



从零开始，手把手教你用C++语言编程

本书特色：

基础知识→核心技术→典型应用→综合练习→项目实践
168个典型实例、90个练习题、2个项目开发案例

超值、大容量DVD

18小时多媒体视频教学
本书源代码、本书教学PPT
赠送19小时相关知识多媒体视频

本书技术支持

QQ群：21948169
论坛：<http://www.rzchina.net>

• 从零开始学编程 •

从零 开始学



C++

18小时多媒体教学视频

◎ 古超 赵丹 等编著

循序渐进·由浅入深 内容充实·层次清楚 实例丰富·步骤清晰 对比讲解·理解深刻 习题指导·巩固学习 案例精讲·深入剖析



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

• 从零开始学编程 •

从零 开始学

C++

◎ 古超 赵丹 等编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书由浅入深、循序渐进地讲解如何使用 C++ 开发环境。本书采用理论与实践相结合的写作方法, 不仅使读者对 C++ 的语法有全面的了解, 而且能让读者从实践中得到清晰的思路。

本书全面地讲述了 C++ 语言中的各个知识点, 共包括四篇, 内容包括 C++ 概述、C++ 开发利器大集合、C++ 初次编程、C++ 的数据类型、运算符和表达式、分解 C++——语句、数组的使用、灵活的指针、函数的使用、变量的生存范围、让数据到文本中去——文件的操作、预编译、字符串的处理、类和对象、财富的积累——继承、让程序变得更短——模板、代码的工厂——虚拟和多态、让系统变得更健康——异常处理等内容。在本书的每一章中都对相应的知识进行了串烧, 有利于学习者系统地复习所学知识。

本书配有 DVD 光盘 1 张, 内容为本书的实例文件和作者专门为本书录制的全程多媒体语音教学视频。

本书不仅适合 C++ 语言的初学者, 还适合 C++ 语言编程爱好者和有志于使用 C++ 语言进行编程的程序员作为参考和学习用书使用。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

从零开始学 C++ / 古超等编著. —北京: 电子工业出版社, 2010.12
(从零开始学编程)
ISBN 978-7-121-12260-6

I. ①从… II. ①古… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 221677 号

责任编辑: 胡辛征

文字编辑: 江 立

印 刷: 北京中新伟业印刷有限公司
装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 20 字数: 538 千字

印 次: 2010 年 12 月第 1 次印刷

印 数: 5000 册 定价: 39.80 元 (含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



如果没有学习 C 语言，那就直接从 C++ 开始吧！

——Bjarne Stroustrup (C++之父)

C/C++语言仍旧是使用范围最多的计算机编程语言，使用 C/C++语言开发出来的软件遍及到生活中的各个领域，且具有较高的性能和稳定性。相比较其他语言而言，其优势更加明显，例如，同样在 Windows 操作环境下，同时打开使用 C++开发的软件和使用 Java 开发的软件，前者的启动速度明显高于后者。

在刚开始学习 C++语言的时候，有些学习者很难找到一本合适的入门书籍，学习过程中充满着迷惑，往往不知道该怎么入手。本书在编写过程中充分考虑了这种情况，使读者在掌握理论知识的同时，实际的编程能力也会得到很大提高，本书的每个例子都进行了详细解释，并在程序后面附上了运行结果，真正实现了理论与实际相结合的学习方法。

本书的特点

C++语言的流行，使越来越多的人加入到 C++学习的大军中，但是在实际的学习过程中很难找到一本真正适合 C++初学者的教材，因此，市场上急需一本能够真正为初学者提供的参考书，从而适合各种类型的人员进行 C++的学习。作者在经过了深思熟虑之后，结合实际的情况对书本进行了编排，其特点如下：

