

# 2023

# 面向对象程序设计

## 第一讲：面向对象程序设计概述

李际军 lijijun@cs.zju.edu.cn



- 课时：

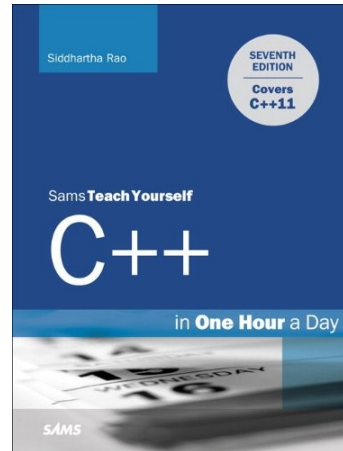
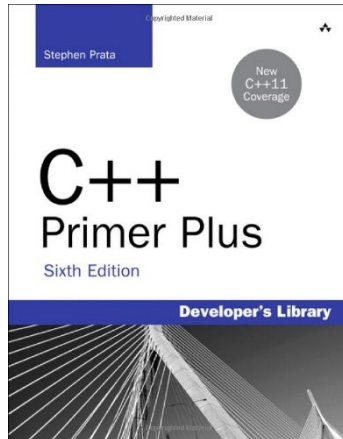
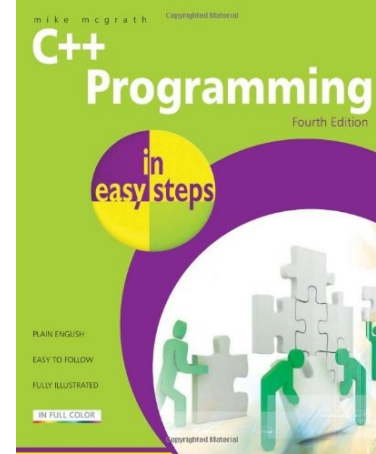
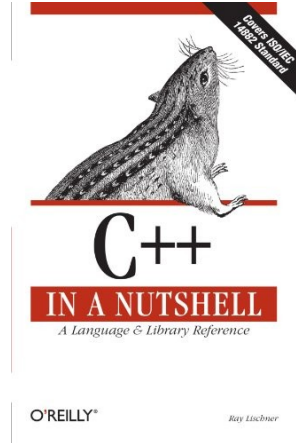
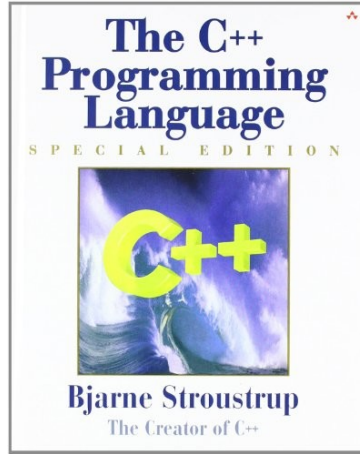
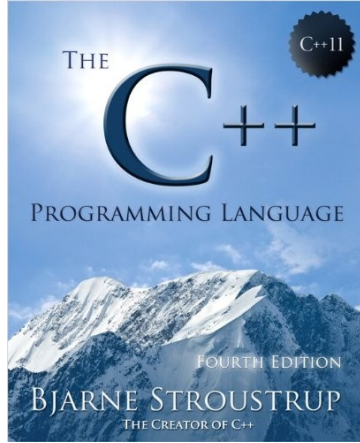
- 16 次课，共 32 学时，上机实验 16 学时；

- 成绩评定（课程组统一！）

本门课程的评分分为 4 个部分，每个部分分数分配如下：课堂表现：5 %；闭卷考试：50%；平时作业 + 大程序：45 %。

- 期末考试 50%；
  - 平时 50%
    - Pintia 作业 30%；
    - 大作业 15%；
    - 课堂表现 5%；（随堂点名 +pintia 平时作业综评）

# Useful Books on C++



# 参考教材

- Intruduction to Programming with C++ , Y.Daniel Liang , 机械工业出版社;
- Thinking in C++ , Bruce Eckel , 机械工业出版社;
- 面向对象程序设计高级教程 陈奇编著, 高等教育出版社;
- The C++ Programming Language , Bjarne Stroustrup , Addison-Wesley Publishing Company ;
- *C++ How to Program*– Fourth Edition, by H. M. Deitel, P. J. Deitel, Prentice Hall, New Jersey ;
- Programming: Principles and Practice Using C++, Ed.Bjarne Stroustrup , Addison-Wesley Publishing Company ;
- C++ primer ; Stanley B.Lippman, Josee Lajoie, Barbara E.Moo;
- 不限定.....

# C++ 部分网络资源

- <https://www.tutorialspoint.com/cplusplus/index.htm>
- <http://www.learncpp.com/>
- <http://www.cplusplus.com/>
- [https://www.cprogramming.com/tutorial/c++-tutorial.html/](https://www.cprogramming.com/tutorial/c++-tutorial.html)
- <http://www-h.eng.cam.ac.uk/help/tpl/languages/C++.html>
- <https://en.wikipedia.org/wiki/C%2B%2B>
- <https://isocpp.org>

## Useful Links on C++

- [C++ Programming Language Tutorials](#) – C++ Programming Language Tutorials.

<http://www.cs.wustl.edu/~schmidt/C++/>

- [C++ Programming](#) – This book covers the C++ programming language, its interactions with software design and real life use of the language.

[https://en.wikibooks.org/wiki/C++\\_Programming](https://en.wikibooks.org/wiki/C++_Programming)

- [Free Country](#) – The Free Country provides free C++ source code and C++ libraries for a number of C++ programming areas including compression, archiving, game programming, the Standard Template Library and GUI programming.

<https://www.thefreecountry.com/sourcecode/cpp.shtml>

- [C and C++ Users Group](#) – The C and C++ Users Group provides free source code from C++ projects in a variety of programming areas including AI, animation, compilers, databases, debugging, encryption, games, graphics, GUI, language tools, system programming and more.

<http://www.hal9k.com/cug/>



**01**

面向对象程序设计方法的发展历史

**02**

面向对象程序设计方法概述

**03**

面向对象程序设计的基本术语

**04**

面向对象程序设计的基本特征

**05**

面向对象程序设计语言

**06**

基于 Visual Studio 201X 的 C++ 应用程序的开发

# 01



## 面向对象程序设计方法 的发展历史



# 前言 / PREFACE

- 随着计算机技术的发展和开发软件复杂度的逐渐增加，计算机程序设计方法和程序设计语言也不断地演变和改进。
- **程序设计方法**历经了程序设计的自然描述、结构化程序设计（面向过程的程序设计方法）、面向对象的程序设计方法、面向对象的可视化编程方法；
- **程序设计语言**历经了机器语言、汇编语言、高级语言（面向过程的高级语言）、面向对象的编程语言、面向对象的可视化编程语言。
- **面向对象程序设计方法**为目前主流的程序设计方法。适合大型的、复杂的软件设计。

- 面向对象程序设计（Object-oriented programming, OOP）作为 20 世纪 90 年代以来程序设计的新思想、新方法，被认为是程序设计方法学的一场实质性的革命，是程序设计方法学的一个里程碑。

在面向对象程序设计的方法出现之前，我们都是采用面向过程的程序设计方法。

面向过程解决问题的思维方式是将一个大程序划分成若干个很小的结构，每个结构完成一个或多个功能，所有结构集合起来就可以完成一个大的程序结构。便于开发和维护。

这样的编程思想在 20 世纪 70 年代和 80 年代初期比较流行，并占绝对统治地位，但是到了 80 年代后期，随着计算机科学的发展和应用领域的不断扩大，软件规模增大和开发复杂度提高，它的弊端就暴露出来了。

## 01 在程序的可维护性方面


由于结构化编程仅仅是将大程序细化成若干个小结构，而并没有考虑数据的安全性问题，数据是属于整个程序的，这样就导致有些地方对数据的修改时，会对整个程序造成难以预料的影响。

## 02 在程序的可重用性方面

由于结构化设计并没有将相关数据和结构看做一个整体，所以无法利用已有的代码来创新新的代码。

随着软件工程的发展，软件越来越大，数据越来越多，面向结构程序设计所带来问题也越来越多，越来越严重，甚至曾一度导致“**软件危机**”，面向对象程序设计为解决“**软件危机**”而提出。

在 20 世纪 40 年代，在对数字模拟的分析研究中就引入了“对象”的概念，随后在对模拟系统的分析中，出现了大量的模拟仿真语言，如 Simscript、GPSS、CSL 和 Simula II。在 Simula II 中的“活动（Activity）、过程（Process）”概念正是如今面向对象程序设计语言中“类”和“对象”概念的雏型。



20 世纪  
40 年代

The diagram consists of five hexagonal nodes arranged horizontally. The first node is dark blue and contains the text '20 世纪 40 年代'. The subsequent four nodes are light blue and contain the text '20 世纪 60 年代', '20 世纪 70 年代', '20 世纪 80 年代', and '20 世纪 90 年代' respectively. The nodes are connected by a faint, light blue line.

