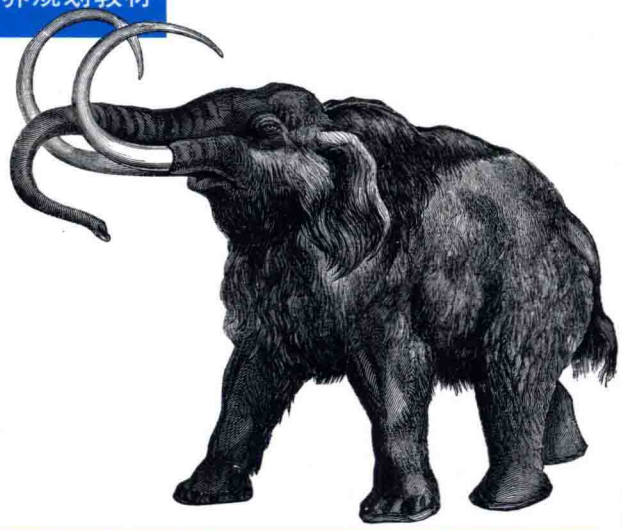
普通高等教育“十一五”国家级规划教材
工业和信息化“十三五”人才培养规划教材



C++ 程序设计 第3版

C++ Programming Language

齐建玲 邓振杰 ◎ 主编
斯庆巴拉 侯晓芳 ◎ 副主编

将难以理解的**概念及应用增加图示**，帮助读者理解
对示例程序的相应行处**标注提示**，方便读者阅读
将两个大规模实训案例**拆分到各章形成独立的小实例**，贯穿所有知识点

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十一五”国家级规划教材
工业和信息化“十三五”人才培养规划教材

C++程序设计 第3版

C++ Programming Language

齐建玲 邓振杰 ◎ 主编
斯庆巴拉 侯晓芳 ◎ 副主编

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

C++程序设计 / 齐建玲，邓振杰主编. -- 3版. -- 北京 : 人民邮电出版社，2017.2
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-42378-8

Ⅰ. ①C… Ⅱ. ①齐… ②邓… 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(2016)第103540号

内 容 提 要

本书以介绍 C++语言的基本知识为主，旨在帮助读者建立面向对象程序设计的编程思想，主要内容包括 C++与面向对象程序设计概述、C++程序设计基础、数组、函数、指针和引用、结构体和共用体、类与对象、静态与友元、继承与派生、运算符重载、虚函数和多态性、C++输入/输出流、模板和异常处理等。

本书概念清楚、通俗易懂、实例丰富，注重基础知识与典型应用相结合，具有较高的系统性、实用性和可操作性。书中所有程序代码均在 Visual Studio 2013 环境下运行通过。

本书为普通高等学校、高等职业院校计算机类各专业学习“C++面向对象程序设计”课程的教材，也可作为其他专业的程序设计入门教材和广大计算机应用人员的自学参考书。

◆ 主　　编　齐建玲　邓振杰
副 主 编　斯庆巴拉　侯晓芳
责任编辑　马小霞
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：19.25　　2017 年 2 月第 3 版
字数：483 千字　　2017 年 2 月北京第 1 次印刷

定价：49.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广字第 8052 号

第3版前言

C++语言是目前应用最广泛的面向对象程序设计语言，《魔兽世界》等70%以上的网络游戏都是基于C++语言开发的，掌握C++语言已经成为游戏开发的基本要求；操作系统、搜索引擎、移动互联网应用等我们所用的大多数软件也都是用C++语言编写的，在涉及大规模、高性能计算时，C++语言的运算速度和稳定性优势非常明显，非常适合现在流行的移动互联网应用程序开发。

《C++程序设计（第2版）》自2008年4月发行以来，受到广大读者的欢迎，特别是得到普通高等学校、高等职业院校师生的一致好评。编者结合C++语言的最新进展和当前业界的最佳实践，结合近几年课程教学经验的总结、教学改革实践的成效以及广大读者的反馈意见，在保留第2版特色的基础上，对教材进行了全面的修订。这次修订的主要工作如下。