- ❑ 从零开始，即使是没有任何编程基础的各层次人员都可以进行流畅的学习。
- ❑ 内容翔实，基本覆盖了 C++程序语言的各个知识点，为学习者以后的深入学习奠定了基础。
- ❑ 结合实际，从基本出发，对每一个程序都进行了实际的编程运行，保证了程序的正确性和可读性。
- ❑ 语言通俗易懂，使枯燥的理论学习更加易于接受。

① **知识点介绍** 准确、清晰是其显著特点，一般放在每一节开始位置，让零基础的读者了解相关概念，顺利入门。

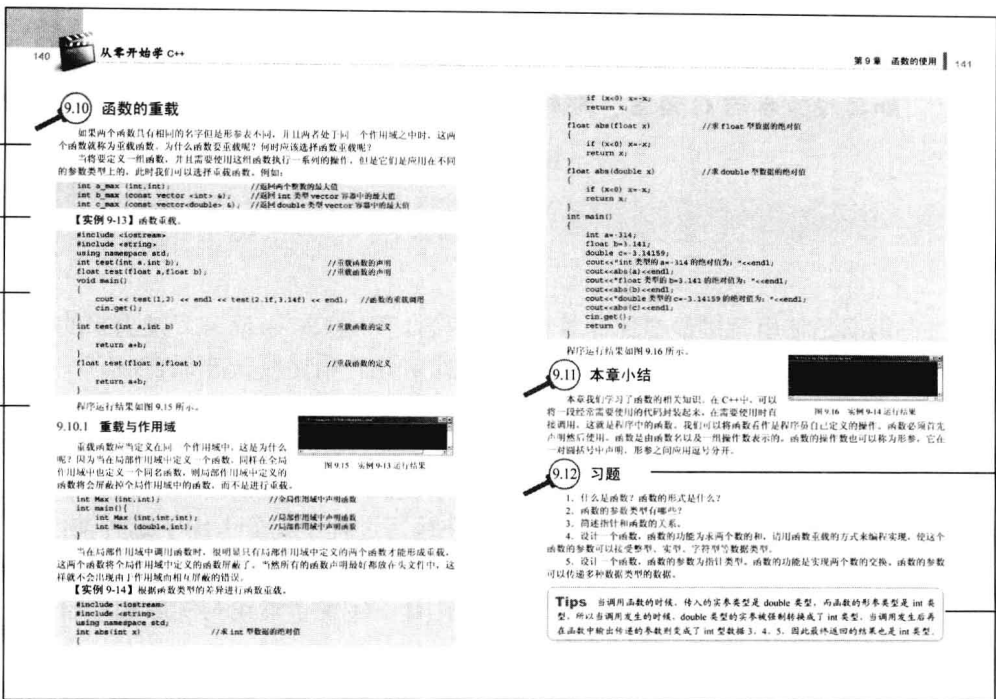
② **实例** 书中出现的完整实例，以章节顺序编号，便于检索和循序渐进地学习、实践，放在每节知识点介绍之后。

③ **实例代码** 与实例编号对应，层次清楚、语句简洁、注释丰富，体现了代码优美的原则，有利于读者养成良好的代码编写习惯。对于大段程序，均在每行代码前设定编号便于学习。

④ **运行结果** 对实例给出运行结果和对应图示，帮助读者更直观地理解实例代码。

⑤ **习题** 每章最后提供专门的测试习题，供读者检验所学知识是否牢固掌握。

⑥ **贴心的提示** 为了便于读者阅读，全书还穿插着一些技巧、提示等小贴士，体例约定如下：



提示：通常是一些贴心的提醒，让读者加深印象或提供建议，或者解决问题的方法。

注意：提出学习过程中需要特别注意的一些知识点和内容，或者相关信息。

警告：对操作不当或理解偏差将会造成的灾难性后果做警示，以加深读者印象。

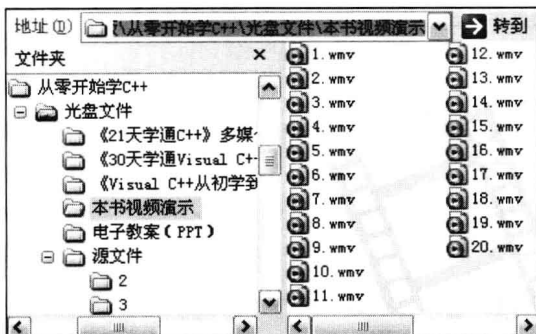
经作者多年的培训和授课证明，以上讲解方式是最适合初学者学习的方式，读者按照这种方式，会非常轻松、顺利地掌握本书知识。

