

高等院校非计算机专业教材

面向对象 程序设计

—— C++高级语言

主	编	赵	宏
编	著	李	敏
		王	恺
		王	刚

南开大学出版社

◆高等院校非计算机专业教材

◆南开大学教材资助立项项目

面向对象程序设计

——C++高级语言

赵宏 主编

李敏 王恺 王刚 编著

南开大学出版社

天津

图书在版编目(CIP)数据

C++高级语言 / 赵宏主编. —天津: 南开大学出版社, 2010.8
(面向对象程序设计)
ISBN 978-7-310-03500-7

I. ①C… II. ①赵… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 136211 号

版权所有 侵权必究

南开大学出版社出版发行

出版人: 肖占鹏

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022)23508339 23500755

营销部传真: (022)23508542 邮购部电话: (022)23502200

*

天津泰宇印务有限公司印刷

全国各地新华书店经销

*

2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷

787×1092 毫米 16 开本 22.375 印张 566 千字

定价: 40.00 元(配光盘)

如遇图书印装质量问题, 请与本社营销部联系调换, 电话: (022)23507125

内容提要

高级程序设计语言 C++ 是在 C 语言基础上的一种能够进行面向对象程序设计和传统过程化程序设计的语言，是 C 语言的超集。本书较详细地介绍了 C++ 语言的基础知识，通过大量的程序实例，针对初学者容易出现错误和困惑的地方提供了大量的附注，帮助读者更好地理解 C++ 的基本概念和技术。

全书共分 19 章，分别介绍了程序设计的基本概念、C++ 程序的基本组成，数据类型、常量和变量，运算符、表达式和语句，程序控制结构，函数初步与变量的存储类型，数组，指针和引用，字符串，函数，构造数据类型，编译预处理，类与对象，继承，多态性，运算符重载，输入/输出流，文件，模板，MFC 入门等内容。

本书是专门为高等院校非计算机专业 C++ 高级语言程序设计课程编写的教材，面向 C++ 初学者，不要求读者已经熟悉相关的编程概念和有 C 语言方面的背景知识。本书也适合自学者使用。

前言

目前, 不仅许多计算机专业和多数软件学院的程序设计课程选择了 C++ 作为程序设计的第一门语言, 越来越多的理工科专业也把 C++ 作为计算机基础课, 一方面是由于 C++ 是应用最广的面向对象语言, 另一方面是由于它有利于初学程序设计的学生学习一般的编程技巧。南开大学理工科公共计算机基础课程“面向对象程序设计”, 选用的就是 C++ 语言, 该课程的目标是培养学生基本的程序设计能力。然而, 公共计算机基础课课时有限, 近几年的教学经验表明, 把 C++ 语言作为高级语言程序设计的教学语言对于教师和学生都是有难度的, 主要的问题是 C++ 的规模和 C++ 程序的复杂程度, 往往使刚刚步入大学的学生感到困难重重。教材的编写一直是一个艰巨而具有探索性的工作。面对非计算机专业的理工科学生, 如何使初学者在有限的课时内打下良好的程序设计基础, 目前仍有许多值得探索的地方。由于他们不再有相关的后续课程, 进一步提高程序设计能力的学习机会较少, 公共计算机基础课教材的编写更是一个挑战。

国内外同类教材主要是面向计算机专业的学生, 即使是为非计算机专业的理工科学生编写的教材, 也仅仅是对专业教材的简单取舍, 较少地考虑非计算机专业的理工科学生学习程序设计的需求和特点, 在内容和程序实例的选取上, 在文字叙述上存在不足, 在教材的编写风格上也过于传统, 较难激发学生的学习兴趣。

本书是南开大学教材建设资助立项教材。编著者力求做到:

(1) 适应 21 世纪课程体系和教学内容改革方向要求, 抓住授课对象是非计算机专业的本科一年级学生的特点, 注重内容的选取和章节的安排。力求做到学习内容循序渐进; 文字叙述简单、易于理解; 全书难易得当、重点突出, 适合非计算机专业学生和 C++ 语言自学者学习程序设计语言时使用。

(2) 注重语法规则的清晰讲解和配备完整的程序实例。加强程序实例的选择和比例配置, 训练和培养学生分析解决问题的思想和能力。

(3) 根据初学者容易出错的地方, 给出了大量的提示、提问和学习指导, 适合课堂教学和自学。

(4) 理论与实际相结合。同时为本书编写了配套的实习指导和习题集, 根据各章节的内容给出相应的上机实习内容; 强化并丰富相应的习题, 以逐步提高学生的程序设计能力, 从而能够使用 C++ 高级语言解决实际的问题。