- 对教材内容结构进行了重新调整和编排，使整体结构更加清晰，内容编排更加合理。对个别章节的内容进行了删减，教学内容更有条理，重点内容更突出，非重点内容相对弱化，将弱化的知识点并入合适的章节，在应用时进行简要说明。
- 考虑到部分读者可能已经学过C语言，所以本次修订把面向过程程序设计的内容进行了一定程度的简化，重点内容放在面向对象程序设计部分。在难以理解的概念及应用上，增加了一些图示，帮助读者理解书中内容。
- 对一些示例程序进行了替换，选用更实用、更新颖、更有趣味性的例子，在说明问题的同时提高读者的学习兴趣，对原教材的典型示例进行了保留。书中用文字说明示例程序时，在对应行处增加了提示，以方便读者阅读，提高可读性。
- 为培养学生的程序设计能力，加强实践教学，设计了两个具有实际应用背景、实用性强、程序规模较大的实训案例，涵盖了所有章节的内容。编者将这两个案例分别拆分到每一章，形成相对独立的应用实例，并作为案例实战内容安排到每章的最后，读者可以边学边练，在完成每章案例的基础上，完成最后两个完整的实训案例，这样更有利于读者明确学习目标，加强程序设计能力培养。
- 集成开发环境更新为Visual Studio 2013，所有例题程序均在Visual Studio 2013集成环境中运行通过。

本书以介绍C++语言的基本知识为主，旨在帮助读者建立面向对象程序设计的编程思想，对C++的技术讲解全面，语言表述准确，简明扼要，具有循序渐进、深入浅出的特点。主要内容包括C++与面向对象程序设计概述、C++程序设计基础、数组、函数、指针和引用、结构体和共用体、类与对象、静态与友元、继承与派生、运算符重载、虚函数和多态性、C++输入/输出流、模板和异常处理等。

本书适合作为初学C++语言的入门级教材，对每个知识点都精心设计了示例程序，全书以实训案例为主线驱动C++语言的学习，将难以理解的问题简单化，轻语法，重实践，示例恰当，重点突出，使读者树立良好的编程思维，激发读者的学习兴趣。本书的参考学时为64学时（其中实践学时为20~24学时），教师可根据实际情况进行适当调整。

本书提供丰富的配套教学资源，包括多媒体课件、教学大纲、习题答案、程序源代码、

模拟试卷、在线教学网站等。由于教材篇幅限制，实训案例的完整代码就不放在教材中，读者可登录 www.ryjiaoyu.com 下载。

本书由齐建玲、邓振杰主编，斯庆巴拉、侯晓芳任副主编，各章编写分工如下：第 1 章由邓振杰编写，第 2、5 章由齐建玲编写，第 3、6、9、11、12 章由侯晓芳编写，第 4、7、8、10、13 章由斯庆巴拉编写，李新荣、崔岩、王静、李瑛、孙红艳、张春娥、朱蓬华、李杰、张艳和王健也参加了大纲讨论和文稿整理工作。全书由齐建玲、邓振杰负责统稿。

由于编者水平有限，书中难免存在一些错误和不妥之处，敬请读者批评指正。

编者

2016 年 10 月

目 录 CONTENTS

第4章 函数 64

第5章 指针和引用 82

第6章 结构体和共用体 106

第7章 类与对象 120

第 12 章 C++输入/输出流 269

第 13 章 模板和异常处理 288

PART 1

第1章 C++与面向对象程序设计概述

20世纪90年代以来，面向对象程序设计（Object Oriented Programming，OOP）成为计算机程序设计的主流，其设计思想已经被越来越多的软件设计人员所接受。C++是在C语言的基础上发展起来的，它完全兼容C语言，不仅继承了C语言灵活高效、功能强大、可移植性好等优点，而且引入了面向对象程序设计的思想和机制，可以在很大程度上提高编程能力，减少软件维护的开销，增强软件的可扩展性和可重用性。因此，C++语言一出现就受到了广大用户的青睐，其版本不断更新，功能不断完善，迅速成为面向对象程序设计的首选语言。