2. 实用超值的 DVD 光盘

为了帮助读者比较直观地学习，本书附赠 DVD 光盘，内容包括多媒体视频、电子教案（PPT）和实例源代码等。

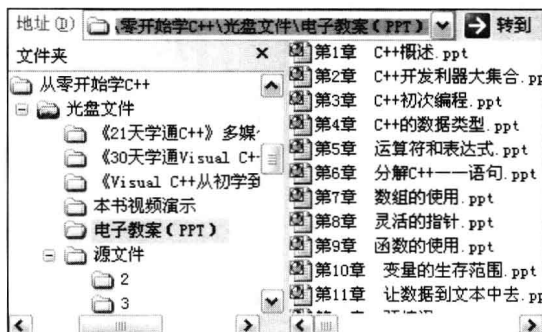
● 多媒体视频

配有长达 18 小时手把手教学视频，讲解关键知识点界面操作和书中的一些综合练习题。作者亲自配音、演示，手把手教会读者使用。



● 电子教案（PPT）

本书可以作为高校相关课程的教材或课外辅导书，所以笔者特别为本书制作了电子教案（PPT），以方便老师教学使用。



3. 提供完善的技术支持

本书提供了论坛：<http://www.rzchina.net>，读者可以在上面提问交流。另外，论坛上还有一些小的教程、视频动画和各种技术文章，可帮助读者提高开发水平。

4. 丰富的额外素材下载

相关的开发素材文件，在 www.broadview.com.cn 提供下载。

适合阅读本书的读者

- ☐ 想从事编程，且对 C++ 有兴趣的人
- ☐ 各大中专院校的在校学生和相关授课老师
- ☐ 了解 C++，但还需要进一步学习的人
- ☐ 其他编程爱好者

本书作者

本书主要由古超和沈阳市电子技术学校的赵丹编写。其他参与本书编写的人员有昊燃、刘永纯、曾光、张双、朱照华、黄永湛、孟祥嵩、张贺军、李勇、关涛、王岩、李晓白、魏星，在此一并表示感谢！

目 录

第 1 篇 C++入门

第 1 章 C++概述 ( 教学视频: 49 分钟)	15
1.1 C++与 C	15
1.1.1 C 语言	15
1.1.2 C++	17
1.1.3 C 和 C++比较	17
1.2 程序和算法	19
1.2.1 程序	19
1.2.2 算法	19
1.3 程序设计方法	21
1.3.1 面向过程的程序设计	21
1.3.2 面向对象的程序设计	22
1.4 C++编程风格	24
1.5 C++编程的过程	25
1.6 本章小结	25
1.7 习题	26
第 2 章 C++开发利器大集合 ( 教学视频: 31 分钟)	27
2.1 Visual C++与集成开发环境	27
2.1.1 Visual C++概述	27
2.1.2 Visual C++ 6.0 集成开发环境介绍	27
2.1.3 设置个性化的开发环境	32
2.1.4 创建一个简单的程序	33
2.2 C++ Builder	34
2.2.1 C++ Builder IDE (集成开发环境) 简介	34
2.2.2 创建一个简单的程序	35
2.3 Linux/UNIX 下的 C++开发工具	35
2.4 本章小结	36
2.5 习题	36
第 3 章 C++初次编程 ( 教学视频: 39 分钟)	37
3.1 初识输入与输出	37
3.2 注释符	38
3.3 控制语句简介	39
3.3.1 if 语句	39

