

教育部高等学校文科大学计算机课程教学指导分委员会立项教材
高等院校计算机基础教育应用型系列规划教材

C++ 程序设计

C++ CHENGXU SHEJI

温斯琴◎主 编
于 鹰 西佳平◎副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

教育部高等学校文科大学计算机课程教学指导分委员会立项教材
高等院校计算机基础教育应用型系列规划教材

C++ 程序设计

主 编 温斯琴

副主编 于 鹰 西佳平

参 编 王平泉

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

C++ 语言是现今广泛使用的程序设计语言,深受广大程序员的青睐。本书从简单的样例入手,全面系统地讲述了 C++ 语言的基础知识、基本语法和编程方法,重点讲述解决问题的思路,培养学习者计算思维能力。

全书分为两大部分共 13 章,第一部分(前 7 章)是基础知识部分,包括 C++ 语言概述,数据类型、运算符与表达式,程序流程控制结构,数组,指针与引用,函数,自定义数据类型;第二部分(后 6 章)是面向对象程序设计部分,包括类与对象,静态数据成员与友元,运算符重载,继承与派生,虚函数与多态性,文件流与文件操作。全书文字通俗易懂,内容深入浅出,概念系统全面,突出重点,侧重应用。

本书适合作为高等院校计算机相关专业程序设计的初学者。

图书在版编目(CIP)数据

C++程序设计 / 温斯琴主编. — 北京:中国铁道出版社,2016.7

教育部高等学校文科大学计算机课程教学指导分委员会
立项教材 高等院校计算机基础教育应用型系列规划教材
ISBN 978-7-113-21866-9

I. ①C… II. ①温… 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135017号

书 名: C++ 程序设计
作 者: 温斯琴 主编

策 划: 刘丽丽
责任编辑: 周 欣
编辑助理: 吴 楠
封面设计: 一克米工作室
封面制作: 白 雪
责任校对: 汤淑梅
责任印制: 郭向伟

读者热线: (010) 63550836

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: [http:// www.51eds.com](http://www.51eds.com)

印 刷: 北京华正印刷有限公司

版 次: 2016年7月第1版 2016年7月第1次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 17 字数: 399 千

印 数: 1~2 000 册

书 号: ISBN 978-7-113-21866-9

定 价: 42.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

高等院校计算机基础教育应用型系列规划教材

编 审 委 员 会

主 任：斯日古楞（呼和浩特民族学院） 王彪（内蒙古财经大学）

杨宝勇（内蒙古电子信息职业技术学院）

副主任：王 胜（内蒙古师范大学） 苏布达（呼和浩特民族学院）

丽 平（呼和浩特民族学院） 温斯琴（呼和浩特民族学院）

梅 荣（呼和浩特民族学院） 李双月（呼和浩特民族学院）

冯培禄（呼和浩特民族学院） 于 鹰（呼和浩特民族学院）

委 员：（按姓氏拼音顺序排列）

敖金山（呼和浩特民族学院） 白领兄（呼和浩特职业学院）

白迎霞（呼和浩特民族学院） 宝勒尔（呼和浩特民族学院）

包乌格德勒（呼和浩特民族学院） 高迎梅（呼和浩特民族学院）

格勒娃（呼和浩特民族学院） 郝秉华（内蒙古财经大学）

李海军（呼和浩特民族学院） 李 娟（呼和浩特民族学院）

李 雪（内蒙古商贸职业学院） 刘 瑞（内蒙古财经大学）

宁宏亮（呼和浩特民族学院） 欧艳鹏（呼和浩特民族学院）

秦 鹏（呼和浩特民族学院） 色登丹巴（呼和浩特民族学院）

斯日古楞（呼和浩特民族学院） 斯 琴（呼和浩特民族学院）

苏雅丽（呼和浩特民族学院） 王平泉（呼和浩特民族学院）

西佳平（内蒙古电子信息职业技术学院）

新吉乐（呼和浩特民族学院） 姚浩斯拉（呼和浩特民族学院）

本书是教育部高等学校文科大学计算机课程教学指导分委员会立项教材，同时也是高等院校计算机基础教育应用型系列规划教材。

C++语言是目前广泛使用的一种程序设计语言，它是在C语言基础上发展演变而来的一种面向对象的程序设计语言。它支持面向过程的程序设计方法，也支持面向对象的程序设计方法。C++不仅扩充了C语言面向过程的功能，而且增加了面向对象的功能，这就使C++成为“能适应面向对象程序设计的现代大型语言”。同时C++增加了面向对象编程、数据抽象、类属编程等技术支持，通过继承和多态性，使程序具有很高的可重用性，使软件的开发和维护更加方便。C++语言既可以用于开发系统软件，也普遍用于开发应用软件，同时也广泛地应用于科研和教学。