1.1 面向对象程序设计

1.1.1 基本概念

面向对象程序设计的主要特征是“程序=对象+消息”，其基本元素是类和对象。面向对象程序设计在结构上具有以下特点。

（1）程序一般由类的定义和类的使用两部分组成，在主程序中定义各对象并规定它们之间传递消息的规律。

（2）程序中的一切操作都是通过向对象发送消息来实现的，对象接收到消息后，启动相应的方法完成操作。

一个程序中所使用的类可以由用户自己定义，也可以使用现成的类（包括系统类库中为用户提供的类和其他用户自己设计构建好的类）。尽量使用现成的类是面向对象程序设计所倡导的风格。

1．对象

现实世界中客观存在的任何事物都可以称为对象，它可以是有形的具体事物，如一本书、一张桌子、一辆汽车等，也可以是一个无形的抽象事物，如一次演出、一场球赛等。对象是构成现实世界的一个独立单位，可以很简单，也可以很复杂，复杂的对象可以由简单的对象构成。

现实世界中的对象既具有静态的属性（或称为状态），又具有动态的行为（或称为操作）。例如，每个人都有姓名、性别、年龄、身高、体重等属性，都有吃饭、走路、睡觉、学习等

行为。所以，现实世界中的对象一般可以表示为“属性+行为”。

在面向对象程序设计中，对象是由对象名、若干属性和一组操作封装在一起构成的实体。其中属性数据是对象固有特征的描述，操作是对这些属性数据施加的动态行为，是一系列的实现步骤，通常称为方法。

对象通过封装实现了信息隐蔽，其目的就是防止外部的非法访问。对象与外部世界是通过操作接口联系的，在外边看不见操作的具体实现步骤，操作接口提供了这个对象所具有的功能。

打个形象的比喻，一个对象就好比一台 MP3，我们使用 MP3 时通过播放、暂停、录音等按钮就可以操作 MP3，通过这些按钮就可以与 MP3 实现交互，我们没有必要了解这些交互操作是如何具体实现的，因为它们被封装在机器内部，其内部电路对我们来说是隐蔽的，是不可见的，也是无法修改的，我们只能借助这些按钮实现对 MP3 的操作。

如果把 MP3 看作对象，则 MP3 的颜色、存储容量、产品尺寸等参数就相当于对象的属性，播放、暂停、录音等动作就相当于对象的操作，而它的按钮就相当于对象的接口。当在程序设计中使用对象时，就像通过按钮操作 MP3 一样，通过对象与外部的接口来操作，这样不仅使对象的操作变得简单、方便，而且具有很高的安全性和可靠性。

2．类

在面向对象程序设计中，类是具有相同属性数据和操作的对象的集合，它是对一类对象的抽象描述。例如，小张、老李等是一个个具体的人，虽然他们每个人的年龄、身高、体重等各不相同，但他们的基本特征都是相同的，都具有相似的生理构造和动作行为，故可以把他们统称为“人”类，而小张、老李等每一个人只是“人”类的一个个具体实例——对象。

类是创建对象的模板，类包含着所创建对象的状态描述和定义的方法，一般是先声明类，再由类创建其对象。按照这个模板创建的每个具体实例就是对象。

类和对象之间是抽象与具体的关系，类是对很多对象进行抽象的结果，一个对象是类的一个实例。例如，“汽车”是一个类，它是由成千上万个具体的汽车抽象而来的一般概念，而“帕萨特”就可以看作是“汽车”类的一个实例对象。

