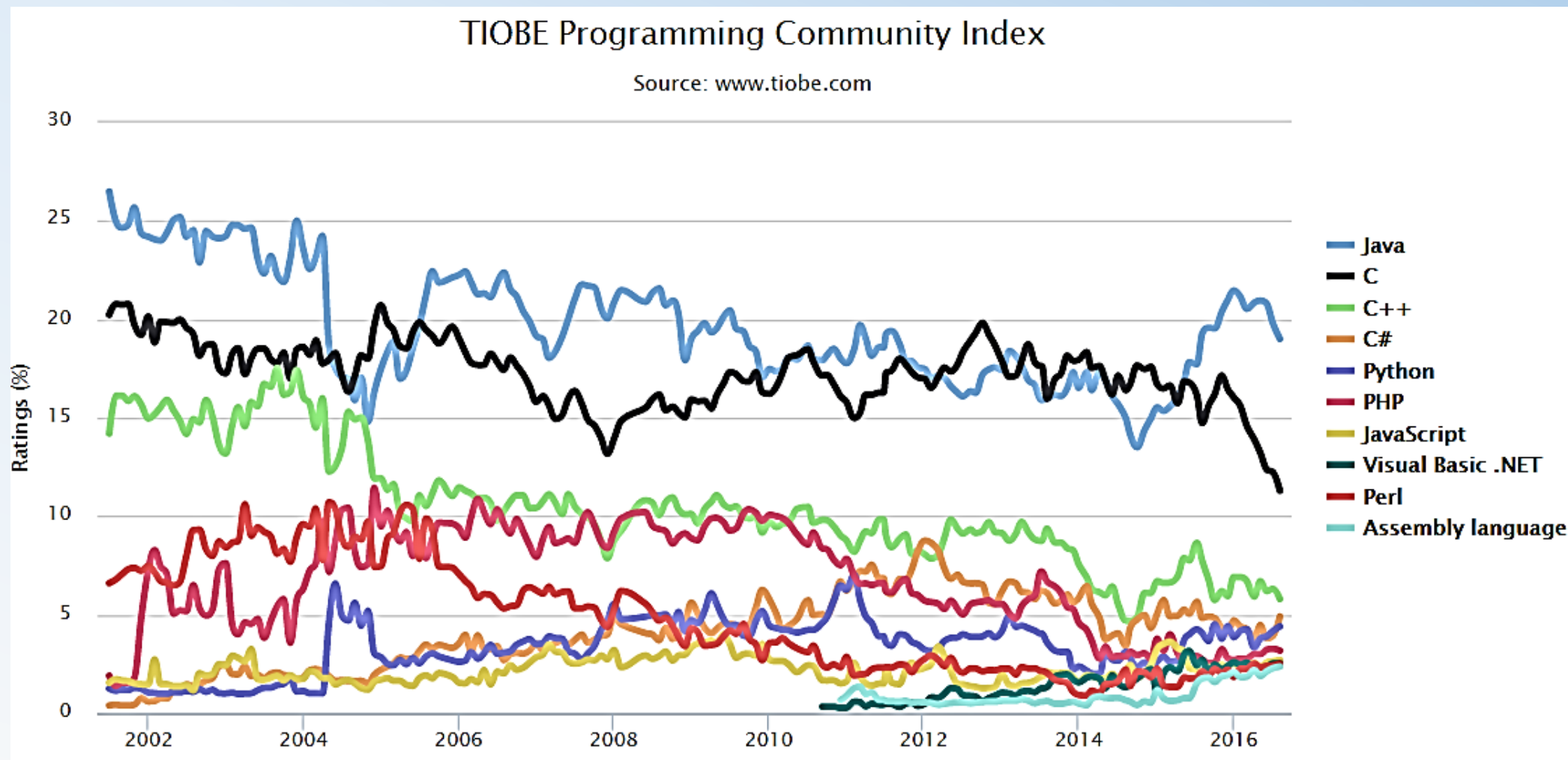


C++概述：行业地位

Aug 2016	Aug 2015	Change	Programming Language	Ratings	Change
1	1		Java	19.010%	-0.26%
2	2		C	11.303%	-3.43%
3	3		C++	5.800%	-1.94%
4	4		C#	4.907%	+0.07%
5	5		Python	4.404%	+0.34%
6	7	^	PHP	3.173%	+0.44%
7	9	^	JavaScript	2.705%	+0.54%
8	8		Visual Basic .NET	2.518%	-0.19%
9	10	^	Perl	2.511%	+0.39%