20 世纪  
60 年代

20 世纪  
70 年代

20 世纪  
80 年代

20 世纪  
90 年代

60 年代中期，挪威计算中心的 Kisten Nygaard 和 Ole Johan Dahl 开发了 Simula67 语言，是第一个的面向对象程序设计语言。它引入了所有后来面向对象程序设计语言所遵循的基础概念：对象、类和消息。被称为面向对象程序设计语言的祖先或前身，为面向对象这一当前最流行、最重要的程序设计技术奠定了基础。

20 世纪  
40 年代

20 世纪  
60 年代

20 世纪  
70 年代

20 世纪  
80 年代

20 世纪  
90 年代

70 年代，美国施乐公司的帕洛阿尔托研究中心（PARC）开发了 Smalltalk 编程语言，又给面向对象的语言注入了新的血液，Smalltalk 被公认为历史上第二个面向对象的程序设计语言和第一个真正的集成开发环境（IDE）。它基于 Simula 语言的类和消息的概念，引入了继承和子类的概念，Smalltalk 编程语言对近代面向对象编程语言影响很大，所以称之为“面向对象编程之母”。

20 世纪  
40 年代

20 世纪  
60 年代

20 世纪  
70 年代

20 世纪  
80 年代

20 世纪  
90 年代

80 年代，面向对象程序设计成为了一种主导思想，相继出现了如 Object-C、C++、Self、Java 等面向对象语言。随着面向对象语言的发展，**面向对象程序设计方法也就应运而生**且得到迅速的发展。

20 世纪  
40 年代

20 世纪  
60 年代

20 世纪  
70 年代

20 世纪  
80 年代

20 世纪  
90 年代



90 年代以来，面向对象程序设计语言、面向对象程序设计方法广泛应用于程序设计，并逐渐形成了面向对象分析、面向对象设计、面向对象编程、面向对象测试等面向对象软件开发方法。从此，全世界掀起了一股面向对象的热潮，至今盛行不衰，面向对象程序设计方法逐渐成为程序设计的主流方法。

20 世纪  
40 年代

20 世纪  
60 年代

20 世纪  
70 年代

20 世纪  
80 年代

20 世纪  
90 年代

- 总之，面向对象程序设计方法是在结构化程序设计方法的基础上发展而来。采用此方法大大提高了软件开发效率，减少了软件开发的复杂性，提高了软件的可维护性、可扩展性。
- 面向对象的程序设计方法是当今普遍使用并大力推广的一种程序设计方法，它是计算机软件开发人员必须掌握的基本技术。

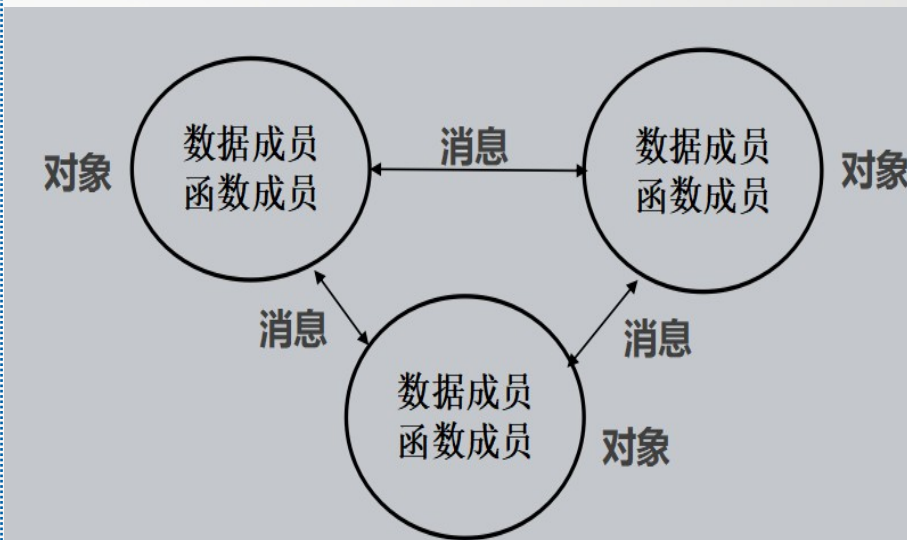
# 02



## 面向对象程序设计方法概述

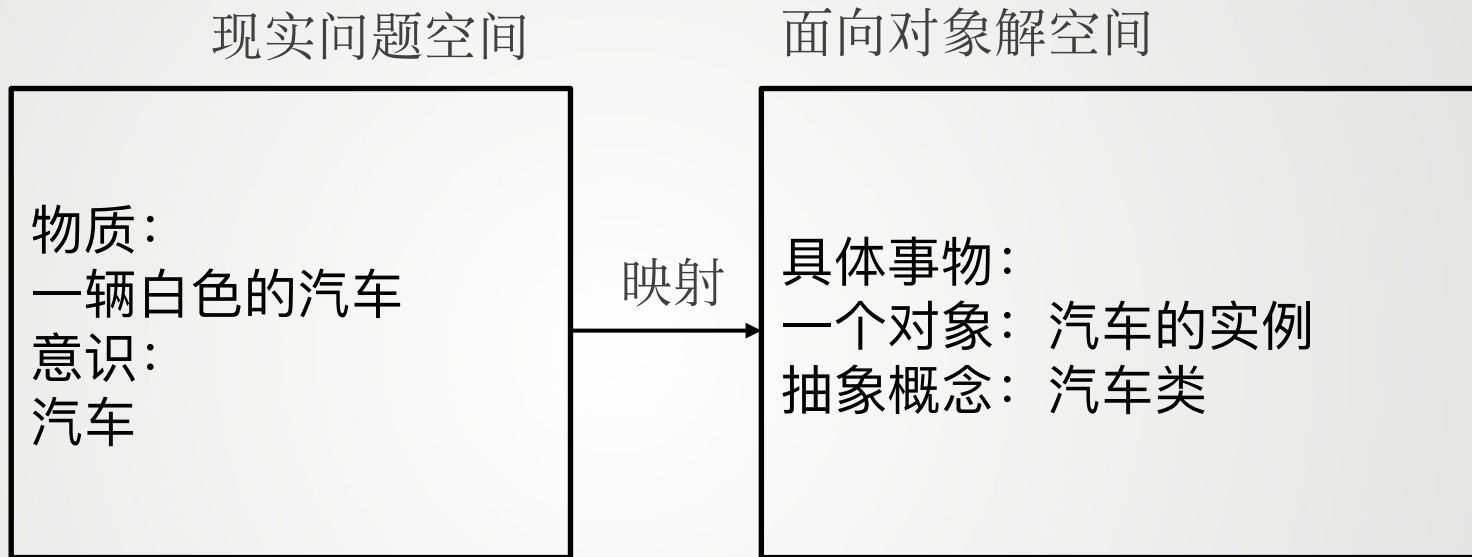
## 面向对象程序设计

- 面向对象程序设计方法（Object Oriented Programming，缩写 OOP）将数据及对数据操作的方法（函数）放在一起，形成一个相互依存，不可分离的整体——对象，从同类对象中抽象出共性，形成类。
- 类有两个成员：数据成员和成员函数。
- 对象之间通过消息进行通信。



面向对象程序设计模型

- ◆ 面向对象程序设计方法采用与客观世界相一致的方法设计软件，其设计方法是模拟人类习惯的思维方式。
- ◆ 软件开发的方法与过程尽接近人类认识世界、解决问题的方法与过程。
- ◆ 描述问题的问题空间（即问题域）与实现解法的解空间（即求解域）在结构上尽可能一致。
- ◆ 现实世界中的事物可以分为两大部分：物质和意识。
  - a. 物质指的是一个具体的事物，意识描述的是一个抽象的概念，是对客观存在事物的一种概括。
  - b. 例如“汽车”和“一辆白色的汽车”，“一辆白色的汽车”是物质，“汽车”是意识，是一个抽象的概念。

