HZ BOOKS
华章教育

高等学校计算机专业规划教材

C/C++程序设计

第2版

宋晓宇 主编
赵艳平 副主编　杨艳春 李世伟 张洁 编著

Programming in C/C++
Second Edition

机械工业出版社
China Machine Press

高等学校计算机专业规划教材

C/C++程序设计

第2版

宋晓宇 主编

赵艳平 副主编　杨艳春 李世伟 张洁 编著

Programming in C/C++

Second Edition

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

C/C++ 程序设计 / 宋晓宇主编 . —2 版 . —北京：机械工业出版社，2017.8
（高等学校计算机专业规划教材）

ISBN 978-7-111-57700-3

I. C… II. 宋… III. C 语言－程序设计－高等学校－教材 IV. TP312.8

中国版本图书馆 CIP 数据核字（2017）第 193371 号

本书针对初学者的特点，采用“提出问题－分析问题－解决问题－归纳提高”的教学模式，突出对学习者计算思维、编程实践能力的培养与训练。全书共 12 章，全面系统地介绍 C/C++ 语言的基本概念、语法及程序设计方法，详细地讲解 C/C++ 中的数据类型、运算符与表达式、基本控制语句、数组、函数、指针、类和对象、继承和派生、输入输出流等内容。

本书定位准确、结构合理、例题丰富，符合学习者的认知规律，适合作为高校 C/C++ 程序设计基础课的教材，也可作为工程技术人员、自学人员及参加全国计算机等级考试（二级 C/C++ 语言程序设计）人员的参考书。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：余　洁　　责任校对：李秋荣
印　　刷：北京文昌阁彩色印刷有限责任公司　　版　　次：2017 年 8 月第 2 版第 1 次印刷
开　　本：185mm×260mm　1/16　　印　　张：19.75
书　　号：ISBN 978-7-111-57700-3　　定　　价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

前 言

程序设计是高等学校计算机基础教育的重要内容和入门课程，C/C++ 语言以功能丰富、表达能力强、应用面广等特点，在整个计算机基础教育课程设计中占有重要地位。2013 年，根据教育部《关于进一步加强高等学校计算机基础教学的意见》的要求，参考《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》，我们编写了《 C/C++ 程序设计》第 1 版。

本书第 1 版在使用过程中受到广大读者的广泛好评，但在实际教学中我们发现内容编排上存在一定的问题，导致学生对一些知识的理解出现困难和偏差。针对上述问题，第 2 版进行了修订，在内容与顺序上进行了调整、改进和补充，并针对参加程序设计等级考试的读者增加了重点章节的课后习题，具体调整如下。

1. 第 1 版的“第 1 章绪论”拆分成“C++ 概述”和“算法”两章

重要的知识点单独成章，每章内容相对独立，与其他知识点关联少，条理清楚，易于初学者掌握。

2. 部分章节顺序调整

- 数组和指针的顺序问题。一般教材都是先讲数组，再讲指针。带来的问题就是无法对数组名进行解释，于是产生了“数组名是一个地址”的错误说法，实际上数组名在多数情况下都是一个指针。在不介绍指针的情况下，很难把数组一章的内容讲清、讲透，不易于学生理解。
- 数组和函数的顺序问题。一般教材都是把函数放在数组之后讲解，原因是便于把数组名作为参数放在函数一章中。看起来似乎恰当，但是这样一来就掩盖了函数一章的重点。函数一章最应该教给学生的是如何把函数设计得当，以便于其他函数调用，只要突出这一重点就可以了。

