

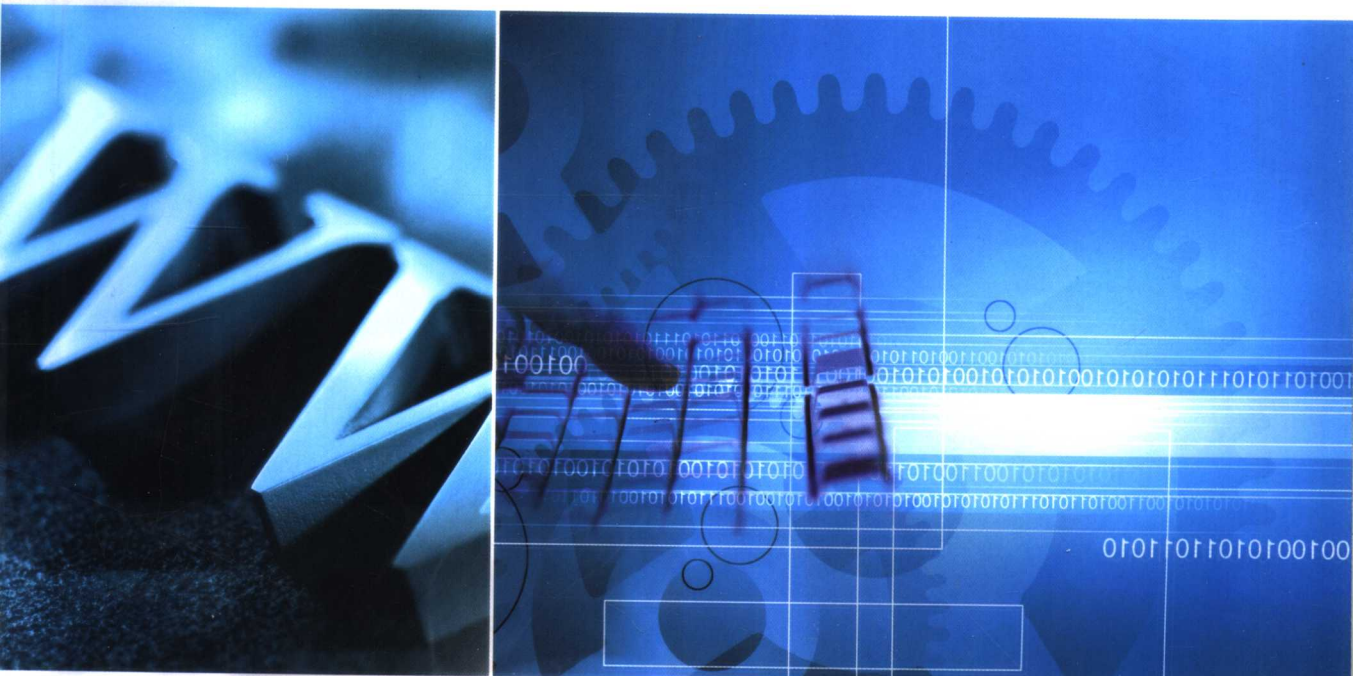


软件职业技术学院“十一五”规划教材

C++

面向对象程序设计

曹 静 主 编 董 宁 陈 丹 副主编



中国水利水电出版社
www.waterpub.com.cn

软件职业技术学院“十一五”规划教材

C++面向对象程序设计

曹 静 主 编

董 宁 陈 丹 副主编

中国水利水电出版社

内 容 提 要

本书系统地介绍了面向对象程序设计的概念和 C++ 语言的基础知识。全书将编程语言和编程思想融为一体,使读者在一步步学习 C++ 语言的同时,可以一层层掌握面向对象的编程原理。为方便读者理解 C++ 面向对象程序设计的相关知识,书中采用通俗易懂的方式介绍了有关理论和原理,并辅以大量实例和插图。每章均配有相关内容的项目设计,便于读者进行实际操作和练习。

本书内容翔实、叙述清晰、注重实践、习题丰富,可作为高职高专各个专业的计算机入门教材,也适合各领域想在计算机方面进修提高的人员自学使用。

本书电子教案及全部实例源代码可从中国水利水电出版社网站免费下载,网址:
[http:// www.waterpub.com.cn/softdown/](http://www.waterpub.com.cn/softdown/)。