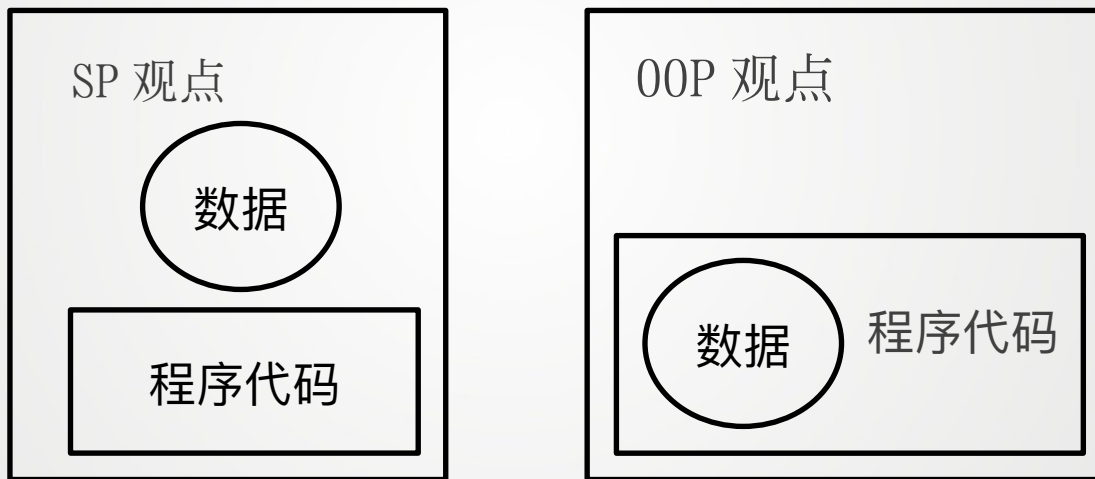


现实世界与面向对象系统之间对应关系

### 面向对象程序设计的特点

- ◆ 面向对象程序设计方法的主要特点是：程序 = 对象 + 消息。
- ◆ 每个对象都具有特定的属性（数据结构）和行为（操作自身数据的函数），它们是一个整体。
- ◆ 整个程序由不同类的对象构成，各对象是一个独立的实体，对象之间通过消息传递发生相互作用。

### SP 与 OOP 中代码和数据关系



SP 与 OOP 中代码和数据的关系



### 面向对象程序设计的优点

- ◆ 结构化的程序设计的数据和程序代码是分离的，而面向对象程序设计则将数据和操作数据的程序代码绑在一起构成对象。
- ◆ 面向对象程序设计方法使得开发的软件产品易重用、易修改、易测试、易维护、易扩充，降低了软件开发的复杂度。
- ◆ OOP 达到了软件工程的三个主要目标：重用性、灵活性和扩展性。降低软件开发难度，适合大型的、复杂的软件开发。

◆ 面向对象的程序设计方法是当今普遍使用并大力推广的一种程序设计方法，它是计算机软件开发人员必须掌握的基本技术。

◆ 目前，面向对象程序设计语言广泛使用的有： C++、 VisualBasic、 PowerBuilder、 Delphi、 C#、 Java 等。

◆ 面向对象程序设计是针对开发较大规模的程序而提出，目的是提高软件开发的效率。但不要把面向对象和面向过程对立起来，面向对象和面向过程不是矛盾的，而是各有用途、互为补充的。

# 03



## 面向对象程序设计 的基本术语

## 1. 对象

在现实世界中，一切事物都可以看作一个对象。

- ◆ 对象既可以是一个**有形**的具体事物，如一个人、一颗树、一台计算机；也可以是**无形的、抽象**的事件，如一场演出、一场球赛；一个对象既可以是个**简单对象**，也可以是由多个对象构成的**复杂对象**。
- ◆ 现实世界中的对象可以认为是：**对象 = 属性 + 行为**。
- ◆ 现实世界中对象具有如下特性：
  - 有一个名字以区别于其他对象；
  - 有一个状态用来描述它的某些特征，这个状态称为**属性**；
  - 有一组操作，每一个操作决定对象的一种功能或者行为，操作包括：
    - (1) 自身所承受的操作
    - (2) 施加其它对象的操作。

在面向对象程序设计中，对象是描述其属性的数据及对这些数据施加的一组操作封装在一起构成的一个独立整体。

- **对象 = 数据 + 操作**
- 对象中的数据表示对象的状态，对象中的操作可以改变对象的状态。

## 2. 类

### 现实世界中的类

在现实世界中，类是一组具有共同属性和行为的对象的抽象。

例如：李星、王晓、陈悦等是不同的学生对象，但他们有共同的特征：

- ◆有姓名、班级、学号等属性；

- ◆有能选课、听课、做作业等行为。

将所有同学都共有的这些属性和行为抽象出来，就构成一个学生类。

### 面向对象程序中的类

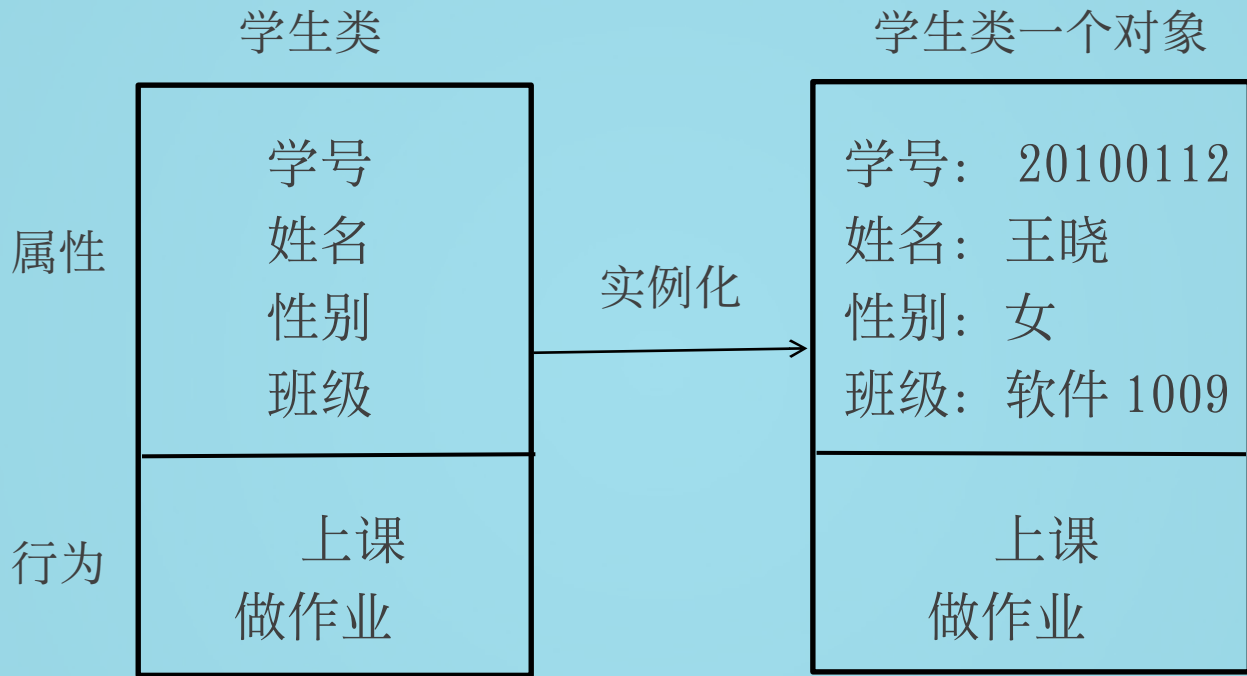
在面向对象程序设计中，类是一组具有相同数据和相同操作的一组对象的集合。

同一个类的不同对象具有其自身的数据，处于不同的状态中。

面向对象程序设计中，总先申明一个类，再由类生成一个具体对象。

### 类和对象的关系

类和对象的关系是抽象和具体的关系，类是多个对象进行综合抽象的结果，一个对象是类的一个实例。下图表示一个学生类和其中一个对象的关系。



类与对象的关系

### 3. 实例

- ◆ 实例就是由某个特定的类所描述的一个具体的对象。比如汽车就是交通工具的一个实例。
- ◆ 类是建立对象时使用的“模板”，按照这个模板所建立的一个个具体的对象，就是类的实际例子，简称实例。
- ◆ 如图所示，学生王晓是学生类的一个实例。

### 4. 属性

- ◆ 属性是类中所定义的数据，它是对客观世界实体所具有的性质的抽象。
- ◆ 类的每个实例即具体对象都有自己特有的属性值。
- ◆ 例如学生王晓的属性值有：姓名：王晓；年龄： 19 ； 班级： 软件 1102 ； 专业： 软件工程； C++ 成绩： 85 等。



### 5. 消息

- ◆ 在面向对象程序设计中，对象之间的联系是通过消息传递来实现的。一个对象向另一个对象发出的“请求”或“命令”被称为“消息”。
- ◆ 当对象收到消息时，就调用有关的方法，执行相应的操作。消息是一个对象要求另一个对象执行某个功能操作的规格说明。
- ◆ 通过消息传递完成对象间相互请求和相互协作。

### 5. 消息

消息具有三个性质：

- (1) 同一对象可接收不同形式的多个消息，产生不同的响应。
- (2) 相同形式的消息可以发送给不同对象，所做出的响应可以是截然不同的。
- (3) 消息的发送可以不考虑具体的接收者，对象可以响应消息，也可以对消息不予理会，对消息的响应并不是必须的。