全书内容上可以分为两大部分：前7章介绍C++语言的基础知识；后6章介绍C++语言面向对象程序设计。全书从简单的样例入手，精心设计每一章节的样例，由浅入深、讲解透彻，重点讲述解决问题的思路，培养学者计算思维能力；书中每章后都备有思考与练习，主要包括选择题、填空题和编程题，使学生及时巩固所学知识，对于检验学生的学习情况和准备相关的考试都有裨益。

本书由温斯琴任主编，于鹰、西佳平任副主编，王平泉参加编写。其中第1、2章由王平泉编写，第3、4章由西佳平编写，第5、6、7、13章由于鹰编写，第8、9、10、11、12章由温斯琴编写，全书由温斯琴统稿。在此向所有对本书的编写提供支持和帮助的领导和老师表示感谢！

由于编者水平有限，书中疏漏之处在所难免，希望广大读者批评指正。

编 者

2016年4月

第一部分 基础知识

第 1 章 C++ 语言概述	2
1.1 编程语言的发展	2
1.2 C++ 语言的特点	4
1.3 面向对象程序设计方法概述	5
1.4 C++ 语言的语法规则与书写格式	6
1.5 简单的 C++ 程序	7
1.6 C++ 程序的实现	8
思考与练习	10
第 2 章 数据类型、运算符与表达式	11
2.1 基本数据类型	11
2.2 数据类型转换	15
2.3 C++ 运算符与表达式	16
思考与练习	25
第 3 章 程序流程控制结构	27
3.1 C++ 语句和程序结构概述	27
3.2 选择结构	29
3.3 循环结构	35
3.4 goto 语句	41
思考与练习	42
第 4 章 数组	49
4.1 一维数组	49
4.2 二维数组	53
4.3 字符数组	56

思考与练习	61
第 5 章 指针与引用	66
5.1 指针的概念	66
5.2 指针变量	67
5.3 指针运算	71
5.4 指针与数组	73
5.5 动态内存分配	77
5.6 引用	81
思考与练习	83
第 6 章 函数	86
6.1 函数定义的一般形式	86
6.2 函数调用	91
6.3 变量及作用域	95
6.4 内部函数和外部函数	100
6.5 函数与数组	102
6.6 函数与指针	105
6.7 内联函数	109
6.8 函数的默认参数	111
6.9 函数重载	112
6.10 函数的嵌套调用	114
6.11 函数的递归调用	115
思考与练习	116
第 7 章 自定义数据类型	119
7.1 结构体	119
7.2 枚举类型	127
7.3 用 typedef 定义类型	129
思考与练习	130

第二部分 面向对象程序设计

第 8 章 类与对象	132
8.1 面向对象的程序设计概述	132
8.2 类的对象	137
8.3 this 指针	140
8.4 构造函数与析构函数	140
8.5 对象与指针	150

8.6	对象数组	152
8.7	常对象与常成员	152
	思考与练习	155
第 9 章	静态数据成员与友元	162
9.1	静态成员	162
9.2	友元	168
	思考与练习	173
第 10 章	运算符重载	178
10.1	运算符重载的概念	178
10.2	运算符重载的基本方法	179
10.3	运算符重载的一般规则	180
10.4	用成员函数重载运算符	181
10.5	用友元函数重载运算符	185
10.6	重载 ++、-- 运算符	191
10.7	重载赋值运算符	194
10.8	重载 [] 运算符	195
10.9	重载类型转换符	197
	思考与练习	198
第 11 章	继承与派生	204
11.1	继承与派生的概念	204
11.2	派生类的定义	206
11.3	派生类的构成	208
11.4	类的继承方式	209
11.5	派生类的构造函数和析构函数	214
11.6	多重继承	217
11.7	虚基类	221
11.8	赋值兼容规则	225
11.9	应用举例	226
	思考与练习	229
第 12 章	虚函数与多态性	237
12.1	编译时的多态性与运行时的多态性	237
12.2	虚函数	241
12.3	虚析构函数	243
12.4	纯虚函数与抽象类	244
	思考与练习	248

第 13 章 文件流与文件操作	249
13.1 文件流与文件类型	249
13.2 文件流与文件流对象	250
13.3 文件操作的流程	250
13.4 文本文件的读写操作	252
13.5 二进制文件的读写操作	254
13.6 文件的随机访问	257
思考与练习	260
参考文献.....	262