综上所述，最合适的顺序安排应是指针、函数、数组、指针总结。

3. 化繁为简、化整为零

对第 1 版的第 10 章和第 11 章的内容进行整合、优化，合成一章。

本书注重对学生编程实践和问题求解能力的培养，以 Visual C++ 为平台，在学习 C/C++ 语言基础知识的同时，通过大量精选的例题和习题将程序设计的基本思想和方法介绍给学生。全书共分 12 章，涉及 C/C++ 的基本数据类型、运算符和表达式、程序控制结构、数组和字符串处理、函数和模块化程序设计、指针、结构体和共用体、面向对象程序设计基础等。第 1～10 章以面向过程程序设计方法为出发点，介绍了 C/C++ 语言和程序设计的基础知识。第 11、12 章是面向对象程序设计方法基础，介绍了 C/C++ 语言中类和对象、继承和派生以及 C++ 流类库等知识。全书在内容安排上实现了从结构化程序设计方法到面向对象程序设计方法的过渡，注重知识的系统性和连贯性。

本书由宋晓宇主编，赵艳平担任副主编，其中杨艳春编写了第1～3章，赵艳平编写了第4、5、8章及附录部分，宋晓宇编写了第6、10、12章，李世伟编写了第7、9章，张洁编写了第11章。在本书的编写过程中，兰州交通大学电信学院软件工程系的教师们给予了大力支持与帮助，在此表示衷心的感谢！

因编者水平有限，书中难免有错误和不妥之处，敬请专家和读者提出宝贵意见，编者邮箱：sxy9998@126.com。

编　者

2017年6月

教学建议

教学章节	教学要求	课 时
第 1 章 C++ 概述	了解程序设计方法及 C++ 语言的发展 掌握 C++ 简单程序的编写	1
	掌握 C++ 程序的上机步骤	2
第 2 章 算法	理解算法的概念 熟练掌握流程图的表示方法	1
第 3 章 C++ 语言基础知识	掌握 C++ 语言的基本数据类型、运算符及表达式	6
第 4 章 顺序结构程序设计	掌握 C++ 语言数据的输入与输出 熟练掌握顺序结构程序设计	4
第 5 章 选择结构程序设计	熟练掌握 if 语句的 3 种形式、if 语句的嵌套 掌握多分支选择 switch 语句	6
第 6 章 循环结构程序设计	熟练掌握 while 循环、do-while 循环及 for 循环 掌握 break、continue 语句 掌握循环的嵌套	10
第 7 章 指针	掌握指针的定义及使用 掌握多级指针的定义及使用	2
第 8 章 函数与编译预处理	掌握函数的定义及调用方法 理解变量的存储类别 掌握编译预处理命令的使用	8
第 9 章 数组	熟练掌握一维数组、二维数组的定义及使用 掌握字符数组、字符变量的定义及使用 掌握用指针变量访问数组的方法，了解指针数组	10
第 10 章 自定义数据类型	掌握结构体变量、共用体的定义和使用 了解链表及其基本操作 掌握类型定义符 typedef 的使用	6
第 11 章 面向对象程序设计基础	掌握类的定义和对象的声明 理解构造函数和析构函数 掌握类成员访问权限和访问方法 掌握类的继承及其实现 掌握对象动态创建与撤销的方法	6
第 12 章 C++ 语言的流类库	了解 C++ 语言的流类库 掌握文件流、字符串流及其基本操作	2

说明：1）建议课堂教学全部在多媒体机房内完成，实现讲、练结合。

2）建议将教学分为核心知识技能模块和技能提高模块，其中核心知识技能模块建议教学学时为 64～80 学时，技能提高模块建议教学学时为 16～24 学时，不同学校可以根据各自的教学要求和计划学时数对教学内容进行取舍。

目 录

第1章　C++概述

【本章要点】

- 程序设计语言。
- 程序设计方法。
- C++语言程序开发过程及上机调试步骤。

自然语言（如汉语）是人类交流和表达思想的工具，自从人类进入信息社会以来，计算机作为人类的得力助手进入了社会生活的方方面面。于是人类怎样驾驭计算机，如何与计算机进行“交流”成为一个关键问题。其实，人与计算机“交流”，也需要借助一种“语言”来实现，这种语言就是计算机语言，它是一种人与计算机双方都能理解的语言。计算机语言由一套固定的符号和语法规则组成。人们要利用计算机解决问题，就必须用计算机语言来告诉计算机“做什么”和“怎样做”，这个过程就是程序设计过程。因此，计算机语言是人与计算机进行信息交流的工具，也称为程序设计语言或编程语言。