本书由担任南开大学信息学院公共计算机基础教学部“面向对象程序设计”课程的任课老师, 结合多年的教学经验, 根据我国高校的非计算机专业理工科学生学习程序设计课时少、时间短的特点编写。赵宏负责第 1 章至第 5 章的编写并统编全书, 第 6 章至第 9 章和第 19 章由王恺编写, 第 10 章、第 11 章和第 15 章至第 18 章由李敏编写, 第 12 章至第 14 章由王

刚编写。南开大学信息科学技术学院的硕士研究生曹瀛、边旭霞和陈亮对本书及其配套上机实习指导与习题集进行了文字和习题答案校对工作。本书的编写还得到了南开大学出版社张燕老师的大力支持。

本书的编著者参考了国内外许多C++程序设计语言方面的书籍,力求有所突破和创新。但由于能力和水平的限制,书中出现的不妥乃至错误之处,请阅读本书的老师和同学指正。

编 者

2010年3月于南开园

目 录

第1章 初识 C++	1
1.1 程序设计 (Programming) 的基本概念	1
1.2 高级程序设计语言 C++	4
1.3 第一个 C++ 程序	5
1.4 小结	13
1.5 学习指导	13
第2章 数据类型、常量和变量	16
2.1 数据在计算机中存储的基本概念	16
2.2 基本数据类型	22
2.3 常量	23
2.4 变量	25
2.5 小结	27
2.6 学习指导	27
第3章 运算符、表达式和语句	28
3.1 运算符和表达式	28
3.2 类型转换	35
3.3 语句	37
3.4 小结	43
3.5 学习指导	44
第4章 程序控制结构	45
4.1 顺序结构	45
4.2 选择结构	45
4.3 循环结构	50
4.4 转向语句	55
4.5 应用实例	58
4.6 小结	61
4.7 学习指导	61

第5章 函数初步与变量的存储类型	62
5.1 函数的基本概念	62
5.2 函数的定义	62
5.3 函数的传值调用及函数原形	64
5.4 变量的存储类型、作用域和生存期	66
5.5 小结	70
5.6 学习指导	70
第6章 数 组	72
6.1 数组的概念	72
6.2 数组的声明	73
6.3 数组的初始化	75
6.4 数组的访问	76
6.5 小结	82
6.6 学习指导	82
第7章 指针和引用	83
7.1 指针的概念	83
7.2 指针变量的声明、初始化和访问	84
7.3 指针与数组	86
7.4 指针的运算	89
7.5 指向指针的指针	93
7.6 const 指针	94
7.7 堆内存分配	96
7.8 引用	98
7.9 小结	100
7.10 学习指导	101
第8章 字符串	102
8.1 字符串的概念	102
8.2 字符串的输入输出	103
8.3 指针与字符串	106
8.4 常用的字符串函数	109
8.5 小结	114
8.6 学习指导	114
第9章 函 数	115
9.1 函数的调用机制	115

9.2 函数的递归调用.....	116
9.3 带默认形参值的函数.....	120
9.4 函数的作用域.....	122
9.5 内联函数.....	123
9.6 函数重载.....	124
9.7 函数指针.....	127
9.8 函数与指针.....	128
9.9 函数与引用.....	134
9.10 小结.....	136
9.11 学习指导.....	137
第 10 章 构造数据类型	138
10.1 结构体.....	138
10.2 枚举.....	146
10.3 类型重定义 typedef.....	147
10.4 小结.....	148
第 11 章 编译预处理	150
11.1 宏定义.....	150
11.2 条件编译.....	153
11.3 文件包含和多文件结构.....	155
11.4 小结.....	159
第 12 章 类与对象	160
12.1 类和对象的基本概念.....	160
12.2 类的声明与定义.....	160
12.3 对象.....	163
12.4 构造函数和析构函数.....	168
12.5 拷贝构造函数.....	173
12.6 类的静态成员.....	176
12.7 友元.....	180
12.8 this 指针.....	183
12.9 string 类.....	185
12.10 应用实例.....	190
12.11 小结.....	208
12.12 学习指导.....	209
第 13 章 继 承	210
13.1 什么是继承.....	210