第一部分

基础知识

C++语言概述

教学目标：

通过本章的学习，了解C++语言的概念、发展历史、C++程序的组成和C++程序的实现。

教学重点和难点：

- C++程序的组成；
- 面向过程与面向对象的程序设计思想。

1.1 编程语言的发展

1.1.1 编程和编程语言

早期人们认为编程工作包含认识事物和描述事物两项内容，认为编程应先对需要解决的问题产生正确的认识，然后再用一种语言将它正确地描述出来。这样就把软件开发与编程看成一回事了。从软件工程的角度来看，软件开发和编程是不同的。软件开发包含两项主要活动：一是人们对要解决的问题的认识；二是对这种认识的描述。“认识”是指对要解决的问题进行周密的分析和全面理解，并找出解决的方法；“描述”是指选用一种语言来描述对要解决的问题的认识。可见，编程只是在“认识”基础上进行“描述”，编程时所选用的语言称为编程语言。因此，人们认为编程只是软件开发中的一项内容，而不是全部内容。

开发人员对于要解决的问题的认识又称为对问题域的认识。问题域是指要解决问题的集合，或者指要解决的问题所涉及的业务范围。人们对于问题域的认识往往与自然语言之间存在一个过渡。这个过渡形成的原因很简单，就是因为人们描述客观事物所用的自然语言是机器所不能识别的，而机器能够识别（直接或间接）的编程语言又不符合人们习惯的

思维方式。于是就形成了二者之间的鸿沟。鸿沟的存在实际上耗费了软件开发人员的许多精力，这同时也是许多错误的发源地。

1.1.2 编程语言的发展史

编程语言是从低级到高级逐渐发展的，具体过程如下：

1. 机器语言

机器语言是最原始的编程语言，这种语言是计算机可以直接识别的语言。它使用0和1两种代码，编写出的程序难以理解和记忆，因为它和人们习惯的思维方式相差较大。

2. 汇编语言

汇编语言使用助记符号来代替代码0和1，是一种低级语言。它比机器语言稍有提高，较为符合人们的形象思维，是低层次的抽象。计算机不能直接识别汇编语言，需要编译后才能识别。汇编语言虽然效率较高，但是由于难以记忆，人们使用较少。

3. 高级语言

高级语言是一种采用命令或语句的语言，它屏蔽了机器细节问题，提高了语言的抽象层次，比汇编语言更加接近人们的思维方式。人们容易理解和记忆这种语言，但它还是与自然语言有较大差别。直到20世纪70年代结构化程序设计语言的出现，编程逐渐变得方便起来，自然语言与编程语言的差距进一步缩小。

4. 面向对象语言

面向对象语言是比面向过程语言更高级的一种高级语言。

面向对象语言的出现改变了编程者的思维方式，使设计程序的出发点由着眼于问题域中的过程转向着眼于问题域中的对象及其相互关系。这种转变更加符合人们对客观事物的认识。因此，面向对象语言更接近自然语言，是人们对客观事物更高层次的抽象。

从编程语言发展的历史来看，编程语言由低级向高级发展，使得自然语言与编程语言之间的联系越来越紧密，这就意味着软件开发人员的精力耗费越来越少，软件产品的质量越来越高。

C++是在C语言的基础上演变而来的，而C语言又是从更老的两种程序语言演变而来的。这两种语言分别是BCPL和B。其中，BCPL是Martin Richards在1967年开发成功的，用于编写操作系统软件和编译程序。后来，Ken Thompson以BCPL为基础，在他的B语言中建模构造了许多相似的特性。20世纪70年代，贝尔实验室在一台DEC PDP-7计算机上，用B语言创建早期版本的UNIX操作系统。不管BCPL还是B都是“无类型”语言——每个数据项都在内存中占据一个“字”的空间，要由程序员自行决定将一个数据项作为整数还是作为实数来处理。

C语言是美国贝尔实验室的Dennis Ritchie在1972年根据B语言开发出来的，最初是为

了代替汇编语言而为小型机DEC PDP-11编写的UNIX操作系统使用。随着UNIX的推广，C语言本身也被广大程序设计人员了解和使用，从而得到流行。

C++是对C的一个扩展，它提供了数量众多的特性，目的是对C语言进行改进，但更重要的是它还引入了崭新的面向对象编程的能力。C++属于一种混合型语言，既可采用传统的C风格进行C++编程，也可采用新的面向对象的风格，甚至还能在同一个项目中混合采用两种风格。

➤➤➤ 1.2 C++语言的特点

C++语言是以C语言为基础，并在此基础上扩充、发展而来的，它是C语言的进化版本。Bjarne Stroustrup设计并实现C++的初衷是使C++语言不仅具有像Simula语言管理与组织大型程序的机制，同时又兼有C语言的高效性与灵活性；更重要的是欲使当时大量的C程序和C库函数得以继承使用，大批优秀的C程序员不丢弃长期积累的C编程经验，只需要学习C++加入的一些新特性就能快速、平滑地过渡到这种支持新的程序设计范型且表达力更强的语言。