图书在版编目(CIP)数据

C++面向对象程序设计 / 曹静主编. —北京:中国水利水电出版社, 2006
(软件职业技术学院“十一五”规划教材)
ISBN 7-5084-3874-4

I. C… II. 曹… III. C 语言—程序设计—高等学校:技术学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字 (2006) 第 089439 号

书 名	C++面向对象程序设计
作 者	曹 静 主 编 董 宁 陈 丹 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 20 印张 490 千字
版 次	2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷
印 数	0001—4000 册
定 价	28.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

序

随着信息技术的广泛应用和互联网的迅猛发展,以信息产业发展水平为主要特征的综合国力竞争日趋激烈,软件产业作为信息产业的核心和国民经济信息化的基础,越来越受到世界各国的高度重视。中国加入世贸组织后,必须以积极的姿态,在更大范围和更深程度上参与国际合作和竞争。在这种形势下,摆在我们面前的突出问题是人才短缺,计算机应用与软件技术专业领域技能型人才的缺乏尤为突出,无论是数量还是质量,都远不能适应国内软件产业的发展和信息化建设的需要。因此,深化教育教学改革,推动高等职业教育与培训的全面发展,大力提高教学质量,是迫在眉睫的重要任务。

2000年6月,国务院发布《鼓励软件产业和集成电路产业发展的若干政策》,明确提出鼓励资金、人才等资源投向软件产业,并要求教育部门根据市场需求进一步扩大软件人才培养规模,依托高等学校、科研院所,建立一批软件人才培养基地。2002年9月,国务院办公厅转发了国务院信息化工作办公室制定的《振兴软件产业行动纲要》,该《纲要》明确提出要改善软件人才结构,大规模培养软件初级编程人员,满足软件工业化生产的需要。教育部也于2001年12月在35所大学启动了示范性软件学院的建设工作,并于2003年11月启动了试办示范性软件职业技术学院的建设工作。

示范性软件职业技术学院的建设目标是:经过几年努力,建设一批能够培养大量具有竞争能力的实用型软件职业技术人才的基地,面向就业、产学结合,为我国专科层次软件职业技术人才培养起到示范作用,并以此推动高等职业技术教育人才培养体系与管理体制和运行机制的改革。要达到这个目标,建立一套适合软件职业技术学院人才培养模式的教材体系显得尤为重要。

高职高专的教材建设已经走过了几个发展阶段,由最开始本科教材的压缩到加大实践性教学环节的比重,再到强调实践性教学环节,但是学生在学习时还是反映存在理论与实践的结合问题。为此,中国水利水电出版社在经过深入调查研究后,组织了一批长期工作在高职高专教学一线的老师,编写了这套“软件职业技术学院‘十一五’规划教材”,本套教材采用项目驱动的方法来编写,即全书所有章节都以实例作引导来说明各知识点,各章实例之间并不是孤立的,每个实例都可以作为最终项目的一个组成部分;每一章章末还配有实习实训(或叫实验),这些实训组合起来是一个完整的项目。

采用这种方式编写的图书与市场上同类教材相比更具优越性,学生不仅仅学到了知识点,还通过项目将这些知识点连成一条线,开拓了思路,掌握了知识,达到了面向岗位的职业教育培训目标。

本套教材的主要特点有:

- (1) 课程主辅分明——重点突出,教学内容实用。
- (2) 内容衔接合理——完全按项目运作所需的知识体系结构设置。

(3) 突出实习实训——重在培养学生的专业能力和实践能力,力求缩短人才与企业间的磨合期。

(4) 教材配套齐全——本套教材不仅包括教学用书,还包括实习实训材料、教学课件等,使用方便。

本套教材适用于广大计算机专业和非计算机专业的大中专院校的学生学习,也可作为有志于学习计算机软件技术与开发的工程技术人员的参考教材。

编委会

2006年7月

前 言

C++语言是由AT&T贝尔实验室的Bjarne Stroustrup建立的一种功能强大的高级编程语言。选择C++讲授编程是因为它支持面向对象的方法开发软件。面向对象方法的优点是：我们可以用它来开发极为复杂的软件系统，而且它所使用的许多技术是与建造复杂的物理系统，如制造船舶、飞机或盖大楼时所使用过的技术类似。本书的目标是把它作为大学生学习编程的第一门课程，而且设计成适合所有专业的学生。作者假定他们预先没有学过任何编程语言，而所用的数理知识也仅限于大学一年级水平。