13.2	派生类的定义	211
13.3	函数重定义	213
13.4	保护 (protected) 成员	214
13.5	派生类的继承方式	215
13.6	派生类的构造函数与析构函数	218
13.7	类型兼容	222
13.8	多重继承	224
13.9	应用实例	229
13.10	小结	230
13.11	学习指导	231
第 14 章	多态性	232
14.1	多态性的概念	232
14.2	虚函数	233
14.3	抽象类	240
14.4	应用实例	242
14.5	小结	257
14.6	学习指导	258
第 15 章	运算符重载	259
15.1	运算符重载的概念	259
15.2	运算符重载的方法	261
15.3	运算符重载的规则	270
15.4	特殊运算符的重载	271
15.5	类类型转换	278
15.6	小结	281
第 16 章	输入/输出流	282
16.1	输入/输出流简介	282
16.2	输入/输出流的成员函数	284
16.3	输入/输出的格式控制	290
16.4	自定义数据类型的输入/输出	296
16.5	小结	299
第 17 章	文 件	301
17.1	文件与流	301
17.2	文件的打开与关闭	303
17.3	文本文件的顺序读写	304
17.4	二进制文件的顺序读写	307

17.5	文件的随机读写	311
17.6	小结	317
第 18 章	模 板	318
18.1	函数模板	319
18.2	类模板	325
18.3	小结	329
第 19 章	MFC 入门	330
19.1	认识 MFC	330
19.2	一个简单的 MFC 应用程序	331
19.3	常用控件	336
19.4	MFC 应用程序开发实例	337
19.5	小结	344

第1章 初识 C++



导 读

本章主要介绍程序设计的基本概念、步骤和方法,简单介绍高级程序设计语言 C++, 并通过对第一个 C++ 程序的学习,初步了解 C++ 源程序的组成和组成元素,集成开发环境 Visual C++ 6.0 的编辑、编译、连接和运行步骤,并能够编写具有简单输入输出功能的 C++ 程序。

本章难度指数★,教师授课 2 课时,学生上机练习 2 课时。

1.1 程序设计 (Programming) 的基本概念

计算机程序 (Program) 的作用是指示计算机进行必要的计算和数据处理从而帮助我们解决特定的问题。计算机语言要处理两个概念——数据和算法,数据是程序使用和处理的信息。面对问题,需要找出解决问题的方法,我们把这种能够在有限的步骤内解决问题的过程和方法称为算法 (Algorithm)。程序设计是指设计、编制、调试程序的方法和过程,是寻找算法并用计算机能够理解的语言表达出来的一种活动。

1.1.1 程序设计方法

20 世纪 60 年代末期随着“软件危机”的出现,程序设计方法的研究开始受到重视。结构化程序设计方法 (Structured Programming, SP) 是程序设计历史中最早提出的方法。70 年代中后期,针对结构化程序设计在进行大型项目设计时存在的缺陷,提出了面向对象程序设计 (Object Oriented Programming, OOP) 方法。20 年来面向对象程序设计方法的大量研究工作,逐步替代了传统的结构化程序设计方法,成为目前最重要的程序设计方法。

1. 结构化程序设计 (SP)

SP 方法的核心是将程序模块化,主要通过使用顺序、分支 (选择) 和循环 (重复) 等三种基本结构,形成具有复杂层次的结构化程序。

SP 方法采用“自顶向下,逐步求精”的设计思想,其理念是将大型的程序分解成小型和便于管理的任务,如果其中的一项任务仍然较大,就将它分解成更小的任务。程序设计的过程就是将程序划分成为小型的、易于编写的模块。程序的模块功能独立,只使用三种基本结构,具有单一出口和入口,增加了模块的独立性,可以像搭积木一样根据需要使用不同的模块。C 语言的设计有助于使用结构化程序设计的方法,程序员开发程序单元 (称为函数) 来表示各个任务模块。图 1-1 是采用结构化程序设计的程序结构示意图。

SP 方法反映了过程性编程的思想,根据执行的操作来设计一个程序。它简单易学、容易掌握,模块的层次清晰,降低了程序设计的复杂性,程序的可读性强。SP 方法便于多人分工