在面向对象系统中，消息分为两类：

**公有消息：**由外界对象直接发送给这个对象的消息。

**私有消息：**对象自己发送给本身的消息。

私有消息对外是不开放的，外界不必了解它。外界对象只能向此对象发送公有消息，而不能发送私有消息，私有消息是由对象自身发送的。

### 6. 方法

- 方法就是对象所能执行的操作或所具有的行为，即类中定义的**服务**。例如，学生王晓能执行的操作有：上课、做作业等，实现这些操作的过程就是方法。
- 一个方法**有方法名、参数、方法体**，用来描述对象执行操作的算法、响应消息等。
- 在 C++ 语言中方法是通过成员函数来实现的。

# 04

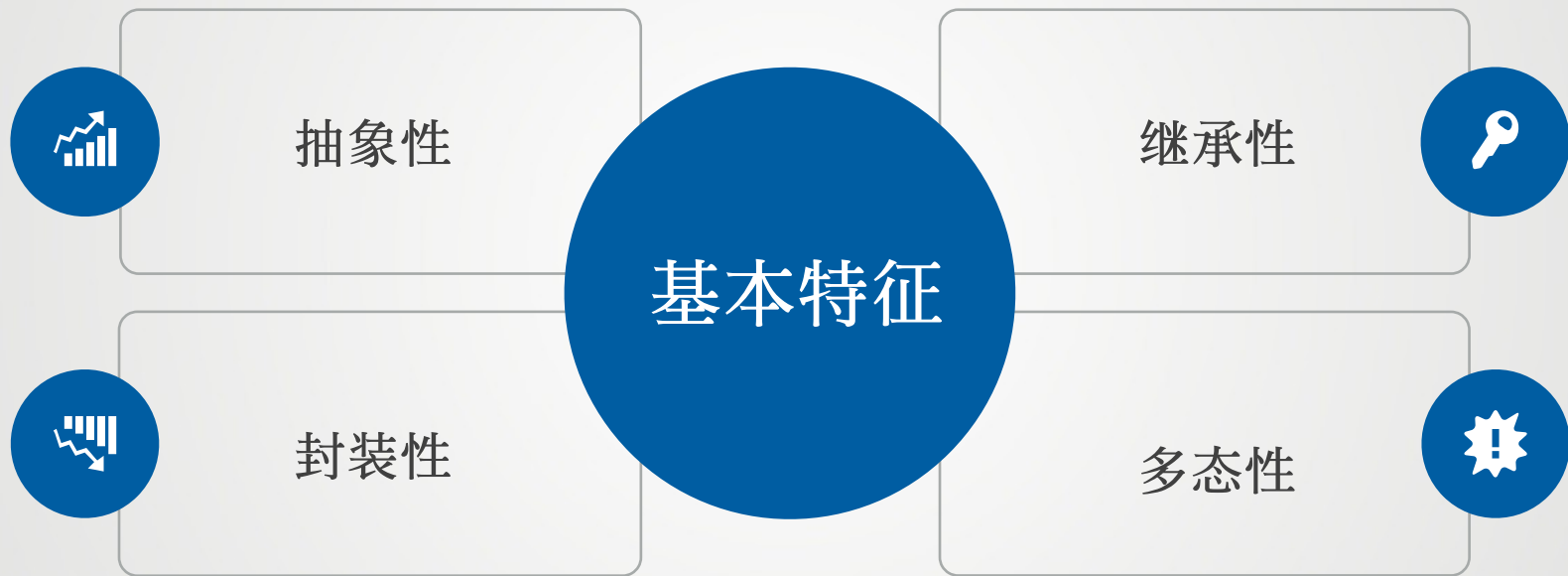


## 面向对象程序设计 的基本特征



## 1.4 面向对象程序设计的基本特征

37



# 抽象性

- 面向对象程序设计的基本要素是抽象。
- 抽象的哲学概念：从众多事物中抽取出共同的、本质的特征，而忽略次要的和非本质的特征。

例如：一个长方形是一个具体的对象，10 个不同尺寸的长方形是 10 个对象，这 10 个长方形有共同的：

属性：长和宽，只是具体值不同

行为：计算周长、计算面积

将这 10 个长方形抽象出一种类型，称为长方形类型。在 C++ 中，这种类型就称为类，这 10 个长方形属于同一类的对象。

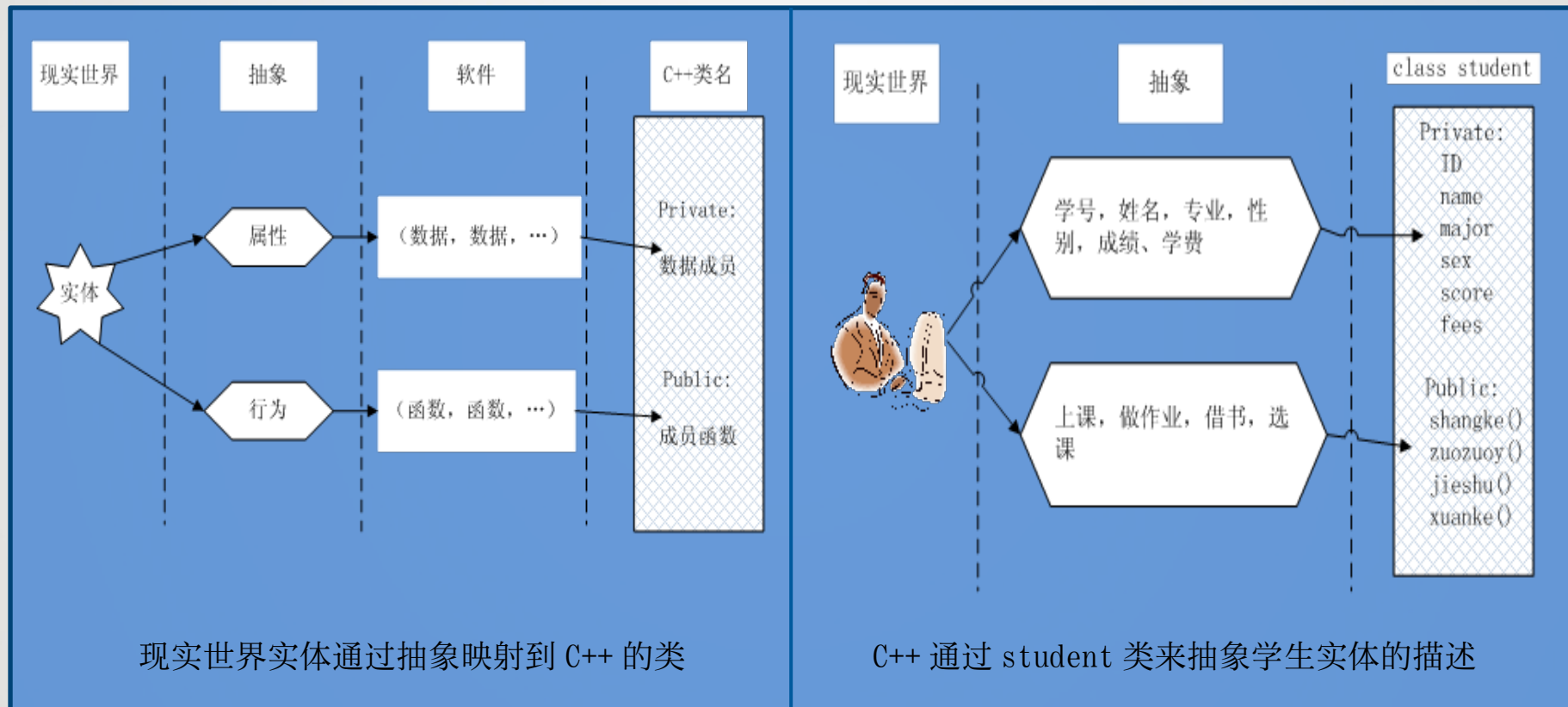
- 类是对象的抽象。



### 抽象

**数据抽象：**针对对象的属性，实现数据封装，在类外不可能被访问，实现数据隐藏。如建立一个学生类，学生会有以下特征：学号、姓名、专业、性别、学费、成绩等，写成类时都应是学生的**属性**。

**过程抽象：**针对对象的行为特征，如学生上课、写作业、借书等，这些方面可以抽象为方法，写成类时都是学生的**行为**。





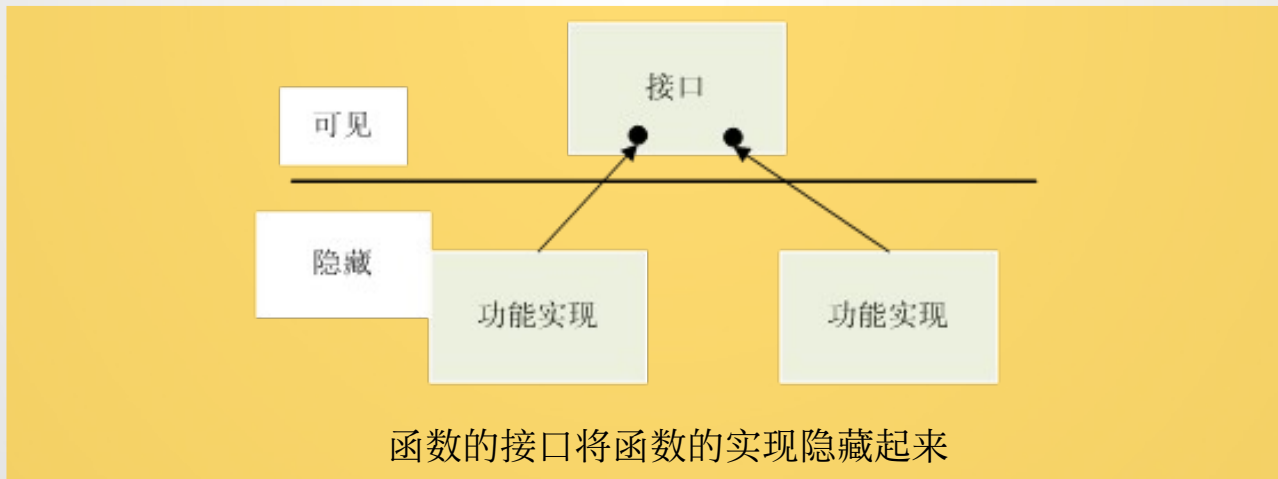
### 封装性

- 封装是面向对象程序设计方法的一个重要特征。
- 封装是将事物的属性和行为包装到对象的内部，形成一个独立模块单位。
- 封装是一种信息隐藏技术。

对象的内部对用户是隐藏的，不可直接访问。使得用户只能见到对象封装界面上的信息即**外特性**（对象能接受哪些消息，具有那些处理能力），而对象的**内特性**（保存内部状态的私有数据和实现加工能力的算法）对用户是隐蔽的。