本书是全国示范性软件职业学院计算机及其相关专业指定教材，针对全国示范性软件职业学院的特点，淡化理论，够用为度，强化技能，重在实际操作，以项目设计贯穿全书，再辅以大量实例，让学生能在实践中掌握C++语言和面向对象的软件开发方法。

本书是作者在多年教学实践和科学研究的基础上，参阅了大量国内外相关教材后，几经修改而成的，主要特点如下：

(1) 知识点全面，语言严谨、精练。

C++中概念众多且比较抽象，本书对这些概念进行了清楚准确地解释并结合实例说明，让读者能全面掌握每一个知识点。

(2) 实际项目开发与理论教学紧密结合。

为了使读者能快速地掌握C++相关知识的使用方法，本书在各个章节的重要知识点后面都附有典型的实训，本书最后一章更结合每章中的实训内容给出一个完整的项目。

(3) 合理、有效的组织。

本书按照由浅入深的顺序，循序渐进地介绍了C++面向对象程序设计的相关知识。各个章节在编写时都是层层展开、环环相扣的。

(4) 内容充实、实用。

本书不再使用传统教材中那些最多只有数百行组成且在实践中极少使用的示例程序，各章的实例都是实用性很强的程序或程序模块，并且每章实例都是作为一个大型软件项目的一部分出现，实例代码可维护性强、重用率高，真正体现了面向对象程序开发的思想。

(5) 本书配有全部的程序源文件和电子教案。

为方便读者使用，书中全部实例的源代码及电子教案均免费赠送给读者。

本书共分四大部分，第一部分包括第1章至第3章，主要介绍C++基本语法概念和程序结构；第二部分包括第4章至第6章，主要介绍类、类的封装、继承和多态等；第三部分包括第7章至第9章，主要介绍I/O流、模板和异常处理；第四部分包括第10章，是一个综合全书内容的实训项目。

本书由曹静任主编，董宁、陈丹任副主编，张宇、文凡、刘嵩、刘洁、肖英参加编写，

董宁、陈丹、库波统编全稿。

由于时间仓促，加之编者水平有限，书中不妥或错误之处在所难免，恳请广大读者批评指正，作者联系方式：jingcao_2002@163.com。

编者

2006年5月

目 录

序

前言

第 1 章 面向对象程序设计概述	1
1.1 面向对象的基本概念	1
1.2 C++面向对象程序设计	2
1.3 C++程序的基本组成	3
1.4 Visual C++ 6.0 开发环境	6
【实训 1】 简单的 C++程序实例	10
本章小结	11
习题一	11
第 2 章 C++语言基础	12
2.1 数据类型和基本输入输出	12
2.1.1 基本数据类型	12
2.1.2 常量	13
2.1.3 变量	16
2.1.4 基本输入/输出	17
2.2 运算符和表达式	18
2.2.1 算术运算符和算术表达式	18
2.2.2 赋值运算符和赋值表达式	19
2.2.3 逗号运算符和逗号表达式	20
2.2.4 关系运算符和关系表达式	20
2.2.5 逻辑运算符和逻辑表达式	20
2.2.6 ++、--与 sizeof 运算符	21
2.2.7 位运算符	22
2.2.8 三目运算符	24
2.2.9 运算符的优先顺序	24
2.2.10 运算式中的类型转换	25
2.3 基本控制结构	26
2.3.1 表达式语句、空语句和复合语句	27
2.3.2 选择控制语句	28
2.3.3 循环控制语句	32
2.3.4 break 和 continue 语句	35
【实训 2】 程序控制结构的应用	37