C++语言不仅继承和发扬了C语言的优点，而且吸纳了其他众多语言的优良特性。例如，C++语言中的一些新特性，单行注释来源于BCPL's；类的概念，包括派生类及虚函数来源于Simula67；运算符重载及自由的变量声明来源于Algol68语言；模版机制主要受Ada语言的启发；错误处理主要来源于Ada、Clu和ML语言。C++语言的一些其他机制，如多重继承、纯虚函数、命名空间等是在C++语言的发展及应用过程中逐步产生的。

C++语言保留了C语言短小精简的风格，并对C语言的不足进行了改进。C++语言对C语言的改进表现如下：

- (1) 与C语言兼容，既支持面向对象的程序设计，也支持结构化的程序设计。同时，熟悉C语言的程序员，能够迅速掌握C++语言。
- (2) 修补了C语言中的一些漏洞，提供更好的类型检查和编译时的分析。使程序员在C++环境下继续编写C代码，也能得到直接的好处。
- (3) 生成目标程序质量高，程序执行效率高。一般来说，用面向对象的C++编写的程序执行速度与C语言程序不相上下。
- (4) 提供了异常处理机制，简化了程序的出错处理。利用throw、try和catch关键字，出错处理程序不必与正常的代码紧密结合，提高了程序的可靠性和可读性。
- (5) 函数可以重载，即可以使用默认参数。重载允许相同的函数名具有不同参数表，系统根据参数的个数和类型匹配相应的函数。默认参数可以使程序员能够以不同的方法调用同一个函数，并自动对某些默认参数提供默认值。

1.3 面向对象程序设计方法概述

1.3.1 传统的结构化程序设计

20世纪60年代,许多大型软件开发项目都面临严重困境:项目往往不能按时完成,实际成本常常超过预算,最终产品也变得极不稳定。人们逐渐意识到,软件开发的复杂程度远远超过自己的想象。因此,通过当时发起的一系列研究,最终促成了结构化编程概念的问世。这是一种比较严格的程序编写机制,用它可方便地写出思路清晰、高度准确、易于测试/调试、易于修改的程序。

结构化程序设计方法,在过去的几十年中,一直是程序设计开发的主导思想。具体做法是:为解决某个实际问题确定一种算法,然后为该算法构造合适的数据结构,通过算法的操作过程体现算法的思想,即程序是在数据某种特定的表示方式和结构的基础上对算法的具体实现,注重的是程序功能的描述。

1.3.2 面向对象的程序设计

面向对象的程序设计是20世纪80年代发展起来的一种程序设计方法,它通过模拟现实世界中的事物和关系,利用抽象、分类、归纳等方法来构造软件系统。

在面向对象程序设计出现之前,人们一直采用结构化程序设计来解决问题。结构化程序设计是面向过程的,其主要思想是将功能分解并逐步求精。

C++是一种面向对象的程序设计语言,它充分支持面向对象思想中的3个主要特征。

1. 封装性

封装性是指将数据和处理数据的方法封装在一起成为一个类。类的实体就是对象,对象一旦定义,就可以看成是一个完全封装的实体,可以作为一个整体单元来使用。类的内部数据表示被隐藏起来,类的用户不需要知道类内数据的表示方法,只需执行类对外提供的算法,就可以完成某项功能。

2. 继承性

继承性是指一种事物保留了另一种事物的全部特征,并且具有自身的独有特征。C++采用继承来支持重用,继承可以在现有类型的基础上扩展功能来定义新类型。新类型是从现有类型中派生出来的,因此被称为派生类。

3. 多态性

多态性是指当多种事物继承自一种事物时,同一种操作在它们之间表现出不同的行为。C++中使用虚函数和抽象类的概念来支持多态性。

1.4 C++语言的语法规则与书写格式

1.4.1 C++语言的基本符号

C++语言的基本符号可分为3种：字母（a～z、A～Z）、数字（0～9）和特殊符号。特殊符号包括：+、-、*、/=、,、_、:、;、?、\、"、'、~、|、!、#、%、&、()、[]、{}、^、<、>、空格。