<http://www.tiobe.com/tiobe-index/>

C++概述：行业地位



<http://www.tiobe.com/tiobe-index/>

C++概述：应用领域



C++概述：应用领域

- ❖ 对效率要求高的行业和领域
- ❖ 要求跨平台应用
- ❖ 底层开发和系统级编程
- ❖ 科学计算领域
- ❖ 分布式系统

1

课程简介

2

C++ 概述

3

C与C++的关系

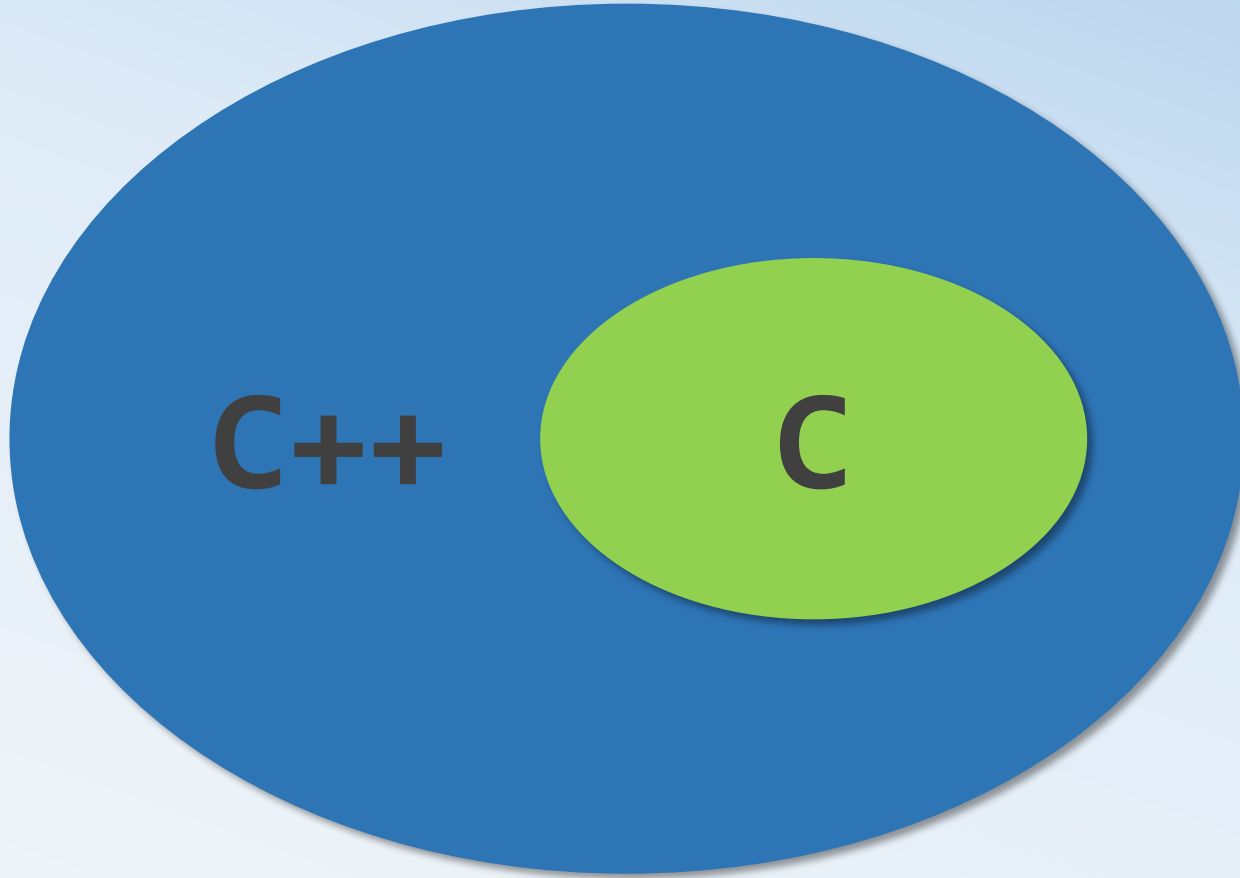
4

面向对象程序设计

5

第一个C++程序

C与C++的关系



C是C++的子集



C与C++的关系

❖ C++是从C的基础上发展而来的

+：新的数据类型和语法改进

+：支持面向对象程序设计

+：支持泛型程序设计

❖ C++既支持面向对象也支持面向过程

❖ 不要把**面向对象**与面向过程对立起来

1

课程简介

2

C++ 概述

3

C与C++的关系

4