2.4	函数	39
2.4.1	函数的定义和调用	39
2.4.2	函数的参数传递	40
2.4.3	函数参数的引用传递	41
2.4.4	带默认形参值的函数	43
2.4.5	函数的嵌套与递归调用	43
2.4.6	内联函数	46
2.4.7	函数的重载	47
	【实训 3】 函数的应用	48
2.5	构造类型	50
2.5.1	数组与字符串	50
2.5.2	结构体	56
2.5.3	共用体	60
2.5.4	枚举类型	61
2.5.5	用 typedef 定义类型	63
2.6	指针	63
2.6.1	指针和指针变量	63
2.6.2	&和*运算符	64
2.6.3	指针的赋值与运算	64
2.6.4	指针和数组	66
2.6.5	指针和结构体	70
2.6.6	指针和函数	71
2.6.7	动态内存分配 new 和 delete	75
	【实训 4】 指针的综合应用	76
	项目设计 1 控制台程序的人机界面设计	83
	本章小结	87
	习题二	88
第 3 章	C++程序的结构	91
3.1	作用域和可见性	91
3.1.1	作用域	91
3.1.2	可见性	92
3.2	生存期	93
3.2.1	静态生存期	93
3.2.2	动态生存期	93
3.3	变量的存储类型	94
3.3.1	局部变量	95
3.3.2	全局变量	96

3.3.3 外部变量	97
3.4 编译预处理和多文件结构	98
3.4.1 编译预处理命令	98
3.4.2 多文件组织结构	101
项目设计 2 预处理及多文件结构的编译	103
本章小结	107
习题三	107
第 4 章 类与对象	110
4.1 面向对象程序设计的基本特点	110
4.1.1 抽象	110
4.1.2 封装	111
4.1.3 继承	112
4.1.4 多态	113
4.2 建立类与对象	114
4.2.1 类的声明与实现	114
4.2.2 类成员的访问控制	115
4.2.3 类的成员函数	116
4.2.4 对象的建立与使用	119
4.3 构造函数和析构函数	121
4.3.1 构造函数	121
4.3.2 析构函数	127
4.4 类的作用域和对象的生存期	128
4.4.1 类的作用域	128
4.4.2 对象的生存期	129
4.5 类的静态成员	130
4.5.1 静态数据成员	130
4.5.2 静态函数成员	133
【实训 5】 对象与类成员的应用	136
4.6 常类型	138
4.6.1 常引用	138
4.6.2 常对象	139
4.6.3 用 const 修饰的类成员	139
4.7 数组、指针与对象	142
4.7.1 对象数组	142
4.7.2 对象指针	144
4.7.3 动态配置对象内存	145
4.7.4 浅拷贝和深拷贝	147

4.7.5 this 指针	152
4.8 类的组合	152
4.8.1 类的组合	152
4.8.2 前向引用声明	157
【实训 6】 对象数组与指针的综合应用	157
4.9 友元	161
4.9.1 友元函数	161
4.9.2 友元类	164
项目设计 3 面向对象程序设计中类的应用	165
本章小结	173
习题四	173
第 5 章 类的继承与派生	175
5.1 继承与派生	175
5.1.1 基类与派生类	175
5.1.2 派生类的声明	176
5.1.3 派生类生成过程	177
5.2 继承的三种方式	178
5.2.1 公有继承	179
5.2.2 私有继承	181
5.2.3 保护继承	184
5.3 派生类的构造函数和析构函数	186
5.3.1 构造函数	186
5.3.2 析构函数	189
5.4 多继承和虚基类	191
5.4.1 派生类成员的标识与访问	191
5.4.2 作用域分辨	192
5.4.3 多继承和虚基类	192
5.4.4 虚基类及其派生类的构造函数	195
5.5 赋值兼容规则	196
【实训 7】 类继承与派生的应用	198
项目设计 4 类的组织结构在校园信息管理系统中的应用	202
本章小结	211
习题五	211
第 6 章 多态性与虚函数	212
6.1 多态性概述	212
6.1.1 多态的类型	212
6.1.2 多态的实现	212