3．属性

对象中的数据称为对象的属性，而类中的特性称为类的属性，例如眼睛、鼻子、嘴巴等是“人”类共有的属性，而张三的身高、年龄、性别、职业等是“张三”这个对象的属性。不同的类和对象具有不同的属性。

4．消息

现实世界中的对象不是一个个孤立的实体，它们之间存在着各种各样的联系。同样，在面向对象程序设计中，对象之间也需要联系，称为对象的交互。

在面向对象程序设计中，当要求一个对象做某一操作时，就向该对象发出请求，通常称为“消息”。当对象接收到消息时，就调用有关方法，执行相应操作。这种对象与对象之间通过消息进行相互联系的机制就叫作消息传递机制，通过消息传递可实现对象的交互。

5．方法

方法就是对象所能执行的操作。方法包括接口和方法体两部分。方法的接口就是消息的模式，它告诉用户如何调用该方法；方法体则是实现操作的一系列步骤，也就是一段程序代码，这些代码封装在对象内部，用户在外部无法看到。

消息和方法的关系是：对象接收到其他对象的消息，就调用相应的方法；反之，有了合适的方法，对象才能响应相应的消息。所以，消息模式和方法接口应该是一致的，只要方法接口保持不变，方法体的改变就不会影响方法的调用。

1.1.2 传统程序设计及其局限性

计算机程序设计语言大致经过了机器语言、汇编语言和高级语言3个阶段。

20世纪50年代初的程序都是用机器语言和汇编语言编写的，程序设计相当麻烦，严重影响了计算机的普及应用。

50年代中期出现了高级程序设计语言Fortran，它在计算机程序设计语言发展史上具有划时代的意义。该语言引进了许多重要的程序设计概念，如变量、数组、循环、条件分支等，但是该语言在使用中也发现了一些不足，例如不同部分的相同变量名容易发生混淆等。

50年代后期，高级语言Algol 60的设计者决定在程序段内部对变量实行隔离，提出了块结构的思想，由“Begin...End”来实现块结构，对数据实行了保护。

1969年，E.W.Dijkstra首先提出了“结构化程序设计”的概念，他强调从程序结构和风格上研究程序设计。结构化程序设计的程序代码是按顺序执行的，有一套完整的控制结构，函数之间的参数按一定规则传递，不提倡使用全局变量，程序设计的首要问题是“设计过程”。因此，结构化程序设计仍是面向过程的程序设计。

到20世纪70年代末，结构化程序设计方法有了很大的发展，但由于程序规模越来越大，数据与处理数据的方法之间的分离往往使程序变得难以理解，不适于大规模程序开发，故出现了“模块化程序设计”。

模块化程序设计将软件划分成若干个可单独命名和编址的部分，称为“模块”。模块化程序设计思路是“自顶向下，逐步求精”，其程序结构是按功能划分成若干个基本模块，各模块之间的关系尽可能简单，在功能上相对独立。模块和模块之间隔离，外界不能访问模块内部信息，即这些信息对模块外部是不透明的，只能通过严格定义的接口对模块进行访问。

模块化程序设计将数据结构和相应算法集中在一个模块中，提出了“数据结构+算法=程序设计”的程序设计思想。模块化能够有效地管理和维护软件研发，能够有效地分解和处理复杂问题。但它仍是一种面向过程的程序设计方法，程序员必须时刻考虑所要处理数据的格式，对不同格式的数据做相同处理或对相同格式数据做不同处理都要重新编程，代码可重用性不好。

综上所述，伴随计算机技术的大规模推广，人们对计算机软件的功能和性能要求越来越高，使得传统的程序设计已不能满足日益增长的需要，表现出以下几方面的局限性。

1．软件开发效率低下

传统程序设计语言尽管经历了从低级语言到高级语言的发展，但还属于过程性语言。程序设计还是从语句一级开始，软件生产缺乏可重用的构件，软件的重用问题没有得到很好的解决，导致软件生产的工程化和自动化屡屡受阻。