1.1　程序设计语言概述

程序设计语言（programming language）是一组用来编写计算机程序的符号语言，它由一组预先定义的指令、数据类型、语法规则构成。人类使用程序设计语言描述解决问题的方法（即编写程序），供计算机阅读和执行，达到借助计算机解决特定问题的目的。从计算机问世以来，程序设计语言也在不断地发展变化着，按其发展过程计算机程序设计语言可分为机器语言、汇编语言和高级语言。

1.1.1　机器语言

机器语言是最早使用的编程语言，它对应计算机的指令系统或指令集，是最底层的程序设计语言。在机器语言编写的程序中，每一条指令都是二进制形式的指令代码，计算机硬件可以直接识别。机器语言是面向机器的，不同的计算机硬件（主要是CPU体系结构差异）其机器语言是完全不同的，因此机器语言的通用性很差。

由于机器语言程序是直接针对计算机硬件编写的，因此它的执行效率比较高，可以充分发挥计算机的性能。但是，用机器语言编写程序，即使是一个简单的算式，也要使用很多条指令才能表达出来，因而难以阅读、理解、记忆，且不易查错。

机器语言的特点：

1）机器语言是计算机能够直接识别并执行的唯一一种语言。

2）机器语言是直接面向机器的语言。

3）机器语言的指令记忆、程序编写和阅读等都很困难。

1.1.2　汇编语言

20世纪50年代中期，人们开始用一些“助记符”来代替机器语言指令中的操作符和操作数，这种用“助记符”表示的语言称为汇编语言。汇编语言是一种符号语言，它在一定

程度上解决了机器语言指令难以记忆、编程困难的问题，但它和机器语言一样也是面向机器的。使用汇编语言编写的程序，计算机不能直接执行，必须借助一种工具将它翻译成机器语言目标程序，这种工具就称为汇编程序，翻译的过程称为汇编。

使用机器语言和汇编语言都需要对计算机的内部结构有较深刻的了解，因此用它们编写程序不仅劳动强度大，而且程序的通用性差。通常将这两种语言称为“计算机低级语言”。

汇编语言的优点：

1）能够直接对硬件（寄存器、存储单元和I/O端口）进行操作，并利用硬件的特点和指令系统提供的各种寻址方式编写出高质量的程序。这种程序占用内存少，执行速度快，因此多用于编写系统程序和实时处理程序，并经常被高级语言所嵌入使用。

2）能够直接对位、字节、字、字节串以及字串等数据类型进行操作，为数据处理提供了灵活手段。

汇编语言的缺点：

1）使用汇编语言，需要程序员了解计算机的硬件结构和指令系统，因此增加了编程难度。

2）根据计算机的CPU不同，其汇编指令也不相同，因此程序的通用性差。

3）使用汇编语言编写的程序与数学模型之间的对应关系不直观。

1.1.3 高级语言

高级语言是相对于汇编语言而言的，它是接近自然语言和数学公式的编程语言，基本脱离了机器的硬件系统，用人们更易理解的方式编写程序。高级语言与计算机的硬件结构及指令系统无关，它有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，而且容易学习，编写出的程序易于阅读、理解、修改，移植性和通用性较好，因而得到了迅速的普及和发展。

1957年，第一个完全脱离机器硬件的高级语言FORTRAN问世了，几十年来，共有几百种高级语言出现，其中影响较大、使用较普遍的有FORTRAN、COBOL、LISP、C、PROLOG、C++、VC、VB、Delphi、Java、C#等。

高级语言的出现使得计算机程序设计语言不再过度地依赖某种特定的机器或环境。这是因为高级语言在不同的平台上会被翻译成不同的机器语言程序，而不是直接被机器执行。计算机并不能直接地接受和执行用高级语言编写的源程序，源程序在输入计算机时，通过“翻译程序”翻译成目标程序，计算机才能识别和执行。这种“翻译”通常有两种方式，即编译方式和解释方式。