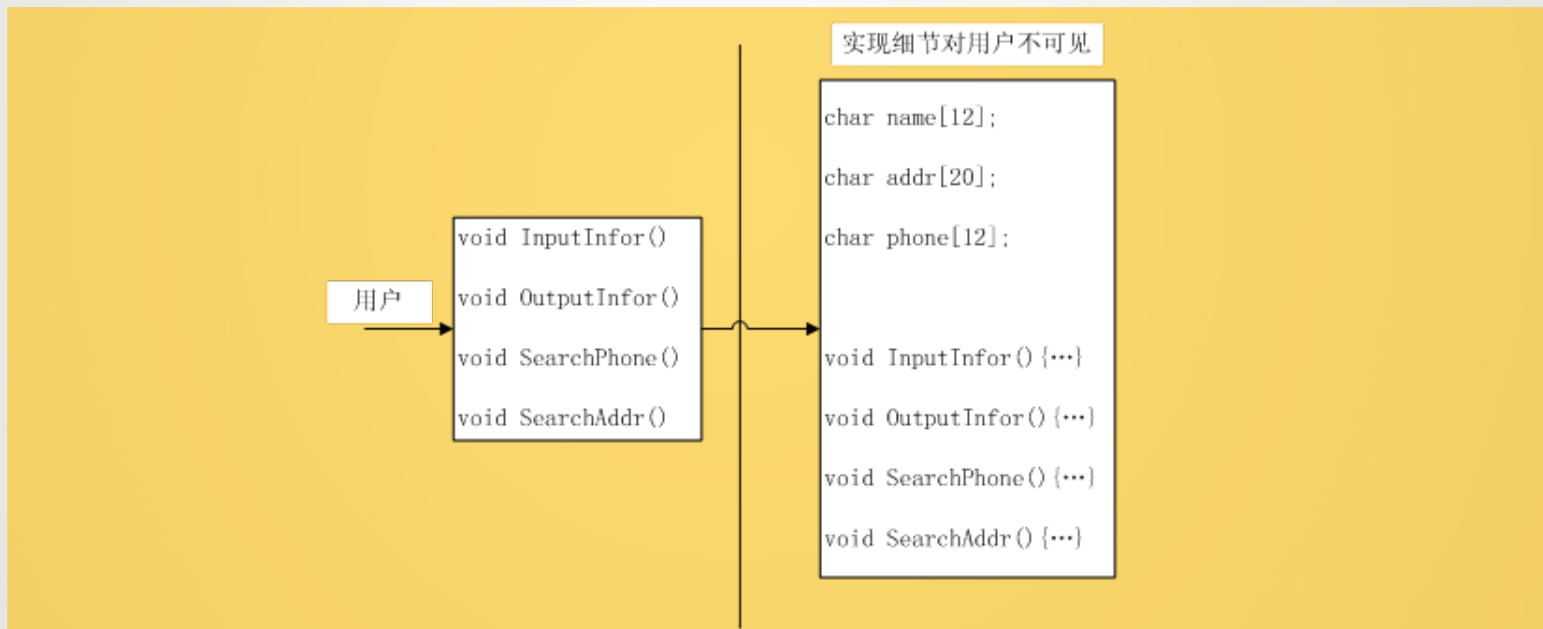
### 封装性

下图说明信息隐藏技术的具体实现，函数的调用者只需要了解函数的接口信息来正确的使用函数，而无需了解函数的具体实现，即函数接口与具体实现是独立的。



### 封装性

学生信息管理系统，如下图所示，系统实现中定义一个学生类。



学生类的数据封装和隐藏

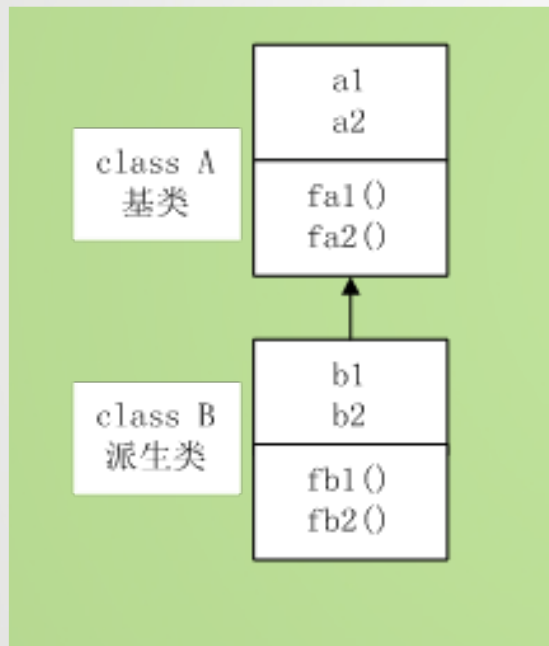
### 继承性

面向对象程序设计也提供了类似大自然中的物种遗传的生物继承机制，即子类自动共享父类之间数据和方法的语言机制。父对象拥有的属性和行为，其子对象通过继承也拥有了这些属性和行为。

继承是面向对象程序设计方法的一个重要特征，是实现软件重用的一个重要手段。

继承允许一个新类从现有类派生而出，新类能够继承现有类的属性和行为，并且能够修改或增加新的属性和行为，成为一个功能更强大、更满足应用需求的类。

## 继承性



- 如图所示的公有继承方式中，类 A 是基类，类 B 是派生类，类 B 可以继承类 A 的属性和行为。
- 类 B 只定义了 b1 、 b2 两个数据成员和 fb1 () 、 fb2 () 两个成员函数。
- 但类 B 继承类 A 后，其成员如下：

**数据成员有：** a1 、 a2 、 b1 、 b2

**成员函数有：**

fa1 () 、 fa2 () 、 fb1 () 、 fb2 () 。

## 继承

### 单继承

每个子类只能有一个父类。

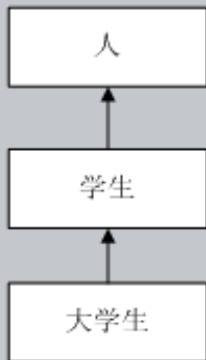


图1-11单继承

### 多重继承

每个子类有多个父类。

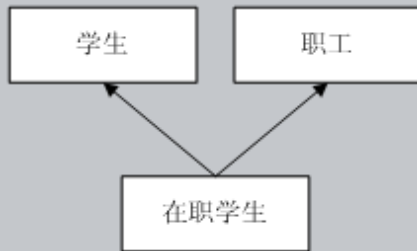


图1-12多继承

### 继承性

#### 优点

1. 继承能够清晰地体现相似类之间的层次结构关系。
2. 继承能够减少代码和数据的重复冗余度，提高程序的可重用性。
3. 继承能通过增强一致性来减少模块间的接口和界面，提高程序的易维护性。
4. 继承是自动传播代码的有力工具。
5. 继承是一种在普通类的基础上构造、建立和扩展新类的最有效手段。