开发和调试，从而提高了程序的可靠性。

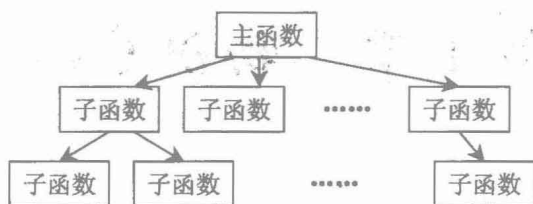


图 1-1 结构化程序设计的程序结构示意图

2. 面向对象程序设计 OOP

与强调算法的过程性编程不同，OOP 方法强调的是数据。它试图让语言来满足问题的要求，理念是设计与问题的本质特性相对应的数据类型——类。OOP 根据人们认识世界的观念和方法，把任何事物都看成对象，复杂的对象是由简单的对象以某种方式组成，认为世界是由各种对象组成的。

OOP 方法以“对象”为中心进行分析和设计，将对象设计成解决目标问题的基本构件，思想是先将问题空间划分为一系列对象的集合，再将具有相同属性和行为的对象抽象成一个类，采用继承机制来建立这些类之间的联系，形成结构层次。面向对象程序设计方法的本质就是不断设计新的类和创建对象的过程。图 1-2 是类和对象的关系示意图。

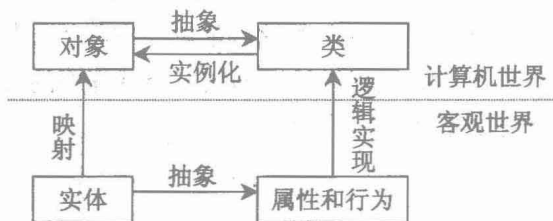


图 1-2 类和对象的关系示意图

OOP 方法中的类通常规定了可以使用哪些数据和对这些数据执行哪些操作。数据表示对象的静态特征——属性，操作表示了对象的动态特性——行为。例如，正在设计一个关于计算圆的面积和周长的程序，可以定义一个描述圆的类。类中定义的数据部分包括圆心的位置和半径；类中定义的操作部分包括设置圆心、设置半径、计算面积和计算周长等。

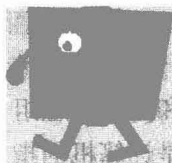
OOP 方法首先设计类，它们准确地表示了程序要处理的东西，然后就可以设计使用这些类的对象的程序。它是自下向上（bottom-top）的编程方法，即从低级组织（如类）到高级组织（如程序）的处理过程。OOP 以对象为中心的设计方式符合人类认识事物和解决问题的思维方式和方法。OOP 方法的主要特征：

- 封装性：封装即以对象的方式将一组数据和与这组数据有关的操作集合封装在一起，对象内部的状态和功能的实现细节对外不可见，很好地实现了信息隐藏，使其免遭不必要的访问。

- 继承性：继承能够使程序员使用旧类派生出新类，即一个类具有另一个类的属性和行为，又可以有自身独有的特征，减少软件开发的工作量，有助于创建可重用的代码，适合复杂的大型软件的开发。

- 多态性：多态性是指不同的对象调用相同名称的函数时，会根据对象的不同产生不同的

行为。高级程序设计语言 C++ 中的多态性是通过操作符和函数重载、模板和虚函数等方式实现。



注意：

若不理解 `std` 和 `C++`，不必担心，它不会对下面内容的学习产生任何影响。其实，以后的学习也会遇到这样的情况，例如头文件、主函数、`cout`、`cin` 等许多，先把它们作为一种定式使用，随着不断地使用和学习深入，你会豁然开朗。

1.1.2 程序设计过程

1. 程序设计过程

程序设计过程是针对要求解的问题，找出算法，然后再使用程序设计语言编写和测试程序，实现算法的过程。它包括以下几步：

- (1) 明确任务和要解决的问题；
- (2) 找出解决问题的算法；
- (3) 用程序设计语言为算法编写代码；
- (4) 调试和测试代码。

2. 编程的步骤

编程是将所设计的算法变成计算机能够运行的代码的过程。编写一个程序并让程序运行起来，一般包括以下几步：

(1) 编写源代码

使用文本编辑器编写程序，并将它保存到文件中，这个文件就是程序的源代码 (source code)。源代码也叫源程序，是程序员使用计算机高级语言 (例如 C++) 编写的，计算机不能直接运行源代码。

(2) 编译源代码

编译器是一个软件，运行该软件将源代码翻译成计算机能够识别的内部语言——机器语言。经过编译后的程序文件就是程序的目标代码 (object code)，也叫目标程序。

(3) 连接成可执行代码

运行连接程序，将程序的目标代码和该程序使用的函数的目标代码以及一些标准的启动代码 (startup code) 组合起来，生成程序的可执行文件即可执行代码，也叫可执行程序。编程步骤如图 1-3 所示。