编译方式是指在源程序执行之前，就将程序源代码“翻译”成目标代码（机器语言），因此其目标程序可以脱离语言环境独立执行，使用比较方便，效率较高。但应用程序一旦需要修改，必须先修改源代码，再重新编译生成新的目标文件才能执行。

解释方式是指应用程序源代码一边由相应语言的解释器“翻译”成目标代码（机器语言），一边执行，因此程序的执行效率较低，而且不生成可独立执行的目标文件，应用程序不能脱离程序的解释系统。但这种方式比较灵活，可以动态地调整、修改应用程序。

计算机语言的发展经历了从机器语言到结构化程序设计语言，再到面向对象程序设计语言的过程。面向过程的语言致力于用计算机能够理解的逻辑来描述需要解决的问题及解决问题的具体方法、步骤。用这类语言编程时，程序不仅要说明做什么，还要详细地告诉计算机如何做，即程序需要详细描述解题的过程、步骤和细节，面向过程的语言种类繁多，如FORTRAN、BASIC、Pascal、Ada、C等。20世纪60年代中后期，软件越来越多，规模越

来越大，而软件的生产基本上是各自为战，缺乏科学规范的系统规划与测试、评估标准。这种落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致在软件开发与维护过程中出现一系列严重问题。

20世纪80年代初，在软件设计方法上产生了一次革命，其成果就是面向对象的程序设计语言的出现。面向对象程序设计（Object Oriented Programming，OOP）语言与以往各种编程语言的根本区别是其程序设计思维方法不同，面向对象程序设计可以更直接地描述客观世界存在的事物（即对象）及事物之间的相互关系。面向对象技术强调的基本原则是直接面对客观事物本身进行抽象并在此基础上进行软件开发，将人类的思维方式与表达方式直接应用在软件设计中。目前，典型的纯面向对象语言有Java和Smalltalk，典型的既面向过程又面向对象的混合型语言有C++。需要指出的是，计算机程序设计语言的高级和低级并不说明语言的优劣，它只表明这种语言离硬件的远近，同时也暗含了程序设计的难易和执行效率的高低。

高级语言具有如下特点。

（1）高级语言是完全独立于机器的通用语言

使用这种语言编写计算机程序，可以完全不考虑机器的结构特点，不必了解机器的指令系统，所编写的程序可以在各种机器上通用。

（2）高级语言是面向过程或面向对象的语言

用高级语言编写的程序与问题本身的数学模型之间有着良好的对应关系，因此程序的编写与阅读都很方便。

（3）计算机不能直接识别和执行高级语言源程序

使用高级语言编写的源程序必须经过“翻译程序”的处理，将之“翻译”成对应的机器语言程序，然后机器才能执行。

高级语言的缺点是“翻译”后生成的目标代码往往比较长，占用内存多，执行时间长，因此不适合需要实时处理的应用。

1.2 程序设计方法

随着大容量存储器的出现以及计算机应用范围的扩大，程序的编制越来越困难，程序的大小以算术级数递增，而程序的逻辑控制难度则以几何级数递增，这样人们不得不考虑程序设计方法。在程序设计的过程中，除了要求程序逻辑正确，能被计算机理解并执行外，还涉及程序的可读性、可靠性、可维护性以及程序的效率等方面的问题。在改善整个程序设计过程、提高程序质量等方面，程序设计方法都起着重要的作用。在软件的发展过程中，涌现出了很多程序设计方法，下面主要讨论两种程序设计方法。

1.2.1 结构化程序设计方法

1. 结构化程序设计的总体思想

结构化程序设计又称为面向过程的程序设计，它的总体思想是采用模块化结构，自顶而下，逐步求精。即首先把一个复杂的大问题分解为若干相对独立的小问题；如果小问题仍较复杂，则可以把这些小问题继续分解成若干子问题，这样不断地分解，使得小问题或子问题简单到能够直接用程序的3种基本控制结构表达为止；然后，对应每一个小问题或子问题编写出一个功能上相对独立的程序块，这种程序块被称为模块；每个模块解决一个子问题，最