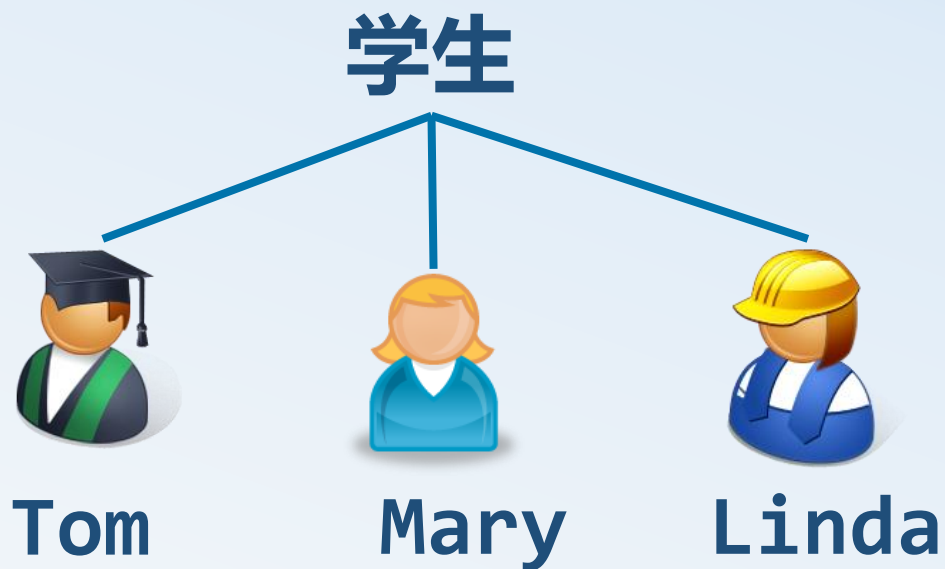
面向对象程序设计

5

第一个C++程序

面向对象程序设计：类与对象

- ❖ **对象**：现实世界中某个具体的**物理实体**，每个对象都包含特定的**属性**和**行为**。
- ❖ **类**：现实世界中具有相同属性和行为的物理实体的抽象。



面向对象程序设计：类与对象

❖ **类是对对象的抽象**，是创建对象的**蓝图**，描述了所创建的对象共同的**属性和行为**

学生：

属性：

姓名
年龄
性别
年级

行为：

学习

轿车：

属性：

颜色
品牌
重量
排量

行为：

行驶

猫：

属性：

名字
颜色
品种

行为：

睡觉
捉老鼠

面向对象程序设计：类与对象

```
struct Student
{
    int num;
    char * name; // 成员变量（属性）
    char sex;
    void study(); // 成员函数（行为）
};
```