继承为软件设计提供了一种功能强大的扩展机制，允许程序员基于已经设计好的基类创建派生类，并为派生类添加基类所不具有的属性和行为，极大地提高了软件的可重用性效率。

### 多态性

- ◆ 多态是面向对象程序设计的另一个重要的特征。
- ◆ 多态性：对象根据所接收的消息做出动作而呈现一定形态，从字面上解释，所谓多态性是指“**有许多种形态**”。
- ◆ 即“**一个接口，多种形态**”

在 OO 中是指，语言具有根据对象的**类型**以不同方式处理，即指同样的消息被**不同类型**的对象接收时导致完全不同的行为。



## 多态性

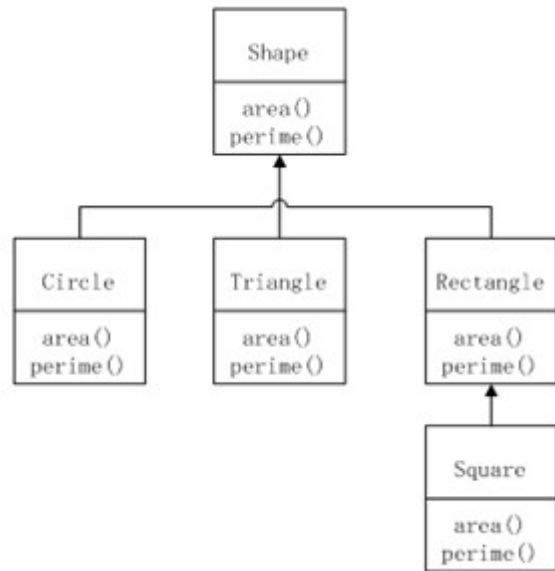
多态性与继承密切相关，利用类继承的层次关系，把具有通用功能的协议存放在类层次中尽可能高的地方，而将实现这一功能的不同方法置于较低层次，这样，在这些低层次上生成的对象就能给通用消息以不同的响应。

在 C++ 语言中，通过**重载**和**虚函数**两个方面来实现多态性。

**重载**称为编译时的多态性，

**虚函数**称为运行时的多态性。

具体内容将在后续的章节进行详解。



# 05



## 面向对象程序设计语言

面向对象  
程序设计  
语言

混合型的面向对象程序设计语言，典型的如 C++，这类语言是在传统的过程化语言中加入了各种面向对象语言的成分。

纯粹的面向对象程序设计语言，在纯粹的面向对象程序设计语言中，几乎所有的语言成分都是类和对象，典型的如 Java。

## 从 C 到 C++

- C 语言既具有高级语言的特点，又具有汇编语言的特点。
- 它由美国贝尔实验室的 D. M. Ritchie 于 20 世纪 70 年代在 B 语言的基础上扩充完善，发展而来的。C 语言应用广泛，具备很强的数据处理能力，适于编写系统软件、三维、二维图形和动画，许多大型应用软件都是用 C 语言编写的。
- C 语言有许多**优点**，例如编写简洁灵活，运算符和数据类型丰富，允许直接访问物理地址对硬件进行操作，程序执行效率高，支持结构化程序设计等特点。产生之后成为最广泛的程序设计语言之一。
- 但是 C 语言存在的一些**不足之处**，例如，对数据类型检查机制比较弱、语言结构不支持代码重用、大规模程序开发中程序员很难控制程序的复杂性等。
- 随着软件规模和复杂度的不断增加，C 语言这种面向过程的结构化程序设计方法已经难以适应开发大型软件的要求，出现了**软件危机**。



## 从 C 到 C++

- 但是当时基于 C 语言的广泛使用和深入人心，解决软件危机的最好方法不是另外发明一种语言去代替 C，而是在 C 的基础加以发展，扩充到面向对象领域，于是诞生了 C++ 语言。
- C++ 语言是在 C 语言的基础上，为克服 C 语言的不足之处，且支持面向对象程序设计而出现的一种通用程序设计语言。
- 它于 20 世界 80 年代美国贝尔实验室的 Bjarne Stroustrup 提出，保留了 C 语言原有的优点，增加了面向对象的机制，由于在 C++ 引入了类的概念，最初的 C++ 语言被称作“带类的 C（C with classes）”。后来为了强调它是 C 的增强版，就采用 C 语言中的自加运算符“++”，改称为“C++”。
- 从 C++ 的名字看出，C++ 是 C 的超集和扩展，因此 C++ 语言既可以用于面向过程的结构化程序设计，又可以用于面向对象的程序设计，是一种功能强大的混合型的程序设计语言。

## 其他的面向对象程序设计语言

其他的面向对象程序设计语言有

VisualBasic、PowerBuilder、Delphi、C#、Java 等。这些语言都是纯粹的面向对象编程语言。在 C++ 之后，影响巨大的就是 Java 和 C# 语言了，下面简单的介绍一下最有典型代表的 **Java 语言**。

- Java 是一种可以撰写跨平台应用程序的面向对象的程序设计语言。
- 由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计和 Java 平台（即 JavaSE, JavaEE, JavaME）的总称。
- Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于个人 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网，同时拥有全球最大的开发者专业社群。
- 在全球云计算和移动互联网的产业环境下，Java 更具备了显著优势和广阔前景。

# 06



## 基于 Visual Studio 201X 的 C++ 应用程序的开发

【例 1-1】创建一个控制平台应用程序，当其运行时在屏幕上显示“我们欢迎你”。

### 01

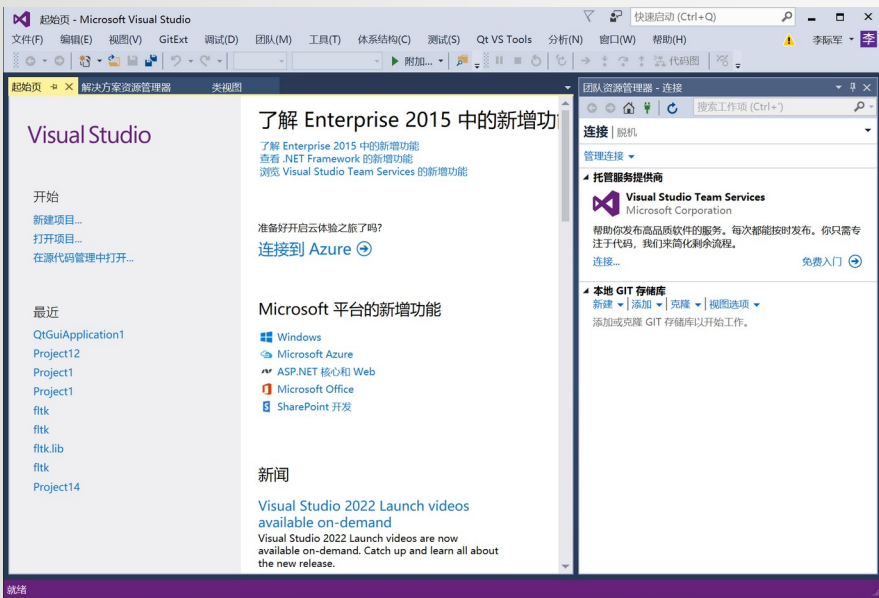
#### OPTION

## 建立一个项目

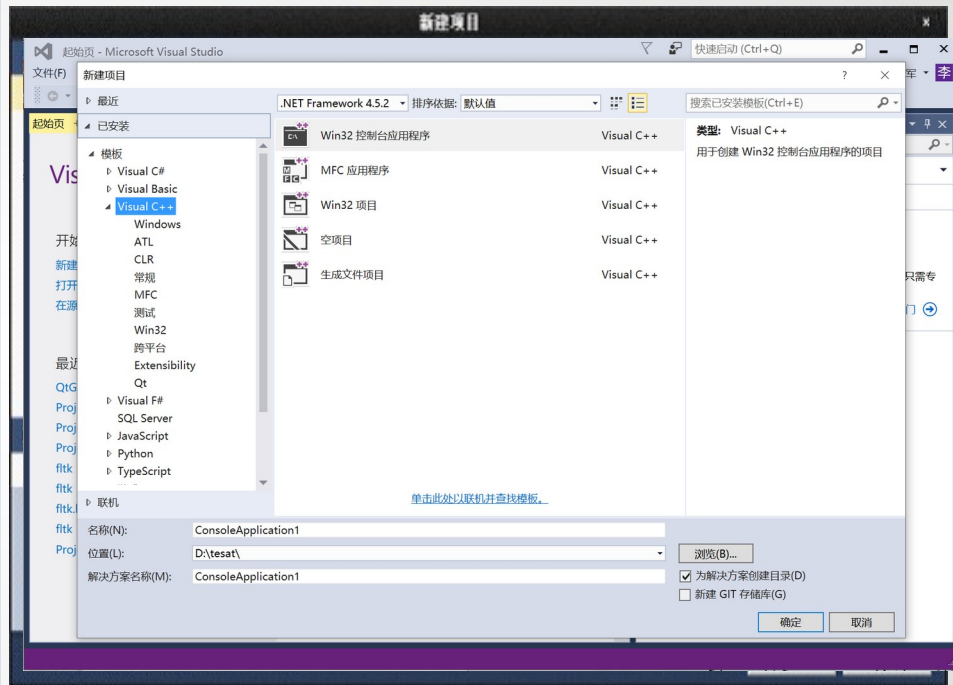
### ( Project )