复杂性也是影响软件生产效率的重要因素。随着计算机应用范围越来越广，软件规模越来越大，要解决的问题也越来越复杂。传统程序设计将数据与数据的操作分离，并且对同一数据的操作往往分散在程序的不同地方。这样，如果一个或多个数据的结构发生变化，这种变化将波及程序的许多地方，甚至遍及整个程序，致使许多函数和过程需要重写，严重时会导致整个程序崩溃。因此，程序的复杂性对传统程序设计是一个很棘手的问题，也是传统程序设计难于有根本性突破的重要原因。

软件维护是软件生命周期中最后一个环节。传统程序设计中数据与数据操作分离的结构，使得维护数据和处理数据的操作过程要花费大量的时间和精力，严重地影响了软件的生产效率。

总之，要提高软件的生产效率，就必须很好地解决软件的重用性、复杂性和可维护性问题，但这是传统程序设计自身无法解决的。

2．难以应付庞大的信息量和多样的信息类型

当前，计算机要处理的数据已从简单的数字和字符发展为具有多种格式的多媒体数据，如文本、音频、视频、图形、图像和动画等，描述的问题从单纯的计算问题发展到复杂的自然现象和社会现象问题。于是，计算机要处理的信息量和信息类型迅速增加，这就要求程序设计语言具有更强大的信息处理能力，而这是传统程序设计无法办到的。

3．难以适应各种新环境

当前，在计算机应用技术中，并行处理、分布式处理、网络和多机系统应用等已经成为程序设计的主流，这就要求系统具有一些能独立处理数据的节点，节点之间有通信机制，即能通过消息传递进行联络，而这也是传统程序设计无能为力的。

显然，传统程序设计已不能满足计算机技术迅猛发展的需要，大规模软件开发迫切需要一种全新的程序设计方法来满足需要。

1.1.3 面向对象程序设计的特点

面向对象程序设计是在结构化程序设计基础上发展起来的一种全新的程序设计方法，其本质是把数据和处理数据的过程抽象成一个具有特定属性和行为的实体——“对象”。面向对象程序设计最突出的特点就是封装性、继承性和多态性。

1．封装性

封装是一种数据隐蔽技术，在面向对象程序设计中可以把数据和与数据有关的操作集中在一起形成类，将类的一部分属性和操作隐蔽起来，不让用户访问，另一部分作为类的外部接口，可以让用户访问。类通过接口与外部发生联系，用户只能通过类的外部接口使用类提供的服务，而内部的具体实现细节则被隐蔽起来，对外是不可见的。

封装性可以描述如下。

（1）一个清楚的边界。对象的所有属性和操作被限定在该边界内部。

（2）一个外部接口。该接口用以描述对象和其他对象之间的相互作用，即给出在编写程序时用户可以直接使用的属性和操作。

（3）隐蔽受保护的属性和内部操作。类所提供的功能的实现细节以及仅供内部使用和修改的属性，不能定义这个对象的类的外部访问。

C++语言通过类来支持封装性。在C++语言中，类是数据及其相关操作的封装体，可以作为一个整体来使用。类中的具体操作细节被封装起来，用户在使用一个已定义类的对象时，无需了解类内部的实际工作流程，只要知道如何通过其外部接口使用它即可。封装的目的就是防止非法访问，可以对属性和操作的访问权限进行合理控制，减少程序之间的相互影响，降低出错的可能性。

2．继承性

在面向对象程序设计中，继承是指新建的类从已有的类那里获得已有的属性和操作。已有的类称为基类或父类，继承基类而产生的新建类称为基类的子类或派生类。由父类产生子

类的过程称为类的派生。

C++语言支持单继承和多继承。通过继承，程序可以在现有类的基础上声明新类，即新类是从原有类的基础上派生出来的，新类将共享原有类的属性，并且还可以添加新的属性。继承有效地实现了软件代码的重用，增强了系统的可扩充性。