6.2 重载与再定义	213
6.2.1 函数的重载	213
6.2.2 函数的再定义	214
6.3 运算符重载	216
6.3.1 运算符重载的规则	216
6.3.2 运算符重载为成员函数	217
6.3.3 运算符重载为友元函数	222
6.4 虚函数	224
6.4.1 一般虚函数成员	224
6.4.2 虚析构造函数	226
6.5 抽象类	227
6.5.1 纯虚函数	227
6.5.2 抽象类	227
【实训 8】 抽象基类及虚函数的应用	229
项目设计 5 多态在校园信息管理系统中的应用	233
本章小结	243
习题六	243
第 7 章 模板	244
7.1 模板的概念	244
7.2 函数模板	244
7.3 类模板	246
项目设计 6 顺序表模板	250
本章小结	254
习题七	254
第 8 章 数据流与文件的输入/输出	256
8.1 I/O 流的概念	256
8.2 数据流的控制	257
8.2.1 数据流的格式控制	257
8.2.2 常用的输入/输出函数	261
8.2.3 数据流的错误侦测	262
8.3 文件的输入/输出	262
8.3.1 fstream 类	262
8.3.2 格式化文字模式的文件 I/O	264
8.3.3 二进制模式的文件 I/O	266
8.3.4 文件指针的使用	267
【实训 9】 利用文件存储对象	270
项目设计 7 校园信息管理系统中的数据存储	272

本章小结	279
习题八	280
第 9 章 异常处理	281
9.1 异常处理的基本思想	281
9.2 C++异常处理的实现	281
9.2.1 异常处理的机制	281
9.2.2 异常处理的规则	283
9.3 异常处理中的构造与析构	285
本章小结	287
习题九	287
第 10 章 项目设计——校园信息管理系统	288
10.1 系统分析与设计	288
10.2 程序设计分析	288
10.3 源程序及说明	290
10.4 运行结果与分析	303
参考文献	307

第1章 面向对象程序设计概述

教学要求

- 掌握面向对象的基本概念、面向对象程序设计和结构化程序设计的区别。
- 理解 C++面向对象程序设计的特点、一个基本 C++程序的组成、Visual C++ 6.0 的开发环境。
- 了解计算机程序设计语言的发展过程。

1.1 面向对象的基本概念

计算机程序设计语言是计算机可以识别的语言，供计算机阅读和执行，用于描述解决问题的方法。计算机程序就是用计算机程序设计语言所要求的规范书写出来的一系列动作，它表达了程序员要求计算机执行的操作。

最早使用的计算机程序设计语言是机器语言，它是由计算机硬件系统可以识别的二进制指令组成的语言。不久出现了汇编语言，它是一种用符号表示的、面向某一特定机型的程序设计语言，它的指令与机器指令一一对应。为了提高编制程序的效率和程序的可读性，FORTRAN、COBOL、Pascal、C、C++等高级语言相继出现。

高级程序设计语言的出现屏蔽了计算机硬件的细节，提高了语言的抽象层次，程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这就使软件设计师在书写程序时可以联系到程序所描述的具体事物，从而大大提高了编写程序的效率。

传统软件开发使用的是结构化程序设计方法，把程序看成是处理数据的一系列过程，采用了模块分解与功能抽象以及自顶向下、分而治之的方法，从而有效地将一个较复杂的程序设计任务分解成许多易于控制和处理的子任务，便于开发和维护，成功地处理复杂问题提供了有力手段。

但是当程序需要处理的数据量增大时，数据与处理这些数据的方法之间的分离使程序变得越来越难以理解。当对某段程序进行修改或删除时，整个程序中所有与之相关的部分都要进行相应的修改，从而使程序代码的维护变得困难。对数据处理能力的需求越强，这种分离所造成的负作用越显著。为了克服结构化程序模式在设计系统软件和大型应用软件时所存在的缺陷，面向对象模式应运而生。

结构化程序设计是以功能为中心来描述系统，而面向对象程序设计是围绕现实世界的概念来组织模块，以数据为中心来描述系统。面向对象程序设计把一个复杂的问题分解成多个能够完成独立功能的对象（类），用对象来描述问题空间的实体，用程序代码模拟现实世界中真实或抽象的对象，使程序设计过程更自然、更直观。

面向对象程序设计的本质是把数据及对数据的操作方法放在一起，作为一个相互依存、不可分离的整体——对象。对同类型对象抽象出其共性，形成类。类中的大多数数据，只能用

本类的方法进行处理。类通过一个简单的外部接口与外界发生关系，对象与对象之间通过消息进行通信。这样，程序模块间的关系更为简单，程序模块的独立性、数据的安全性就有了良好的保障。