1.4.2 C++语言的词汇

1. 关键字

关键字是预定义好的具有特殊含义的标识符，使用关键字时只能按照它们预定义好的意义来使用。用户在自定义标识符时，一定不能与这些关键字相同。C++语言中的关键字如图1-1所示。

asm	default	float	operator	static_cast	union
auto	delete	for	private	struct	unsigned
bool	do	friend	protected	switch	using
break	double	goto	public	template	virtual
case	dynamic_cast	if	register	this	void
catch	else	inline	reinterpret_cast	throw	volatile
char	enum	int	return	true	wchar_t
class	explicit	long	short	try	while
const	export	mutable	signed	typedef	const_cast
extern	namespace	sizeof	typeid	continue	false
new	static	typename			

图1-1 C++语言中的关键字

2. 标识符

标识符是用于表示应用程序中的变量、常量、类型、函数、类和对象的名字。在定义变量、类型、函数和类时通过声明一个标识符来创建它，然后就可以在后面的程序中使用该标识符来定义有关的内容。

C++的标识符只能由字符、下画线、数字组成，以字母或下画线开头，字母区分大小写。建议在自定义标识符中最好不要用下画线，因为系统定义的内容很多以下画线开头，所以容易引起冲突而产生错误。

3. 常量

常量是指在程序运行的整个过程中值始终不可改变的量。常量一般有两种表示形式，即符号常量和数值常量。例如，可以用 pi 来代表 3.14，这时需要先将 pi 定义为符号常量，3.14 是数值常量。C++语言中符号常量的名字是一个标识符。

4. 运算符

C++语言的运算符范围很广，把除了控制语句和输入/输出以外的几乎所有的基本操作都作为运算符处理，例如将“=”作为赋值运算符。

➤➤➤ 1.5 简单的C++程序

下面看一组简单的程序，输出一行文字。

```
1 //我的第一个程序
2 #include<iostream>
3 using namespace std;
4 void main( )
5 {
6     cout<<"Welcome to C++!";
7 }
```

运行结果：

```
Welcome to C++!
```

说明：

第1行以//开头，表明//后的内容为注释。程序员在编写程序时加适当的注释，以便于其他人阅读和理解所编写的程序，而且增强程序的可读性。程序运行时，注释不会导致计算机对其采取任何操作。以//开头的注释称为单行注释，如果注释包含多行，则注释以/*开头，以*/结尾。

第2行：

```
#include<iostream>
```

这是一条预处理命令，是发送给C++预处理器的一条消息。以#开头的行会在程序编译之前由预处理器进行处理。程序中的<iostream>是输入/输出流头文件。对任何程序，只要它需要C++风格的流输入/输出，将数据输出至屏幕，或从键盘输入数据，便应将上述头文件包括在内。在一个需要从键盘输入数据或者需要将数据输出至屏幕的程序中，假如忘了包括iostream文件，会导致编译器报告一条错误消息。

第3行：

```
using namespace std;
```

这一条语句的作用是，说明程序中将要使用std这个命名空间中的名字，例如对象命名

cout。命名空间是避免名字冲突的一种机制。

第4行：

```
void main( )
```

这是每一个C++程序都有的一部分。C++程序可包含一个或多个函数，但其中一个必须是main()函数。C++程序无论如何都会从main()函数处开始执行。main()前面的void关键字指出main()函数没有返回值。有关函数返回值的内容将在后面的章节中详细介绍。

第5行和第7行的花括号，必须用于每个函数主体的开始处和结束处。

第6行：

```
cout<<"Welcome to C++!";
```

本行完成了此程序的主要功能，即向屏幕上输出一行字符串。这里cout是一个输出流对象，包含许多有用的输出功能。“<<”是输出操作符，其作用是将紧随其后的双引号中的字符串原样输出到屏幕上。

➤➤➤ 1.6 C++程序的实现

1.6.1 C++源程序的编辑、编译和运行

1. 编辑

编辑功能就是将编写好的C++语言源程序录入到计算机中，生成磁盘文件加以保存。录入的方法可分为两类：一类是选用计算机中所提供的某种文件编辑器，将源程序代码录入到磁盘文件中，该文件的扩展名应为.cpp；另一类是选用C++编译器本身所提供的编辑器，这种方法比较方便，是一种经常采用的方法。例如，使用Visual C++ 6.0版本的编译器时，该编译器本身带有一个窗口编辑器，它具有编辑器的全部功能，使用起来很方便。

2. 编译

C++是一种以编译方式来实现的高级语言。C++的源程序必须经过编译后才能运行。编译工作是由系统提供的编译器来完成的。

编译器的功能是将程序的源代码转换成为目标代码，然后，再将目标代码进行连接，生成可执行文件。

整个编译过程可分为如下3个子过程：

(1) 预处理过程。源程序经过编译时，首先经过预处理过程，先执行程序中的预处理命令，执行后再继续后面的编译。