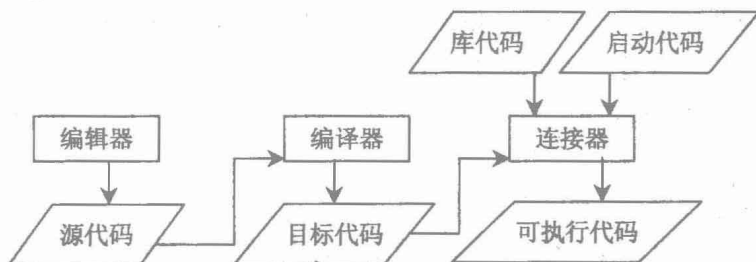


图 1-3 编程步骤

1.2 高级程序设计语言 C++

计算机本身只会完成几十种或上百种不同的简单动作，计算机设计者为每一个动作用一个二进制的编码表示，并为每一个动作设计一种通用的格式，即设计由指令码和内存地址组成的指令。所有指令构成了计算机的指令系统。计算机唯一可以读懂的语言就是计算机的指令，叫做机器语言，被称为低级程序设计语言。程序员把要计算机完成的任务以指令序列的形式写出来，就是机器语言程序设计。

高级程序设计语言，例如 C++，是为解决使用机器语言编写程序困难的问题而逐步发展起来的，其语法符合人的习惯，便于普通用户编程，但计算机却不懂。编译程序（compiler）、编译系统或编译器就能够把用高级语言写出的程序翻译为机器语言的指令序列程序。

有了高级程序设计语言及其编译系统的帮助，人们可以编制出规模更大、结构更复杂的程序。

1.2.1 为什么选择 C++

计算机程序设计课程需要选择教学语言。选择 C++ 作为面向对象程序设计课程的理由：

- OOP 正逐渐成为主流设计技术，无论是在开发领域还是在教学领域，以结构化程序设计为主要特征的 C 和 Pascal 已经不适应形势的发展需要。

- OOP 技术并不取代 SP 和一般程序设计的技巧，C++ 对于 SP 和一般程序设计有较大的兼容性。

- 由于世界各大公司的竞相开发，C++ 语言在各种不同计算机机型上都有优秀的编译系统和相关的开发环境。

- C++ 是主流的软件开发语言，为程序员和软件开发商所接受，并且新的软件开发语言 Java、C# 与 C++ 语言基本一致。

- 国内外出现了一批以 C++ 语言为主，讲授面向对象程序设计技术的教科书，还有一些以 C++ 为主要描述语言的计算机专业的相关教科书，如数据结构、算法设计与分析等。

1.2.2 C、C++ 和 Visual C++

● C 语言

C 语言是一种高级语言，诞生于 20 世纪 70 年代早期，Bell 实验室为开发 UNIX 操作系统，在旧语言的基础上，开发了能将低级语言的效率、硬件访问能力和高级语言的通用性、可移植性融合在一起的 C 语言。C 语言直至今天仍然被广泛应用，C++、Java 和 C# 语言沿用了 C 的大多数语法。

● C++ 语言

C++ 语言也是一种高级语言，在 20 世纪 80 年代同样诞生于 Bell 实验室。C++ 语言是在 C 语言的基础上，引入面向对象的特征，开发出一种过程性与对象性相结合的程序设计语言，最初称为“带类的 C”，1983 年取名为 C++。1998 年国际标准化组织和美国国家标准局制定了 C++ 标准，称为 ISO/ANSI C++，也就是平时所称的 C++。标准 C++ 及其标准库进一步体现了 C++ 语言设计的初衷。

C++语言开发的宗旨是使 OOP 技术和数据抽象成为软件开发者的一种真正实用技术。经过许多次的改进、完善，目前的 C++ 具有两方面的特点：第一，C++ 是 C 语言的超集，与 C 语言兼容，这使得许多 C 代码不经修改就可以经 C++ 编译器进行编译；第二，C++ 支持面向对象程序设计，被称为真正意义上的面向对象程序设计语言。

● C++ 开发工具——Visual C++

Visual C++ 是一款由微软公司开发的软件，是专门负责开发 C++ 软件的工具，称为集成开发环境 (Integrated Development Environment, IDE)。集成开发环境包括编写和修改源代码的文本编辑器、C++ 编译器、连接器、程序调试和运行，以及其他程序开发的辅助功能和工具。通过这个工具，可以大大地提高程序员开发程序的效率。本书采用 Visual C++ 2005 (英文版) 作为开发工具。