3．多态性

在人们的日常生活中，常常把“下象棋”“下跳棋”“下围棋”“下军棋”等统称为“下棋”，也就是用“下棋”这同一个名称来代替“下象棋”“下跳棋”“下围棋”“下军棋”这些类似的活动。

在面向对象程序设计中，多态性是指相同的函数名可以有多个不同的函数体，即一个函数名可以对应多个不同的实现部分。在调用同名函数时，由于环境的不同，可能引发不同的行为，导致不同的动作，这种功能称为多态。它使得类中具有相似功能的不同函数可以使用同一个函数名。

多态既表达了人类的思维方式，又减少了程序中标识符的个数，方便了程序员编写程序。多态是面向对象程序设计的重要机制。

1.1.4 面向对象程序设计语言

伴随面向对象程序设计方法的提出，出现了不少面向对象的程序设计语言，如 Object-C、Java、C++等。这些语言大致可分为以下两类。

（1）重新开发全新的面向对象程序设计语言。其中最具有代表性的语言是 Java、Smalltalk 和 Eiffel。Java 语言适合网络应用编程；Smalltalk 语言则完整体现了面向对象程序设计的概念；Eiffel 语言除了具有封装和继承外，还集成了其他面向对象的特征，是一种很好的面向对象程序设计语言。

（2）对传统程序设计语言进行扩展，加入面向对象程序设计特征形成的一种语言。这类语言又称为“混合型语言”，一般是在其他语言的基础上加入面向对象程序设计的特征开发出来的，典型代表是 C++程序设计语言。

C++程序设计语言是一种高效实用的混合型面向对象程序设计语言，包括两部分：一部分是 C++语言的基础部分，以 C 语言为核心，包括了 C 语言的主要内容；另一部分是 C++语言的面向对象部分，是 C++语言对 C 语言的扩充，加入了面向对象程序设计的概念和特征。这使得 C++语言既支持传统的面向过程程序设计，又支持全新的面向对象程序设计，同时具有 C 语言丰富的应用基础和开发环境的支持。对于已经较好地掌握了 C 语言的用户而言，学习 C++语言要相对容易一些，这些都是 C++语言成为当前最为流行的面向对象程序设计语言的主要原因。

1.2 C++语言的发展和特点

1.2.1 C++语言的发展

各种程序设计语言对程序设计方法的支持是不同的，C++语言完全支持面向对象程序设计。C++语言的出现主要归功于 C 语言，C 语言在 C++语言中作为子集保留下来。

1980 年，美国 AT&T 公司贝尔实验室的 Bjarne Stioustrup 博士为了仿真课题研究，编写了称为“带类的 C”语言版本。1983 年 7 月用 C++将该语言名字定下来，并对外公开发表。

C++语言的名字强调了从C语言而来的演化特性，“++”是C语言的增量运算符，表示是C语言的扩充。C++语言继承了C语言的优点，又极大地扩充了C语言的功能，是在C语言的基础上增加了面向对象程序设计的特征。

C++语言已经在众多应用领域迅速成为程序员首选的程序设计语言，尤其适用于开发大、中型项目，从软件开发时间、费用到软件的可重用性、可扩充性、可维护性以及可靠性等方面都显示出C++语言的优越性。

与之相适应，C++语言的开发环境也随之不断推出。目前，常用的C++语言集成开发环境主要有Visual C++（简称VC）、Microsoft Visual Studio（简称VS）等。

Visual C++是美国微软（Microsoft）公司推出的32位Windows系统下的面向对象可视化集成开发环境，简称MSVC、VC++或VC。美国微软公司于20世纪80年代中期在Microsoft C 6.0的基础上开发了Microsoft C/C++ 7.0，同时引进了Microsoft Foundation Class（简称MFC）1.0版本，完善了源代码。直到微软公司推出的Microsoft C/C++ 8.0，即Visual C++ 1.0版本出现，它是第一个真正基于Windows系统下的可视化集成开发环境，将编辑、编译、连接和执行集成为一体。在Visual C++ 2.0版本以后，微软公司没有推出3.0版本，版本号直接从2.0跳到了4.0，将Visual C++与MFC的版本号取得一致。