3.3.2	for 语句	40
3.3.3	while 语句	41
3.4	类	42
3.4.1	创建一个简单的类	42
3.4.2	初识成员函数	43
3.5	一个简单的 C++ 程序	45
3.6	本章小结	47
3.7	习题	47

第 2 篇 C++ 的利器

第 4 章	C++ 的数据类型 ( 教学视频: 120 分钟)	48
4.1	基本内置类型	48
4.1.1	整型	48
4.1.2	浮点型	50
4.2	常量	51
4.2.1	整数类型字面值常量	51
4.2.2	布尔型和字符型字面值常量	51
4.2.3	浮点型字面值常量	51
4.2.4	字符串字面值常量	51
4.3	变量	52
4.3.1	什么是变量	52
4.3.2	变量名的确定	53
4.3.3	变量的声明与定义	53
4.3.4	初始化变量	53
4.4	枚举类型	54
4.5	自定义类型 typedef	55
4.6	引用	55
4.7	const 限定符	56
4.8	类类型	57
4.9	标准库类型	58
4.9.1	string 类型	58
4.9.2	顺序容器类型	60
4.9.3	关联容器类型	62
4.10	标准 I/O 库类型	64
4.10.1	输入输出流	64
4.10.2	字符串流	65
4.11	本章小结	66
4.12	习题	66
第 5 章	运算符和表达式 ( 教学视频: 64 分钟)	67
5.1	算术运算符与算术表达式	67

5.1.1	算术运算符	67
5.1.2	算术表达式	68
5.2	关系运算符与关系表达式	69
5.2.1	关系运算符	69
5.2.2	关系表达式	70
5.3	逻辑运算符与逻辑表达式	70
5.3.1	逻辑运算符	70
5.3.2	逻辑表达式	71
5.4	赋值运算符与赋值表达式	72
5.4.1	赋值运算符	72
5.4.2	赋值表达式	73
5.5	条件运算符与条件表达式	74
5.6	逗号运算符与逗号表达式	74
5.7	自增和自减运算符及其表达式	75
5.8	sizeof 运算符	76
5.9	箭头运算符与点运算符	77
5.10	位运算符及其表达式	78
5.11	new 和 delete 表达式	79
5.12	复合表达式	80
5.13	本章小结	82
5.14	习题	82
第 6 章	分解 C++——语句 ( 教学视频: 63 分钟)	83
6.1	简单语句	83
6.2	声明语句	84
6.3	复合语句	85
6.4	语句的作用域	86
6.5	分支语句	86
6.5.1	条件语句	86
6.5.2	switch 语句	91
6.6	循环语句	93
6.6.1	for 语句	93
6.6.2	while 语句	95
6.6.3	do...while 语句	96
6.6.4	嵌套循环语句	97
6.6.5	break 语句	98
6.6.6	continue 语句	100
6.6.7	goto 语句	100
6.6.8	return 语句	101
6.7	综合应用	102
6.8	本章小结	108

6.9	习题	108
第 7 章	数组的使用 ( 教学视频: 36 分钟)	109
7.1	数组的概念	109
7.2	一维数组	109
7.2.1	一维数组的定义	109
7.2.2	一维数组的初始化	110
7.2.3	一维数组元素的访问	112
7.2.4	字符数组	112
7.3	多维数组	113
7.3.1	多维数组的定义	114
7.3.2	多维数组的初始化	114
7.3.3	多维数组元素的访问	116
7.4	数组的应用	117
7.5	本章小结	125
7.6	习题	125
第 8 章	灵活的指针 ( 教学视频: 79 分钟)	126
8.1	什么是指针	126
8.2	指针的定义和初始化	128
8.2.1	指针变量的定义	128
8.2.2	获得地址并对指针变量进行初始化	129
8.2.3	指针变量的使用	130
8.2.4	指针与引用	132
8.3	指针的操作	133
8.4	指向指针的指针	135
8.5	使用指针访问数组元素	136
8.5.1	指向数组元素的指针	136
8.5.2	一维数组与指针	137
8.5.3	多维数组与指针	140
8.6	指针和 <code>const</code> 限定符	144
8.6.1	指向 <code>const</code> 对象的指针	144
8.6.2	<code>const</code> 指针	145
8.7	指针和字符串	145
8.8	本章小结	146
8.9	习题	146
第 9 章	函数的使用 ( 教学视频: 81 分钟)	147
9.1	函数的概念	147
9.2	函数的声明	148
9.3	函数的定义	150
9.3.1	不同函数类型的定义形式	150