面向对象程序设计方法具有 4 个基本特征：抽象、封装、继承和多态性。

抽象（Abstract）就是忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性，把具有共同性质的事物划分为一类，得出一个抽象的概念。面向对象方法中的“类”为具有相同属性和行为的一组对象提供了抽象的描述，一个属于某类的对象称为该类的一个实例。

封装（Encapsulation）就是把每个对象的数据（属性）和操作（行为）包装在一个类中，并尽可能隐藏对象的内部细节。一般限制直接访问对象的属性，而应通过操作接口访问，这样使程序中模块之间的关系更简单、数据更安全，对程序的修改也仅限于类的内部，使得由于修改程序所带来的影响局部化。

继承（Inheritance）是指一个新类可以从现有的类派生而来。新类继承了现有类的特性，包括一些属性和行为，并且可以修改或增加新的属性和行为，使之适合具体的需要。继承是面向对象技术能够提高软件开发效率的重要原因之一，很好地解决了软件的可重用性问题。

多态性（Polymorphism）是指允许不同类的对象对同一消息作出的响应不同。一般类中定义的属性或行为被特殊类继承后可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或行为在一般类及其各个特殊类中具有不同的语义。

通过使用面向对象程序设计的抽象、封装、继承和多态性等机制，程序更易于维护、更新和升级。利用代码可重用性，程序员可以在程序中大量使用成熟的类库，从而缩短程序的开发时间，提高程序员的工作效率和程序的可靠性。

1.2 C++面向对象程序设计

面向对象程序设计语言很多，如 Smalltalk、Ada、Eiffel、Object Pascal、Visual Basic、C++ 等。作为 C 语言的继承者，C++ 语言首先继承了 C 语言的以下特点：丰富的运算符和数据类型、结构化的程序设计方法、高效的机器代码、良好的可移植性。同时，由于 C 是 C++ 的子集，这使得许多 C 语言代码不需要修改就可以被 C++ 编译器通过。C++ 的错误检查机制强，它提供了专门的机制检查类，更适合大中型程序的开发。

C++ 语言扩展了 C 语言的功能，它引入了一些新特性，强调代码的有效性和紧凑性。在程序设计方法方面，C++ 是一种混合型程序设计语言，它既支持传统的结构化程序设计，又支持现在的面向对象程序设计。C++ 既适合作为系统软件的设计语言，又适于编写应用软件。C++ 的目标是为程序员的程序开发活动提供一个优良的设计工具，以编写模块化程度高、可重用性和可维护性俱佳的程序，因此，可以说 C++ 是程序员的语言。

本书介绍的主要就是使用最为广泛的 C++ 语言和使用 C++ 语言进行面向对象程序设计的方法。在学习编程之前，首先简单了解一下 C++ 程序的开发过程。在后续章节的学习和编程实践中，读者将对它们有不断深入的理解。

程序的开发通常要经过编辑、编译、连接、运行调试这几个步骤。编辑是将源程序输入到计算机中，生成后缀为 .cpp 的磁盘文件。编译（Compile）是将程序的源代码转换成机器语

言代码。程序员编制的源程序被编译后，会生成一个目标文件，这个文件通常以.obj 作为文件扩展名，该目标文件为源程序的目标代码。但是编译后的程序还不能由计算机执行，因为目标代码只是一个个程序块，需要相互衔接成为一个适应一定操作系统环境的程序整体。为了把它转换成可执行程序，必须进行连接（link）。C++程序通常是通过同时连接一个或多个目标文件以及一个或几个库而创建的。库（.lib）是一组由机器指令构成的程序代码，是可连接文件。库有标准库和用户生成的库。标准库是由 C++提供的，用户生成的库是由软件开发商或程序员提供的。文件与库连接的结果，即生成一个后缀为.exe 的计算机可执行程序。

程序员首先在集成开发环境中编辑源程序，或在其他编辑器中输入源程序，然后在集成环境中启动编译程序将源程序转化为目标文件。在编译和连接时，都会对程序中的错误进行检查，并将查出的错误显示在屏幕上。编译阶段查出的错误是语法错误，连接时查出的错误称为连接错误。如果出现编译错误或连接错误，程序员必须回到编辑状态重新开始编辑程序和编译、连接的过程，直到排除所有错误、成功生成可执行文件为止。