❖ **类**：通常即包含数据成员和成员函数的**抽象数据类型**，如结构体。

面向对象程序设计：类与对象

❖对象是类的实例

定义**小强**是一个学生

属性：

小强.姓名: 小强

小强.年龄: 06岁

小强.性别: 男

小强.年级: 学前班

行为：

小强.学习

定义**Tom**是一只猫

属性：

Tom.名字: Tom

Tom.颜色: 黑色

Tom.品种: 波斯猫

行为：

Tom.睡觉

Tom.捉老鼠

面向对象程序设计：类与对象

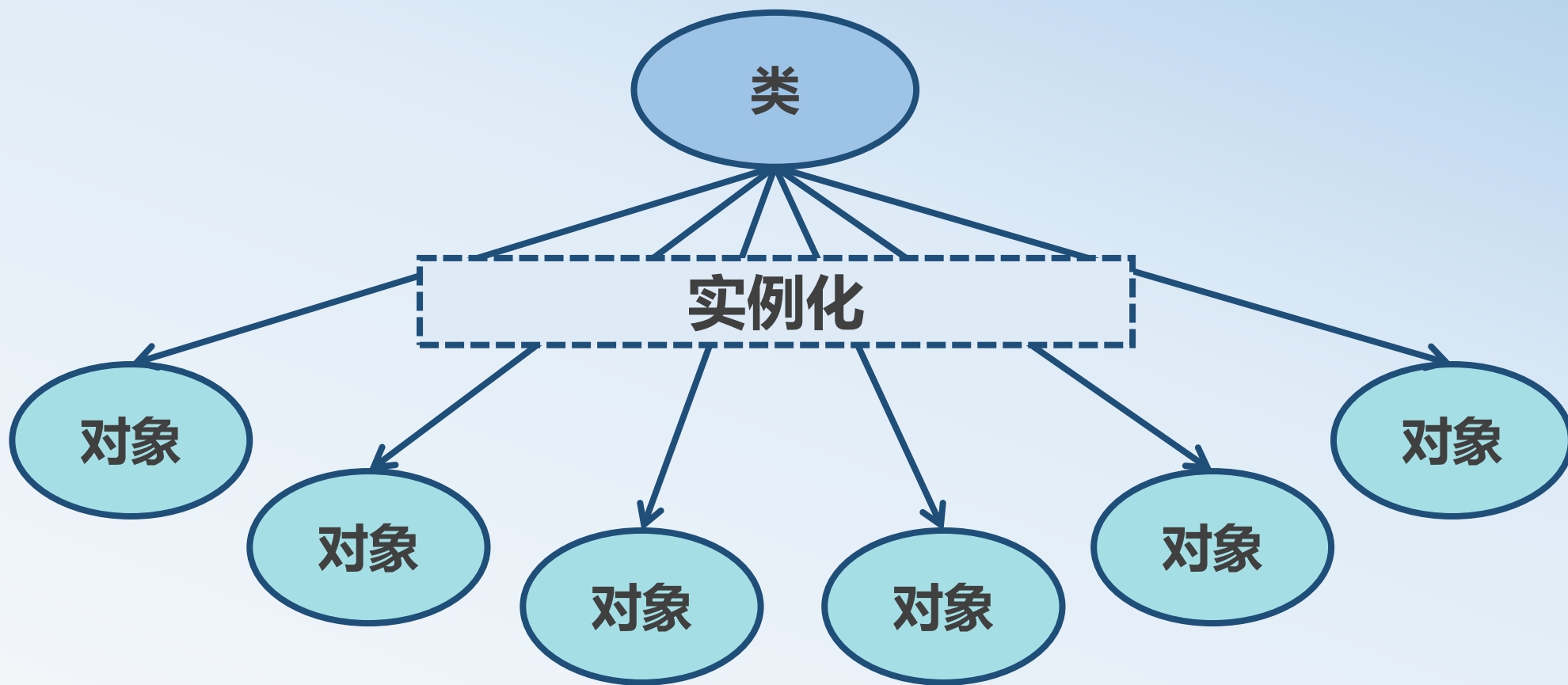
```
struct Student
{
    int num;
    char * name; // 成员变量（属性）
    char sex;
    void study(); // 成员函数（行为）
};
struct Student test; // 对象
.....
```

❖ **对象**：按照类类型创建的变量

面向对象程序设计：类与对象

- ❖ **对象**：现实世界中某个具体的**物理实体**在计算机逻辑中的映射和体现。
- ❖ **类**：是一种抽象的数据类型，是同种对象的集合与抽象，是具有共同行为和属性的若干对象的统一描述体。

面向对象程序设计：类与对象



面向对象程序设计：三大特性 封装

❖ C中的封装

❖ 现实生活中的封装

➤ 傻瓜相机

❖ C++中的封装

➤ Step1: 将数据和处理数据的函数封装为类；

➤ Step2: 可以将某些成员声明为private从而达到信息隐藏的目的。

面向对象程序设计：三大特性 封装

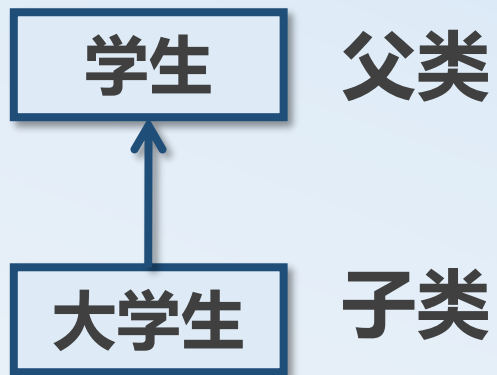
❖ 封装性 就是将**属性(数据)**和处理这些**属性(数据)**的**行为(方法)**结合起来，仅对外公开**接口**以达到**信息隐藏**的目的。