9.3.2	函数的返回值	152
9.3.3	函数的形参表	153
9.4	函数的参数传递	154
9.4.1	形式参数与实际参数	154
9.4.2	非引用形参	155
9.4.3	引用形参	155
9.5	函数的调用	158
9.6	函数的嵌套调用	158
9.7	内联函数	159
9.8	变量的存储方式	161
9.9	指针和函数	161
9.9.1	指针参数	161
9.9.2	指针作为返回值	162
9.10	函数的重载	164
9.10.1	重载与作用域	165
9.10.2	默认参数	166
9.10.3	参数类型上不同的重载函数	166
9.10.4	参数个数上不同的重载函数	167
9.11	本章小结	168
9.12	习题	168
第 10 章	变量的生存范围 ( 教学视频: 26 分钟)	169
10.1	局部作用域	169
10.2	全局作用域	170
10.3	变量的存储类型	171
10.3.1	自动存储类型	171
10.3.2	静态存储变量	171
10.3.3	寄存器存储类型	172
10.4	定义和使用命名空间	173
10.5	本章小结	176
10.6	习题	176
第 11 章	让数据到文本中去——文件的操作 ( 教学视频: 42 分钟)	177
11.1	文件的概述	177
11.2	I/O 流的描述	177
11.3	文件的打开模式	178
11.4	打开一个文件	178
11.5	向文件中写入数据	181
11.5.1	文本文件的写入方法	181
11.5.2	二进制文件的写入方法	182
11.5.3	将结构体数据写入到文件	182
11.6	从文件中读出数据	183

11.6.1	get()方法	183
11.6.2	getline()方法	183
11.6.3	rdbuf()方法	184
11.6.4	read()方法	184
11.7	文件指针的定位	185
11.7.1	确定文件指针在文件流中的位置 (tellp()和 tellg())	185
11.7.2	移动文件流中指针 (seekg()和 seekp())	187
11.8	关闭文件	188
11.9	文件处理的实例	189
11.10	本章小结	192
11.11	习题	192
第 12 章	预编译 ( 教学视频: 28 分钟)	193
12.1	宏定义	193
12.1.1	不带参数的宏定义	193
12.1.2	带参数的宏定义	194
12.2	#include 文件包含	196
12.3	条件编译	196
12.3.1	什么是条件编译	196
12.3.2	几种常用的条件编译格式	197
12.4	预定义宏	199
12.5	本章小结	200
12.6	习题	200
第 13 章	字符串的处理 ( 教学视频: 44 分钟)	201
13.1	string 对象的定义和初始化	201
13.2	字符串类的成员函数	202
13.3	字符串处理类的操作符	204
13.4	字符串类中的指针	206
13.5	string 类串和 C 风格串之间的转换	207
13.6	字符串应用实例	208
13.7	本章小结	212
13.8	习题	212

第 3 篇 C++高级特性

第 14 章	类和对象 ( 教学视频: 78 分钟)	213
14.1	类	213
14.1.1	类的定义	213
14.1.2	类的 struct 定义方式	216
14.2	定义一个类的实例——对象	217
14.3	类的构造函数	217