伴随Windows 98操作系统的发布，微软公司隆重推出了Visual C++ 6.0，它提供了许多新特点，功能更加强大，是一种广泛应用的C++语言集成开发环境。Visual C++ 6.0允许编辑器自动完成通用语句编辑，使用Developer Studio不仅可以创建被Visual C++ 6.0使用的源文件和其他文档，还可以创建、查看和编辑与任何ActiveX控件有关的文档。Visual C++ 6.0包括了一些新增加的MFC库功能，增加的内容包括用于Internet编程的库和新的通用控件。Visual C++ 6.0还提供了快捷的数据库访问方式，允许用户建立强有力的数据库应用程序，可以使用开放式数据库互连（Open Database Connectivity,ODBC）来访问各种数据库管理系统，也可以使用数据访问对象（Data Access Objects,DAO）访问和操作数据库中的数据并管理数据库。

Microsoft Visual Studio是微软公司近些年推出的开发工具包系列产品，是一个完整的开发工具集，它包括了整个软件生命周期中所需要的大部分工具，如UML工具、代码管控工具、集成开发环境（Integrated Development Enviroment,IDE）等。所写的目标代码适用于Microsoft支持的所有平台，包括Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework、Microsoft Silverlight和Windows Phone。

Microsoft Visual Studio是目前最流行的Windows平台应用程序的集成开发环境，Visual Studio中就包含了Visual C++。考虑到Visual Studio的流行性，本书将以Visual Studio 2013作为平台介绍C++程序设计的基础知识，它具有以下特点。

（1）功能强大，先进的代码编辑器和无缝调试使得编写代码比以往更加快速和流畅。

（2）良好的通用性，可使用集成的工具创建Windows、Android和iOS设备的应用程序。

（3）良好的扩展性，可以在Visual Studio中安装或者创建用户的扩展插件，所有Visual Studio的插件都可以免费使用。

（4）提高了开发人员的工作效率，改进了用户界面，内置了许多提高工作效率的功能，如自动补全方括号，团队资源管理器增强了主页设计功能。

总之，C++语言经历了若干版本的不断发展，软件系统逐渐庞大，功能也日臻完善。

1.2.2 C++语言的特点

C++语言具有如下特点。

（1）C++语言全面兼容 C 语言，许多 C 语言代码不经修改就可以在 C++语言中使用。

（2）用 C++语言编写的程序可读性更好，代码结构更为合理。

（3）生成代码质量高，运行效率仅比汇编语言低 10%~20%。

（4）从开发时间、费用到形成软件的可重用性、可扩充性、可维护性和可靠性等方面有很大提高，使得大、中型软件开发变得容易很多。

（5）支持面向对象程序设计，可方便地构造出模拟现实问题的实体和操作。

1.3 C++语言程序基本结构

1.3.1 C++语言程序基本结构

现在通过一个小程序来说明 C++程序的基本结构，该程序在屏幕上输出字符串"Hello！"和"This is my first C++ program！"。

【例 1.1】一个简单的 C++程序。

```
#include<iostream>
using namespace std;
void sayhello();
int main()
{
    sayhello();
    // 在显示器上输出显示一行字符串
    cout<<"This is my first C++ program! "<<endl;
    return 0;
}
//函数定义
void sayhello()
{
    cout<<"Hello! "<<endl;
}
```

1．头文件

在 C++程序开始部分出现以#开头的命令，表示这些命令是预处理命令，称为预处理器。C++提供了 3 类预处理命令：宏定义命令、文件包含命令和条件编译命令。例 1.1 中出现的“#include<iostream>”命令是文件包含命令，指示编译器将头文件 iostream 中的代码嵌入程序中该命令所在之处。其中 include 是关键字，尖括号内是被包含的文件名，iostream 是一个头文件，该文件包含程序输入/输出操作所必需的标准输入/输出流对象。