❖ 封装的优势：

- 封装是面向对象程序设计语言实现信息隐藏的方法。
- 封装隐藏了具体的实现细节，使某些成员设为私有从而提高了安全性和可靠性。

面向对象程序设计：三大特性 继承

❖现实生活中的继承：



❖C++中的继承：

在软件开发中，若已有类A，要创建类B，而类A和类B属性和行为基本相同，则只需在A的基础上增加些新的内容即可！

面向对象程序设计：三大特性 继承

❖ **继承**（ inheritance ）是指子类（ subclass ）继承父类（ superclass ），会自动取得父类除私有成员外的全部成员,或者对父类某些成员进行**改造**，从而做到代码重用（ reuse ）。

❖ **继承的优势**

➤ 提高了代码的重用率，提高了编程效率。

面向对象程序设计：三大特性 多态

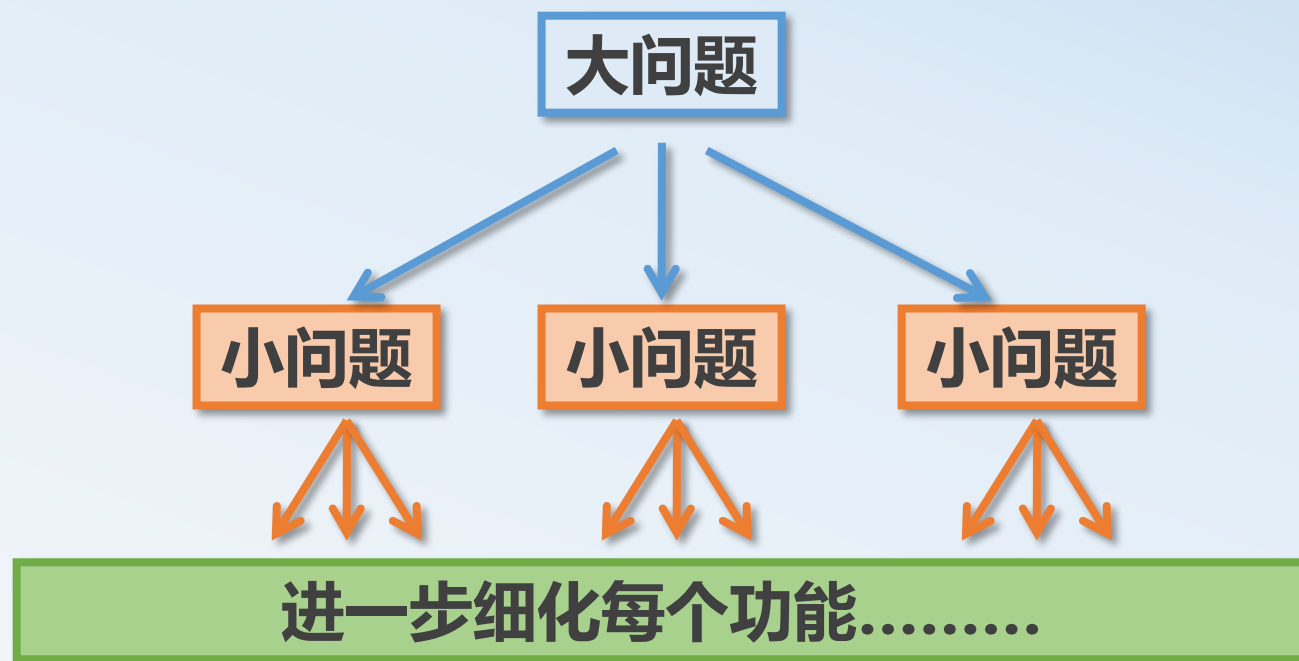


面向对象程序设计：三大特性 多态

- ❖ 多态性就是多种表现形式，具体来说，可以用“一个对外接口，多个内在实现方法”表示。
- ❖ 在面向对象理论中，多态性的定义是：**同一操作作用于不同的类的对象，将产生不同的执行结果。**
- ❖ 多态的优势：
 - 增强了程序的灵活性。