在 Microsoft Visual Studio 201X 下开发程序时，首先要创建一个项目。项目中存放了建立程序所需要的全部信息。启动 Microsoft Visual Studio 201X；“文件”菜单上一次单击“新建” → “项目”菜单项。



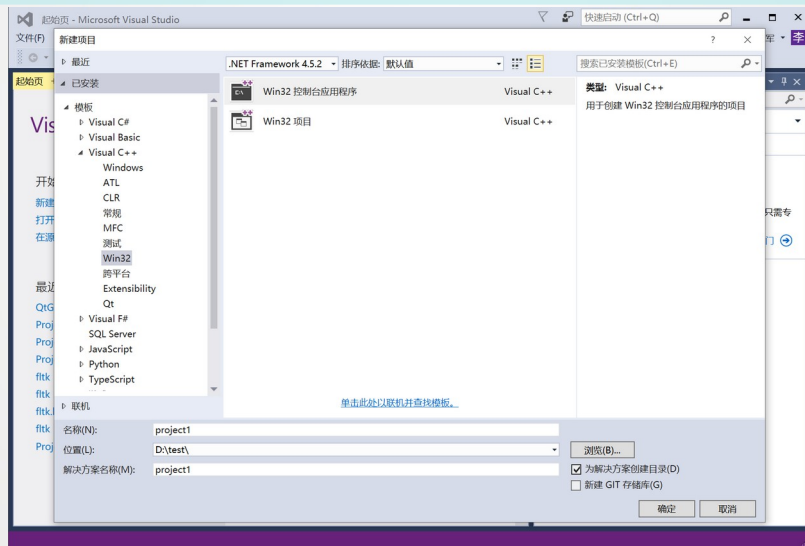


启动 Microsoft Visual Studio 2015



新建项目

选择“已安装模版”下的“Visual C++”的“Win32”，再选择对话框中间的“Win32 控制台应用程序”，然后在项目“名称”字段中输入“Project1”，在项目“位置”字段中输入要保存项目的位置。



单击“确定”按钮出现“Win32 应用程序向导”窗口，  
在“Win32 应用程序向导”窗口中单击“完成”按钮。

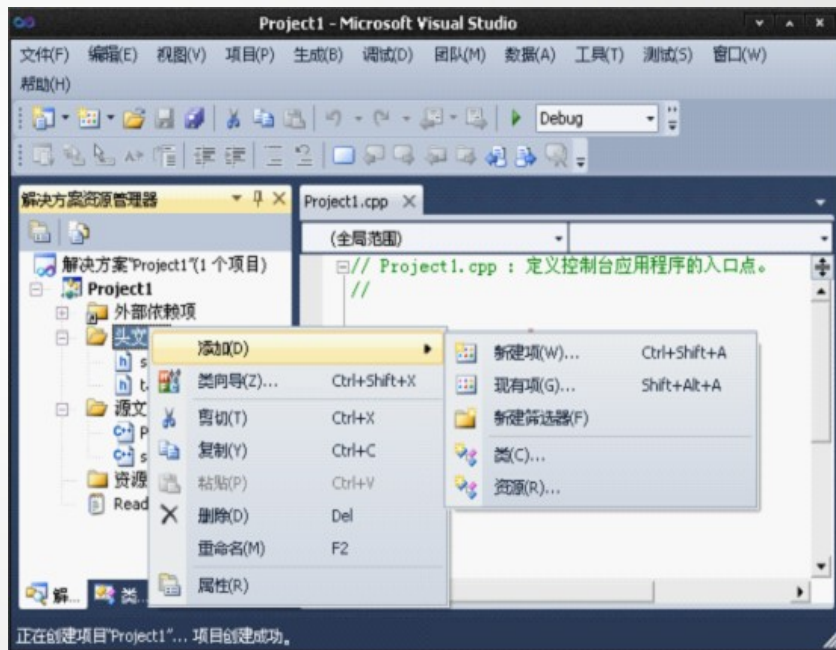


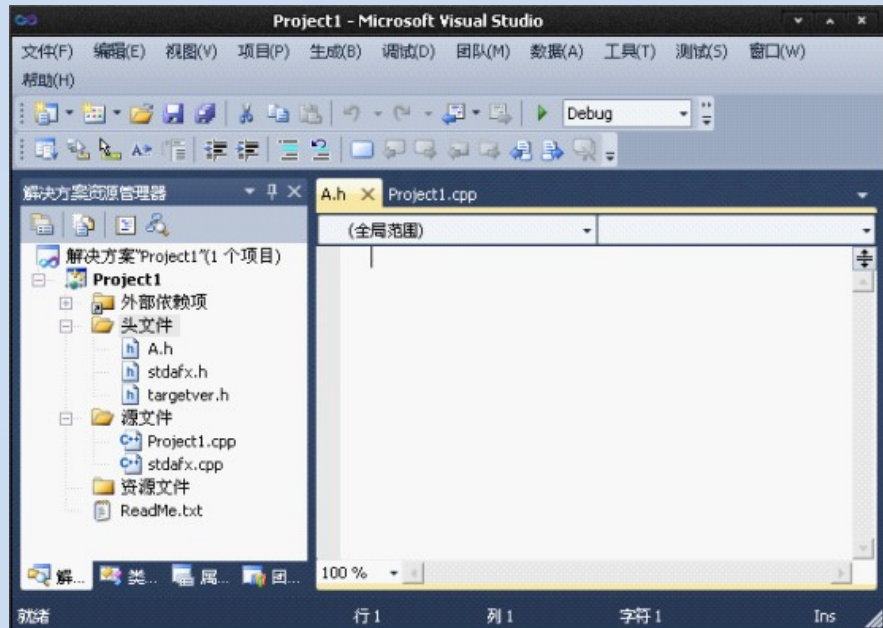
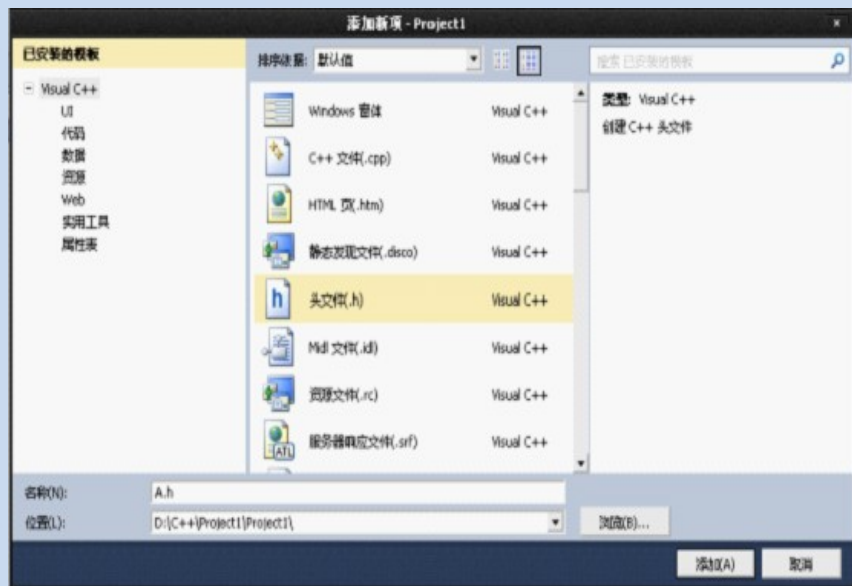
## 02

### OPTION

## 创建类

创建类 A，在图中选择“头文件”，在弹出的快捷菜单中选择“添加”→“新建项”，弹出“添加新项”对话框，在对话框中选择“头文件（.h）”，在名称中输入“A”。单击“添加”按钮，建立 A.h 头文件。