C++语言包含头文件的格式有以下两种。

（1）#include<文件名>

编译器并不是在用户编写程序的当前目录查找文件，而是在 C++系统目录中查找。这种包含方法常用于标准头文件，如 iostream、string 等。

（2）#include"文件名"

编译器首先在用户编写程序的当前目录中查找文件，然后再在 C++系统目录中查找。

对于一个存在着标准输入/输出的 C++控制台程序，一般会在#include <iostream>的下面

有一行语句“using namespace std;”，这条语句就是告诉编译器，这行代码之后用到的 cout、cin 等标准输出/输入流对象都是在 std 这个命名空间内定义的，其实就是表示所有的标准库函数都在标准命名空间 std 中进行了定义，其作用就在于避免发生重命名的问题。C++语言引入命名空间 namespace，就是为了解决多个程序员在编写同一个项目时，可能出现的函数重名问题，解决方法就是在函数之前加上自己的命名空间。

2．函数

C++的程序是由若干个文件组成的，每个文件又由若干个函数组成。函数之间是相互独立的，相互之间可以调用。但函数在调用之前，必须先定义。

函数要先声明后调用，一般用函数原型进行声明。例 1.1 中第 3 行 void sayhello();就是函数原型声明，其作用是告诉编译器，这个函数可以使用，该函数在其他地方已经定义了。

C++程序中的函数可分为两大类，一类是用户自己定义的函数，另一类是系统提供的标准函数。使用系统提供的标准函数时，可以直接调用，但需要将该函数的头文件包含在程序中。

例 1.1 中第 12～15 行是 sayhello()函数的定义。一个 C++函数中的任何语句都被括在一对花括号“{}”中，这部分内容称作函数体，而函数名 sayhello 和它后面的小括号“()”称为函数头。

3．主函数

在组成 C++程序的若干个函数中，必须有且仅有一个主函数 main()。执行程序时，系统先从主函数开始运行，其他函数只能被主函数调用，或通过主函数被其他函数所调用，函数还可以嵌套调用。

例 1.1 中第 4 行定义了一个主函数 main()。它是程序开始执行的地方，即在程序生成可执行文件后，将从此处开始运行程序。主函数可以带参数，也可以不带参数。由{}括起来的内容是主函数 main()的函数体，其中左大括号“{”表示函数的开始，右大括号“}”表示函数的结束。函数体部分由许多 C++语句组成，这些语句描述了函数实现的具体功能。

4．注释

程序中的注释只是为了阅读方便，并不增加执行代码的长度，在编译时注释被当作空白行跳过。C++语言中的程序注释有以下两种书写格式。

第一种注释方法是以“//”表示注释开始，本行中“//”后面的字符都会被作为注释处理，这种注释方式多用于较短的程序注释。

第二种注释方法是以“/*”开始，以“*/”结束，二者之间的所有字符都会被作为注释处理，此时的注释可以是一行，也可以是多行，适合于大块的注释。

例 1.1 中的第 7、11 行都是程序的注释部分，其中第 7 行用来说明下面的语句功能是在显示器上输出一行字符串。注释对于较复杂的 C++程序是非常必要的，可以解释一行语句或几行语句的作用或功能，提高程序的可读性。

5．输入/输出

输入/输出语句是 C++最基本的语句。例如例 1.1 中的语句“cout<<"Hello!"<<endl;”和“cout<<"This is my first C++ program! "<<endl;”都是输出语句，例 1.1 中没有输入语句。

这里的 cout 是标准输出流对象，实际指定显示器为输出设备；“<<”是 cout 中的运算符，表示把它后面的数据在输出设备上输出显示，双引号表示要显示的内容是一个字符串，最后的 endl 表示回车换行，分号“;”表示语句结束。C++规定语句必须要用分号“;”