面向对象程序设计：面向过程

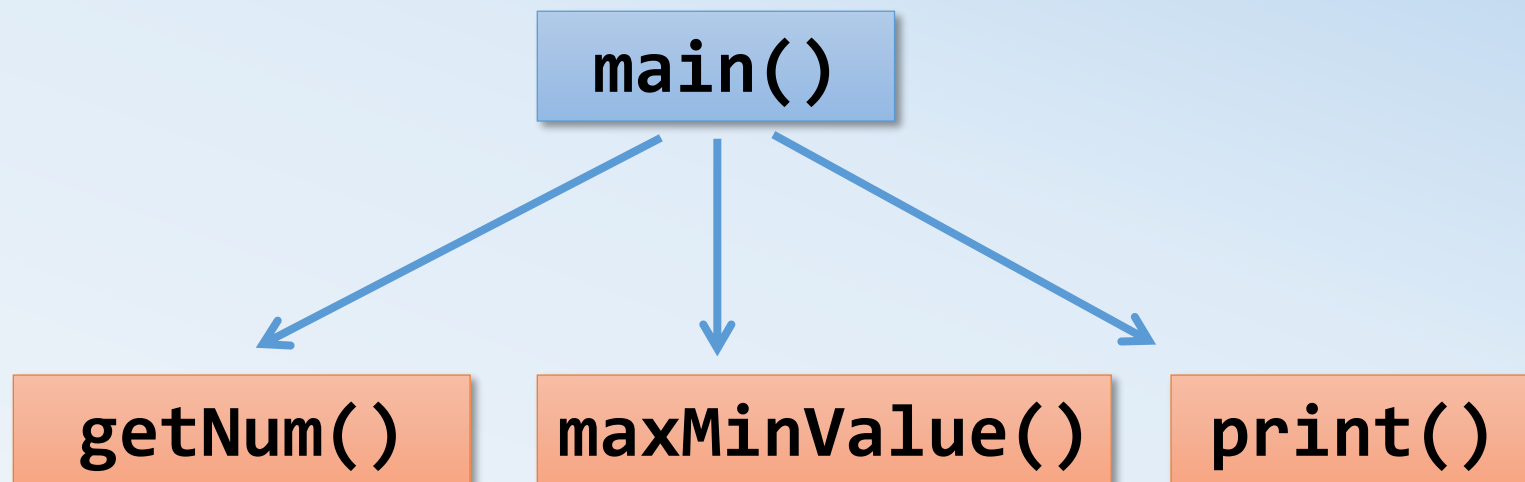
概念：以功能为基础，采用**自顶向下**，**逐步细化**的方法



面向对象程序设计：面向过程

自顶向下，逐步细化

只需考虑写几个功能



具体的实现每个功能

■ 面向对象程序设计：面向过程

❖ 面向过程的缺陷：

- 代码的**重用性**和**可维护性**差
- 数据和数据的操作分离，**数据安全不能保证**

面向对象程序设计：面向对象

1、概念：

将数据及其操作封装为类，以类的对象作为程序的基本元素，通过对象发送消息，从而调用相应方法完成各种功能

面向对象程序设计：面向对象

2、面向对象程序设计的特点：

- ①对象为基本元素
- ②数据——安全
- ③程序的维护量较小、代码的重用率高(继承)
- ④程序 = 对象 + 对象 + 对象 +

1

课程简介

2

C++概述

3

C与C++的关系

4

面向对象程序设计

5

第一个C++程序

第一个C++程序

C++ IDE集成开发环境

Microsoft Visual Studio (Visual C++)

Borland C++ Builder

Eclipse(MyIn + CDT + MinGW32 + GCC)

Dev-C++(MinGW32 + GCC)

Code::Blocks (可配合多款编译器使用)

CodeLite

C-Free

Qt Creator (可配合多款编译器使用)

KDevelop

C++编译器

微软的cl.exe编译器 (捆绑于Visual Studio套装中)

Intel C++ 编译器

GNU的GCC编译器

LLVM的Clang编译器

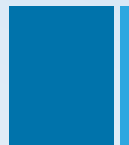
Borland公司的bcc.exe编译器 (捆绑于Borland C++ Builder套装中)