## 03

### OPTION

## 编辑 A.h 文件

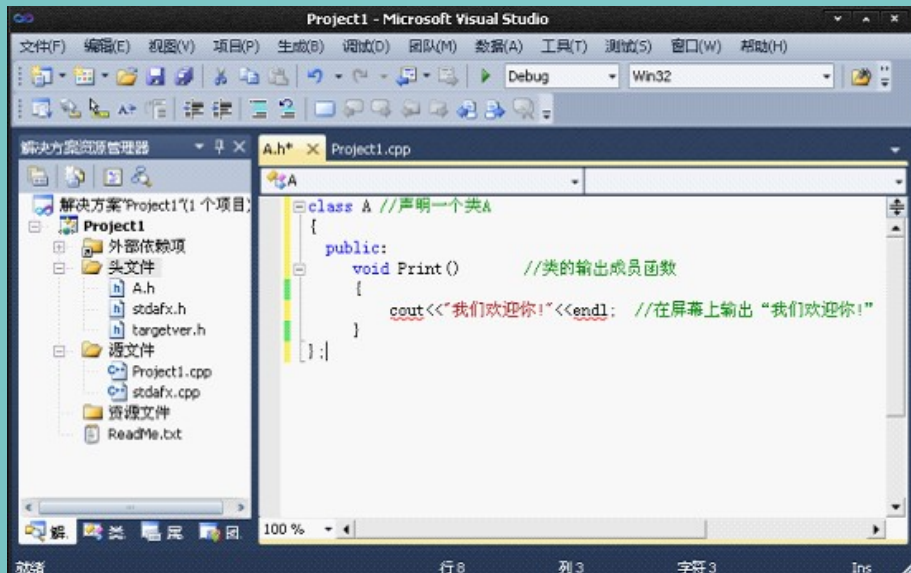
在类 A.h 的编辑区的空白区域中输入如下代码。

```
class A      // 声明一个类 A
{
    public:
        void Print()          // 类的输出成员函数
        {
            cout<<" 我们欢迎你 !"<<endl; // 在屏幕上输出 “我们
欢迎你 !”
        }
};
```

## 03

### OPTION

## 编辑 A.h 文件



在 “Project1.cpp” 文件中输入如下代码。

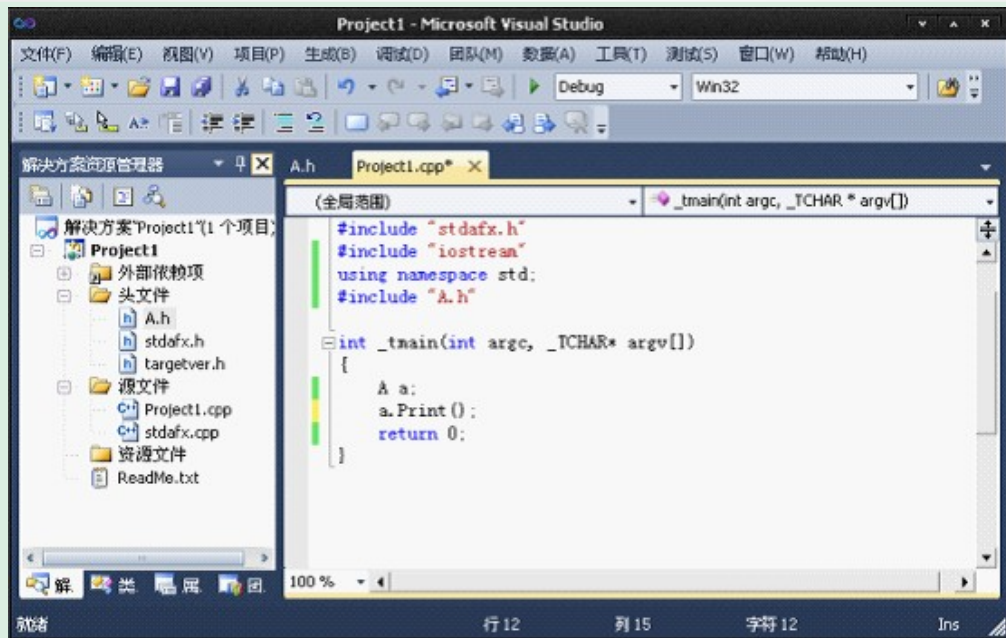
```
#include <iostream>
using namespace std;
#include "A.h"
int main()
{
    A a;
    a.Print();
    return 0;
}
```



## 04

### OPTION

## 编辑 Project1.cpp 文件

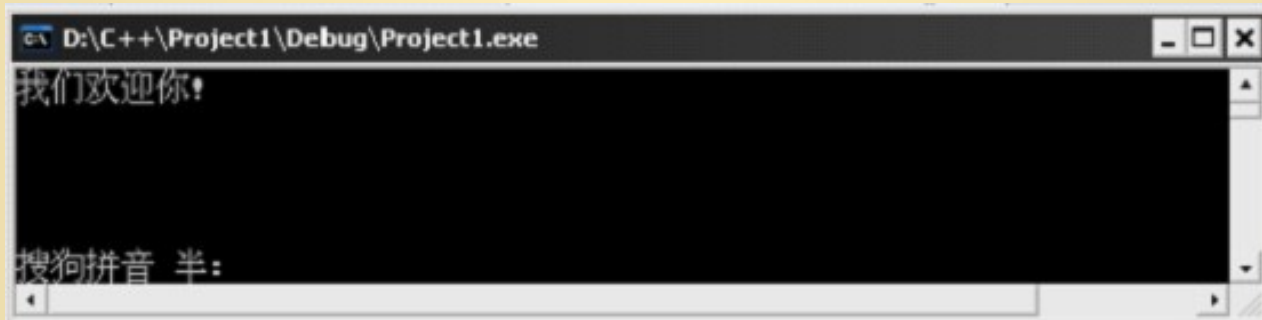


## 05

### OPTION

## 运行

执行“调试”→“开始执行（不调试）”命令或 Ctrl+F5 组合键，进行程序的编译、链接和运行，运行结果如图所示。



## 1. 面向过程程序设计方法

面向过程程序设计是以模块功能和处理过程设计为主的软件开发方法。其设计思想是采用“自顶向下，逐步求精，模块分解，分而治之”的解决问题方法。“自顶向下，逐步求精”是指将分析问题的过程划分成若干个层次，每一个新的层次都是上一个层次的细化，实现层次化和精细化；“模块分解，分而治之”是将整个系统分解成若干个易于控制、处理，完成一定功能的子任务或子模块，每分解一次都是对问题的进一步的细化，直到最低层次模块所对应的问题足够简单为止。各模块可以由顺序、选择、循环 3 种基本结构组成，每个模块功能可由结构化程序设计语言的子程序（函数）来实现。

## 2. 面向对象程序设计方法

面向对象程序设计方法是将数据及对数据操作的方法（函数）放在一起，形成一个相互依存，不可分离的整体——对象，从同类对象中抽象出共性，形成类。同类对象的数据原则上只能用本类提供的方法（成员函数）进行处理。类通过封装将接口与实现分离开来，通过接口与外界联系。对象之间通过消息进行通信。

### 3. 面向对象程序设计的有关术语

面向对象程序设计方法有关术语有对象、类、实例、属性、消息、方法等。

## 4. 面向对象程序设计的基本特征

面向对象程序设计方法的基本特征有抽象性、封装性、继承性和多态性。抽象就是从众多事物中抽取出共同的、本质的特征，而非本质的特征；封装是一种信息隐藏技术，它是将事物的属性和行为包装到对象的内部，形成一个独立模块单位，即对象的内部对用户是隐藏的，不可直接访问；继承反映的是对象之间的相互关系，它允许一个新类从现有类派生而出，新类能够继承现有类的属性和行为，并且能够修改或增加新的属性和行为，成为一个功能更强大、更满足应用需求的类，封装是实现软件复用的一个重要手段；多态是同一消息为不同的对象接受时可产生完全不同的行为，多态性与继承性密切相关。



## 5. 面向对象程序设计方法的特点

面向过程程序设计方法的缺陷是程序难以调试、修改和维护，代码的可重用性和共享性差，适用于小型系统或者是不复杂系统的开发。常用的语言有 BASIC、PASCAL、Fortran、C 等。

面向对象程序设计方法的优点是开发的软件产品易重用、易修改、易测试、易维护、易扩充，降低了软件开发的复杂度。达到了[软件工程](#)的 3 个主要目标，即重用性、灵活性和扩展性，适合大型的、复杂的软件开发。目前，面向对象程序设计语言广泛使用的有 C++、Visual Basic、Power Builder、C#、Java 等。



### 6. C++ 语言

C++ 是 C 语言的超集，C++ 对 C 语言的最大改进是引进面向对象机制，同时 C++ 依然支持所有 C 语言特性，保留对 C 语言的兼容，这种兼容性使得 C++ 不是一种纯正的面向对象的程序设计语言。