1.3 C++程序的基本组成

为了使读者能更加直观地认识 C++语言，下面就从最简单的例子来分析 C++程序的构成。例 1-1 的程序代码就是一个完整的可运行的 C++程序，它可以输出两个句子。

【例 1-1】第一个 C++程序。

```
//Ch01_01.cpp
#include <iostream>
using namespace std;
void main()
{
    cout<< "Hello,World!\n ";
    cout<< "I'm a student.\n ";
}
```

程序运行结果为：

```
Hello,World!
I'm a student.
```

一个 C++程序由三部分组成：注释部分、编译预处理和程序主体。

程序的注释部分是程序员为读者作的说明，是提高程序可读性的一种方法。一般将注释分为序言注释和注解性注释两种。前者用于程序的开始部分，用以说明程序或文件的名称、功能、编写的时间、作者以及输入输出说明等，而后者用于程序主体中，对难以理解的语句作说明。

C++程序中，有两种给出注释的方法。一种是沿用 C 语言的方法，使用“/*”和“*/”括起注释文字，它可以占多行。例如：

```
/* This is my first program.
   This is comment.
  */
```

这里“/*”和“*/”之间的内容都会作为注释处理。而另一种方法就是例 1-1 中的注释，它是从“//”开始直至换行，中间的部分就是注释。

注释部分仅仅用于作为说明,提高程序的阅读质量,并不是程序的必要部分,它是可选的。在生成可执行程序之前,C++的编译系统在对源程序进行编译时不处理注释部分,因此注释部分对程序的功能实现不起任何作用。而且由于编译系统忽略注释部分,所以注释部分也不会增加最终产生的可执行程序的大小。

每个以符号“#”开头的行,称为编译预处理行。如本例中的“#include”称为编译指令。而“#include <iostream>”指示编译器在对程序进行预处理时将文件 iostream.h 中的代码嵌入到程序中该指令所在的地方。文件 iostream.h 中设置了 C++ 的 I/O 相关环境,定义了输入输出流对象 cin 和 cout 等。由于这类文件常被嵌入在程序的开始处,所以称之为头文件。在 C++ 程序中如果使用了系统中提供的一些功能,就必须嵌入相关的头文件。using namespace std; 表示使用 C++ 标准中定义的 std 类库中的头文件。

在程序接下来的部分中,main 是主函数名,函数是 C++ 程序中最小的功能单位,每一个 C++ 程序都必须有一个 main() 函数,并且 main() 表明了程序执行的开始点。main() 函数之前的 void 表示 main() 函数没有返回值。函数体用一对大括号 {} 包含。用程序语句来描述一个函数所执行算法的过程称为函数定义。在这个例子中,main() 函数头和函数体构成了一个完整的函数定义。

程序由语句组成,每条语句都以分号作为语句的结束符。在 main() 函数体中,cout 是一个代表标准输出的流设备,它是 C++ 系统预定义的对象,其中包含了用于输出的功能。当程序要在设备上输出时,就需要在程序中指定该对象。输出操作由操作符“<<”来表达,其作用是将紧随其后的双引号中的字符串输出到标准输出设备上。

程序中用双引号括起的部分“Hello,World!\n”称为字符串常量,其中的字符“\n”表示一个换行控制符。

下面再看一个进行简单计算的程序。

【例 1-2】一个进行加法计算的 C++ 程序。

```
// Ch01_02.cpp
#include <iostream>
using namespace std;
void main()
{
    int m,n,t;
    cout<<" Please input two integers: \n";
    cin>>m>>n;
    t=m+n;
    cout<<"The result is "<<t<<endl;
}
```

程序运行结果为:

```
Please input two integers:
35 62 ✓
The result is 97
```

与例 1-1 的程序一样,例 1-2 也是从 main() 开始运行。在 C++ 程序中,每一个变量都需要在声明后才可以使⽤,所以首先在程序中定义将会用到的变量,int m,n,t; 表示分别定义了 m、n、t 三个整型(int) 变量,int 是 C++ 语言提供的标准数据类型。在定义变量时要在变量的前