第一个C++程序

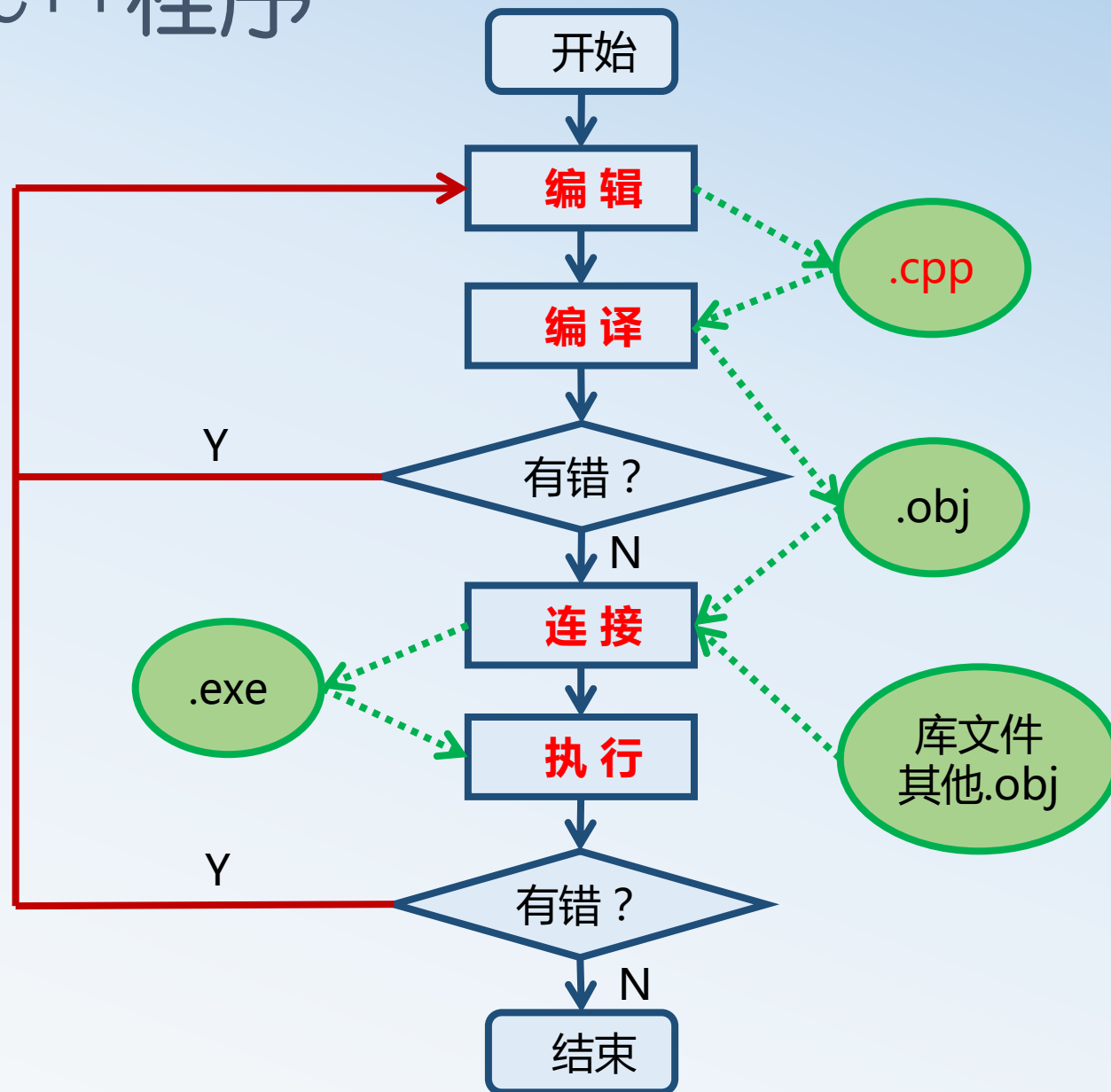
```
#include <iostream>    // 预处理命令
using namespace std;    // 使用命名空间

int main(void)
{
    cout << "Hello world!" << endl;

    return 0;
}
```



第一个C++程序





本讲教学目标

- 了解C++的起源
- 了解C++的应用领域
- 理解对象和类的基本概念
- 理解面向对象的三大特征
- 安装VS集成开发环境



THANKS