后再将各个模块统一组装。这样，对一个复杂问题的解决就变成了对若干个简单问题的求解，这就是自顶而下、逐步求精的程序设计方法，它是结构化程序设计的基本原则。

2. 结构化程序设计的特征

结构化程序设计的特征有：

1）以3种基本控制结构的组合来描述程序。

2）整个程序采用模块化结构。

3）有限制地使用goto转移语句。

4）以控制结构为单位，每个控制结构只有一个入口和一个出口，各单位之间接口简单，逻辑清晰。

5）采用一定的书写格式使程序结构清晰、易于阅读。

3. 结构化程序设计的缺点

在结构化程序设计中，整个程序的功能是通过模块之间的相互调用而实现的。所以结构化程序设计在实际应用中存在一些缺点：

1）将数据和数据处理过程分离成为相互独立的实体，当数据结构发生变化时，所有相关的处理都要进行相应的修改，因此，程序代码的可重用性较差。

2）对于图形用户的应用开发起来比较困难，而图形界面越来越被人们广泛使用。

3）用户的要求难以在系统分析阶段准确定义，从而导致系统在交付使用时会产生许多问题。

1.2.2 面向对象程序设计方法

在程序设计方法的发展过程中，每一次重大突破都使得程序员可以应对更大的复杂性问题。在这条道路上迈出的每一步中，新的方法都运用和发展了以前的方法中最好的理念。今天，许多项目的规模又进一步发展，为了解决这个问题，面向对象程序设计方法应运而生。面向对象程序设计方法把数据看作程序开发中的基本元素，并且不允许它们在系统中自由流动。它将数据和操作这些数据的函数紧密地连接在一起，并保护数据不会被外界的函数意外地改变。面向对象程序设计在程序设计方法中是一个新的概念，对于不同的人可能意味着不同的内容。面向对象程序设计的定义是“面向对象程序设计是一种方法，这种方法为数据和函数提供共同的独立内存空间，这些数据和函数可以作为模板以便在需要时创建类似模块的副本”。从以上定义可以看到，一个对象被认为是计算机内存中的一个独立区间，在这个区间中保存着数据和能够访问数据的一组操作。因为内存区间是相互独立的，所以对象可以不经修改就应用于多个不同的程序中。

面向对象程序设计方法以对象为基础，它最主要的特点和成就是利用特定的软件工具直接完成从对象客体的描述到软件结构之间的转换。与结构化程序设计方法不同，面向对象程序设计方法是一种运用对象、类、继承、封装、消息传递、多态性等概念来构造系统的软件开发方法。它的应用解决了传统结构化开发方法中客观世界描述工具与软件结构的不一致性问题，缩短了开发周期，解决了从分析和设计到软件模块结构之间多次转换映射的繁杂过程。

1.2.3 两种程序设计方法的比较

面向对象程序设计方法克服了结构化程序设计方法中存在的问题。在面向对象程序设计

方法出现以前，结构化程序设计方法是程序设计的主流。在结构化程序设计方法中，问题被看作一系列需要完成的任务，函数（在此泛指例程、函数、过程）用于完成这些任务，解决问题的焦点集中于函数。其中函数是面向过程的，即它关注如何根据规定的条件完成指定的任务。在多函数程序中，许多重要的数据被放置在全局数据区，这样它们可以被所有的函数访问。每个函数都可以具有自己的局部数据。这种结构很容易造成全局数据在无意中被其他函数改动，因而程序的正确性不易保证。面向对象程序设计方法的出发点之一就是弥补结构化程序设计方法中的缺点，即对象是程序的基本元素，它将数据和操作紧密地联系在一起，并保护数据不会被外界的函数意外地改变。