14.3.1	构造函数的定义	217
14.3.2	默认构造函数	219
14.3.3	构造函数的默认实参	220
14.3.4	重载构造函数	221
14.3.5	类类型的隐式转换	224
14.4	类的成员函数	225
14.4.1	成员函数的定义	225
14.4.2	成员函数的调用	226
14.4.3	常成员函数	228
14.4.4	重载成员函数	229
14.5	类中的静态成员	230
14.5.1	静态数据成员	230
14.5.2	静态成员函数	231
14.6	友元	233
14.7	操作符的重载	234
14.8	本章小结	235
14.9	习题	235
第 15 章	财富的积累——继承 ( 教学视频: 46 分钟)	236
15.1	继承的概念	236
15.2	定义基类与派生类	236
15.3	继承方式	237
15.3.1	公有继承	237
15.3.2	保护继承	238
15.3.3	私有继承	239
15.4	派生类的构造与析构	240
15.5	组合与继承	241
15.6	赋值兼容原则	243
15.7	多重继承	244
15.7.1	多重继承的结构	244
15.7.2	多重继承中的构造与析构	246
15.8	本章小结	247
15.9	习题	247
第 16 章	让程序变得更短——模板 ( 教学视频: 59 分钟)	248
16.1	模板的概念与意义	248
16.2	函数模板	249
16.2.1	函数模板的定义	249
16.2.2	函数模板的用法	250
16.2.3	函数模板的重载	251
16.3	类模板	252
16.3.1	类模板的定义	252

16.3.2	类模板的实例化	254
16.3.3	模板类和类模板	255
16.3.4	类模板的应用	255
16.4	本章小结	261
16.5	习题	261
第 17 章	代码的工厂——虚拟和多态 ( 教学视频: 43 分钟)	262
17.1	多态的定义与实现	262
17.1.1	多态的定义	262
17.1.2	多态的实现	264
17.2	虚拟函数的定义与实现	268
17.2.1	虚函数	268
17.2.2	虚拟析构函数	270
17.2.3	避免虚函数的误用	271
17.3	抽象类与纯虚函数	273
17.3.1	纯虚函数	273
17.3.2	抽象类和具体类	274
17.4	本章小结	276
17.5	习题	276
第 18 章	让系统变得更健康——异常处理 ( 教学视频: 62 分钟)	277
18.1	异常处理的含义	277
18.2	抛出异常	278
18.3	捕获异常	279
18.4	使用异常	281
18.4.1	何时避免异常	281
18.4.2	异常的典型应用	282
18.4.3	非错误处理	283
18.5	标准异常处理类	284
18.6	本章小结	285
18.7	习题	286

第 4 篇 综合案例

第 19 章	学生成绩管理系统 ( 教学视频: 58 分钟)	287
19.1	概要设计	287
19.2	详细设计	287
19.3	代码开发	289
19.3.1	学生类的定义	289
19.3.2	操作类的定义	290
19.3.3	设置学生姓名 <code>setname(char na[])</code>	290
19.3.4	设置学生学号 <code>setNo(int n)</code>	291