1.3 第一个 C++ 程序

本节是为了让读者对 C++ 源程序的组成和在 Visual C++ 2005 中进行程序编辑、编译、连接和运行步骤有一个初步了解，并通过简单介绍 C++ 语言的输入和输出，使读者能够立即上手编写简单的 C++ 程序。

【例 1-1】第一个 C++ 程序。程序的功能是：若用户在键盘上输入问候语“大家好！”，程序会将用户输入的话输出到屏幕上。

完整的程序代码如下：

```
/*
这是我的第一个 C++ 程序——p1.cpp
设计日期：2009 年 8 月
*/
#include<iostream>           //预处理命令
using namespace std;         //命名空间
int main()                   //主函数
{
    char str[10];             //变量
    cout<<"请输入你的问候: "; //输出
    cin>>str;                 //输入
    cout<<"你的问候是: ";     //输出
    cout<<str<<endl;          //输出
    return 0;                 //函数返回
}
```

程序运行时，首先在屏幕上提示一句话“请输入你的问候:”；然后用户在键盘上输入“大家好！”后，按回车键；最后程序将“大家好！”再输出到屏幕上。

1.3.1 C++ 源程序的组成

一个 C++ 程序一般由预处理命令、命名空间、函数、语句、变量、输入/输出和注释等几部分组成。下面通过分析例 1-1 程序的语句含义来了解 C++ 程序的组成。

● 预处理命令

“`#include<iostream>`”是预处理命令，它使程序具有输入/输出功能。C++的预处理命令包括：宏定义命令、文件包含命令和条件编译命令，都是以“#”开始，在后面的章节中将具体讲解。

● 函数

“`int main()`”是程序的主函数。一个C++程序一般由多个函数组成。这些函数可以是用户根据需要编写的自定义函数，也可以是直接使用系统提供的标准库函数。函数体用一对花括号“{”和“}”括起来。任何一个程序必须有且仅有一个主函数 `main`，程序从主函数开始执行。

● 命名空间

“`using namespace std;`”是 `using` 编译指令，表示使用命名空间 `std`。命名空间是为了解决C++中的变量、函数的命名冲突的问题而设置的。解决的办法就是将变量定义在一个不同名字的命名空间中。就好像张三有一台电脑，李四也有一台同样的电脑，但能够清楚地区分这两台电脑，就是让他们分属于不同的人。上面例子中的 `main` 函数需要能够访问位于命名空间 `std` 的“`cin`”、“`cout`”和“`endl`”定义。

事实上，“`using namespace std;`”也可以放在 `main` 函数里面。在多函数程序中，为了使多个函数都能够访问命名空间 `std`，只需将“`using namespace std;`”放在所有函数定义之前。本书采用的就是这种方法。

如果不使用“`using namespace std;`”，在需要使用命名空间 `std` 中的元素时，可以使用前缀“`std::`”，例如：

```
std::cout<<str<<std::endl;
```

● 语句

语句是函数的基本组成单元，语句有顺序语句、选择结构和循环结构等三种结构。任何一条语句以“;”结束。本程序只有一个函数——主函数（`main`），它的语句是用花括号“{”和“}”括起来的多条语句。

● 变量

程序中的“`char str[10];`”是变量定义语句。`char` 表示变量的数据类型是字符型，`str[10]` 表示 `str` 是有 10 个字符型变量的数组。

C++程序需要将数据放在内存单元中，变量就是内存单元中数据的标识符，通过变量来存储和访问相应的数据。

● 输入/输出

C++程序中的输入语句用来接受用户的输入，输出语句用来向用户返回程序的运行结果。

程序中的“`cout<<“请输入你的问候:”;`”是输出语句，用 `cout`（读 C-Out）将用两端加上双引号表示的一个字符串“请输入你的问候:”输出到屏幕上。语句的中“<<”，叫插入运算符，不能省略。输出语句“`cout<<str<<endl;`”的功能是将数组 `str` 中的字符串输出到屏幕上，然后再在屏幕上输出一个换行，即 `endl`。

程序中的“`cin>>str;`”是输入语句，用 `cin`（读 C-In）将用户通过键盘输入的字符串放在数组 `str` 中。语句的中“>>”，叫提取运算符，也不能省略。

关于 `cin` 和 `cout` 我们可以直接使用，后面还有章节专门介绍。