表 1-1 在基本思想、代表语言等方面进一步比较了结构化程序设计方法与面向对象程序设计方法的异同。

表 1-1 结构化程序设计方法与面向对象程序设计方法的比较

方法 / 项目	结构化程序设计方法	面向对象程序设计方法
基本思想	自顶向下设计过程库，逐步求精，分而治之	自底向上设计类库
代表语言	BASIC、FORTRAN、C 等	Visual Basic、Java、C++ 等
处理问题的出发点	面向过程	面向问题
控制程序方式	通过设计调用或返回程序	通过“事件驱动”来激活和运行程序
可扩展性	功能变化会危及整个系统，扩展性差	只需修改或增加操作，而基本对象结构不变，扩展性好
重用性	不好	好
层次结构的逻辑关系	用模块的层次结构概括模块和模块之间的关系和功能	用类的层次结构来体现类之间的继承和发展
运行效率	相对高	相对低

1.3 C++ 语言的发展及特点

1.3.1 C++ 语言的发展

程序设计语言的发展是一个逐步递进的过程，究其根源，C++ 语言是从 C 语言发展过来的，而 C 语言的历史可以追溯到 1969 年。

1969 年，美国贝尔实验室的 Ken Thompson 设计了一个操作系统软件 UNIX。接着，他又根据剑桥大学的 Martin Richards 设计的 BCPL（Basic Combined Programming Language）为 UNIX 设计了一种便于编写系统软件的语言，命名为 B。1972 年，贝尔实验室的 Dennis Ritchie 在 B 语言的基础上设计并实现了 C 语言。C 语言既保持了 BCPL 和 B 语言的优点（精炼、接近硬件），又克服了它的缺点（过于简单、无数据类型），得到了广泛的应用。随着 C 语言的广泛使用，它的不同实现版本之间出现了或多或少的差异。为了解决这个问题，1983 年，美国国家标准学会（American National Standards Institute，ANSI）制定了新的 C 语言标准，称之为 ANSI C。C 语言兼有汇编语言和高级语言的优点，既适合开发系统软件，也适合编写应用程序，被广泛应用于事务处理、科学计算、工业控制、数据库技术等领域。但是随着软件规模的增大，用 C 语言编写程序渐渐显得有些吃力了。C 语言是结构化和模块化的

语言，它是面向过程的。在处理较小规模的程序时，程序员用C语言还比较得心应手。但是当问题比较复杂、程序的规模比较大时，结构化程序设计方法就显出它的不足。C程序的设计者必须细致地设计程序中的每一个细节，准确地考虑程序运行时每一时刻发生的事情，例如，各个变量的值是如何变化的，什么时候应该进行哪些输入，在屏幕上应该输出什么等。这对程序员的要求是比较高的，如果面对的是一个复杂问题，程序员往往感到力不从心。当初提出结构化程序设计方法的目的是解决软件设计危机，但是这个目标并未完全实现。

为了解决软件设计危机，面向对象程序设计方法在20世纪80年代被提出，需要设计出能支持面向对象程序设计方法的语言。在实践中，人们发现C语言是如此深入人心，使用如此广泛，面对程序设计方法的革命，最好的办法不是另外发明一种新的语言去代替它，而是在它原有的基础上加以发展。在这种形势下，C++应运而生。C++是由贝尔实验室的Bjarne Stroustrup博士及其同事于20世纪80年代在C语言的基础上进行改进和扩充的，其将“类”的概念引入了C语言，构成了最早的C++语言。后来，Stroustrup和他的同事们又为C++语言引进了运算符重载、引用、虚函数等许多特性，并使之更加精炼，于1989年推出了AT&T C++ 2.0版。随后美国国家标准学会和国际标准化组织（International Organization for Standardization，ISO）一起进行了标准化工作，并于1998年正式发布了C++语言的国际标准ISO/IEC：98—14882。

1.3.2 C++语言的特点

C++语言是广泛使用的程序设计语言之一，因其特有的优势在计算机应用领域占有重要的地位。下面介绍C++语言的主要特点。

1）全面兼容C语言。C++语言继承了C语言简明、高效、灵活等众多优点；绝大多数C语言程序不经修改可以直接在C++语言环境中运行；C语言程序员只需要学习C++语言扩充的新特性，就可以很快地使用C++语言编写程序。