19.3.5	设置英语成绩 setEng(float e)	291
19.3.6	设置语文成绩 setChi(float c)	291
19.3.7	设置数学成绩 setMat(float m)	291
19.3.8	设置总分 settotal(float t)	291
19.3.9	获取姓名 getname()	291
19.3.10	获取学号 getNo()	291
19.3.11	获取英语成绩 getEng()	292
19.3.12	获取语文成绩 getChi()	292
19.3.13	获取数学成绩 getMat()	292
19.3.14	获取总分 gettotal()	292
19.3.15	输入学生信息 getdata()	292
19.3.16	总分排序 sort()	293
19.3.17	显示所有学生记录 disp()	293
19.3.18	初始化界面 initdisp()	294
19.3.19	添加数据记录 adddata()	294
19.3.20	按照学号查找记录 namefind(int)	295
19.3.21	输入学号 inputonename()	296
19.3.22	删除学生记录 dele(int num)	296
19.3.23	计算平均分函数	297
19.3.24	返回记录数目 getns()	297
19.3.25	退出系统 quit()	297
19.3.26	程序主函数 main	298
19.4	本章小结	299
第 20 章	银行自动操作终端系统 ( 教学视频: 27 分钟)	300
20.1	概要设计	300
20.2	详细设计	300
20.3	代码开发	303
20.3.1	普通用户账户信息类的定义	303
20.3.2	普通用户账户信息类的实现	304
20.3.3	普通用户个人信息类的定义	304
20.3.4	普通用户个人信息类的实现	305
20.3.5	普通用户个人操作信息类的定义	306
20.3.6	普通用户个人操作信息类的实现	306
20.3.7	封装类的定义	307
20.3.8	操作类的定义	309
20.3.9	声明关于输出的函数	309
20.3.10	系统实现主函数	310
20.4	程序效果	316
20.5	本章小结	317

第 1 篇 C++入门

第 1 章 C++概述

在网上论坛中，C++初学者的第一个问题经常是“我该怎么学习 C++？”或者“学习 C++难么？”，针对这样的问题不同人会给出不同的答案。自从 C++产生以来，人们的编程习惯开始逐渐改变，直到今天，C++语言仍然在软件开发领域中占据着重要的地位，越来越多的人开始使用 C++，C++正在软件开发领域中发挥着它自己的独特优势，越来越受到广大程序员的欢迎和追捧。

C++既继承了 C 语言的一些特点，又有着自身独特的优势，C++综合了各种语言的长处，有着越来越多的功能，使用 C++可以更加容易和快捷地进行大型复杂软件的开发。当然，既然 C++有着这么多的优势，学习起来自然是有一定困难的。有的人说 C++很难，这很正常，如果有的东西一学就会，试问这个东西还有什么价值。因此要想学习 C++就要坚持不断地去学习，至于最后能够达到什么样的结果这就要因人而异了，有的人可能会成为一个 C++高手，在 C++的世界里纵横驰骋，有的人可能只领略到了 C++的一些皮毛，可以进行一些简单的应用，但对于 C++中的高级技术可能无缘使用了。因此，最终要达到一种什么样的境界完全取决于学习者本人，看学习者自己的学习态度如何了。

在本章中，我们首先对 C 语言和 C++语言的历史做一个介绍并比较一下它们各自的特点，然后会对面编程的思想和 C++的编程风格进行详细的介绍，从而对这门语言有一个大致的了解，俗语说得好：“知己知彼方能百战百胜”，因此在这一章中主要向读者全面介绍一下 C++，让读者在以后的学习过程中更加游刃有余。



1.1 C++与 C

有人说 C++语言是从 C 语言发展过来的，下面我们就先从 C 语言开始来探讨一下 C 语言和 C++语言的发展历史。

1.1.1 C 语言

早期的操作系统等系统软件主要是用汇编语言编写的，如 UNIX 操作系统。由于汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好改用高级语言，但一般高级语言难以实现汇编语言的某些功能，而汇编语言可以直接对硬件进行操作，例如，对内存地址的操作、对位（bit）操作等。人们设想能否找到一种既具有一般高级语言特性，又具有低级语言特性的语言，于是 C 语言就应运而生了，之后成为国际上广泛流行的计算机高级语言。它适合作为系统描述语言，既可用于来写系统软件，也可用来写应用软件。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序，1963 年英国剑桥大学推出了 CPL（Combined Programming Language）。CPL 在 ALGOL 60 的基础上更接近硬件，但规模比较大，难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 作了简化，推出了 BCPL（Basic Combined Programming Language）。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 为基础，又作了进一步简化，它使得 BCPL 能挤压在 8KB 内存中运行，这个很简单而且很接近硬件的语言就是 B 语言（取 BCPL 的第一个字母），并用它写了第一个 UNIX 操作系统，