2）C++语言是C语言的超集。它既保持了C语言的简洁、高效和接近汇编语言等特点，又克服了C语言的缺点，其编译系统能检查更多的语法错误，因此，C++语言比C语言更安全。

3）支持面向对象的程序设计特征。对C语言的兼容性使得C++语言既支持面向过程的程序设计，又支持面向对象的程序设计。C++语言支持抽象性、封装性、继承性和多态性等。程序各个模块的独立性更强，程序的可读性和可理解性更好，程序代码的结构性更加合理。这对于设计和调试一些大的软件尤为重要。

4）具有程序效率高、灵活性强的特点。C++语言使程序结构清晰、易于扩展、易于维护而不失效率。

5）具有通用性和可移植性。C++语言是一种标准化的、与硬件基本无关的程序设计语言，C++语言程序通常无须修改或稍加修改便可在其他计算机上运行。

6）具有丰富的数据类型和运算符，并提供了强大的库函数。

1.4 C++语言程序

1.4.1 C++语言程序举例

首先来看几个简单的C++语言程序。

【例1-1】 在屏幕上输出一行字符“Hello! Welcome to C++!”。

```
#include <iostream>                          //包含头文件iostream
using namespace std;                         //使用命名空间std
int main( )                                  //函数首部
    {                                        //函数体开始
        cout<<"Hello!Welcome to C++! \n";    //输出Hello! Welcome to C++!
        return 0;                            //程序正常结束时向操作系统返回0值
    }                                        //函数体结束
```

程序运行结果如图 1-1 所示。

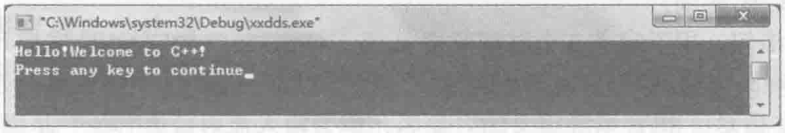

图 1-1 例 1-1 的运行结果

说明：

① #include<iostream>

这是 C++ 语言的编译预处理命令，用“#”开始的行称为编译预处理命令，“#include <iostream >”的作用是将头文件 iostream 的内容包含到该命令所在的程序文件中，代替该命令行。头文件 iostream 的作用是向程序提供输入或输出所需要的一些信息。

② int main()

每个程序都必须有且只有一个名为 main 的函数，称为主函数，该函数是程序执行的入口，main() 函数前 int 的作用是声明该函数的返回值类型是整型。函数体用“{}”括起来。

③ cout<<"Hello!Welcome to C++! \n";

cout 实际上是 C++ 语言系统定义的对象名，称为输出流对象。“<<”是插入运算符，对象 cout 结合运算符“<<”的作用是将其后的内容依据一定的规则显示到标准的输出设备（通常为显示器）中，本例中的参数为字符串 "Hello! Welcome to C++! \n"，它控制显示内容。在字符串尾部的“\n”代表一个称为换行符的单字符。需要注意的是，本行用一个“;”结束，分号是 C++ 语言的语句结束标志。

④ //

为了提高程序的可读性，C++ 语言提供了两种注释方法：一种是单行注释“//”，其后的内容（直到换行）为注释内容，注释只占用一行；另一种是多行注释“/*”和“*/”，它们之间的内容为注释内容，注释可以占用一行或跨越几行。注释仅供阅读程序使用，是程序的可选部分。

【例 1-2】 求 a 和 b 之和，a 和 b 的值由键盘输入。

```
#include <iostream>                  //编译预处理命令
using namespace std;
int main( )
{
    int a,b,sum;                     //定义变量
    cout<<"请输入两个数a和b: ";      //用来提示输入何种信息的输出语句
    cin>>a>>b;                       //输入语句
    sum=a+b;                         //赋值语句
    cout<<"a+b="<<sum<<endl;         //输出语句
    return 0;
}
```