在 DEC PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语言，并写了 UNIX 操作系统。但 B 语言过于简单，功能有限，并且和 BCPL 都是“无类型”的语言。

1972 年至 1973 年间，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练、接近硬件），又克服了它们的缺点（过于简单、数据无类型等）。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工具语言而设计的。1973 年，K. Thompson 和 D. M. Ritchie 两人合作把 UNIX 中的 90% 以上的功能用 C 语言进行了编写，即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国贝尔实验室的 K. Thompson 和 D. M. Ritchie 开发成功的，是用汇编语言写的。这样，UNIX 使分散的计算系统之间的大规模联网以及互联网成为可能。

后来，C 语言多次进行了改进，但主要还是在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本——可移植 C 语言编译程序，使 C 移植到其他机器时所需做的工作大大简化了，这也推动了 UNIX 操作系统迅速地在各种机器上实现。例如，VAX、AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用，C 语言也得到迅速推广。C 语言和 UNIX 可以说是一对孪生兄弟，在发展过程中相辅相成。1978 年以后，C 语言已先后移植到大、中、小、微型机上，如 IBM System/370、Honeywell 6000 和 Interdata 8/32，且已独立于 UNIX 和 PDP 了。现在 C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

以 1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表的 UNIX 第 7 版中的 C 编译程序为基础，Brian W. Kernighan（柯尼汉）和 Dennis M. Ritchie（里奇）合著了影响深远的名著《The C Programming Language》，我们常常称它为“K&R”，也有人称之为“K&R 标准”或“白皮书”（White Book），它成为后来广泛使用的 C 语言版本的基础，但在 K&R 中并没有定义一个完整的标准 C 语言。为此，1983 年，美国国家标准化协会（ANSI）X3J11 委员会根据 C 语言问世以来的各种版本对 C 进行了发展和扩充，制定了新的标准，称为 ANSI C，ANSI C 比原来的标准 C 有了很大的发展。K&R 在 1988 年修改了他们的经典著作《The C Programming Language》，按照 ANSI C 标准重写了该书。1987 年，ANSI 又公布了新标准——87 ANSI C，目前流行的 C 编译系统都是以它为基础的。当时广泛流行的各种版本的 C 语言编译系统虽然基本部分是相同的，但也有一些不同之处。在微型机上使用的有 Microsoft C（MS C）、Borland Turbo C、Quick C 和 AT&T C 等，它们的不同版本又略有差异。到后来的 Java、C++、C#都是以 C 语言为基础发展起来的。

1989 年，C 语言被 ANSI 标准化（ANSI X3.159-1989），标准化的目的是扩展 K&R C，这个标准包括了一些新的特性。在 K&R 推出后，一些新的特征被“非官方”地加到 C 语言中。

在 ANSI 标准化自己的过程中，一些新的特征被加了进去。ANSI 也标准了函数库。ANSI C 标准被 ISO（国际标准化组织）采纳成为 ISO 9899。ISO 的第一个版本文件在 1990 年出版。

在 ANSI 标准化后，C 语言的标准在相当一段时间内都保持不变，尽管 C++继续在改进。实际上，Normative Amendment 1 在 1995 年已经开发了一个新的 C 语言版本，但是这个版本鲜为人知。标准在 20 世纪 90 年代才经历了改进，这就是 ISO 9899:1999（1999 年出版），这个版本就是我们通常提及的 C99，它被 ANSI 于 2000 年 3 月采用。

在 C99 中包括的特性有：

- ❑ 可变范围的数组；
- ❑ 新增加的数据类型，包括 long long int、布尔类型和用于表示复数的类型；
- ❑ 支持用//表示注释（这个特性实际上在 C89 的很多编译器上已经被支持了）。

但是各个公司对 C99 的支持所表现出来的兴趣不同。当 GCC 和其他一些商业编译器支持 C99 的大部分特性的时候，微软和 Borland 却似乎对此并不感兴趣。