

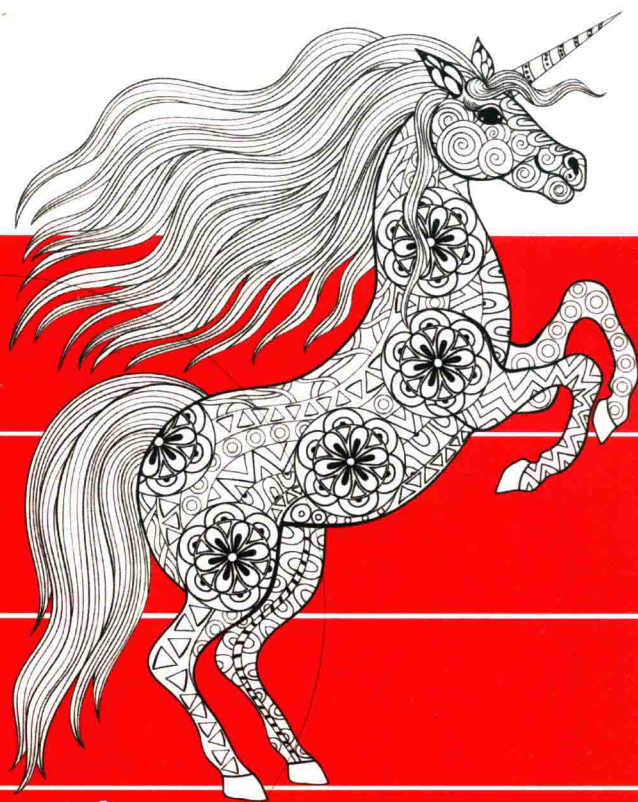


高等院校“十三五”规划教材

新标准 C++ 程序设计

◎ 严 悍 陆建峰 袁 宜 编著

New
Standard
C++
Programming



新标准 C++ 程序设计

严 悍 陆建峰 衷 宜 编著

东南大学出版社
· 南京 ·

内 容 提 要

C++ 是国内外广泛采用的编程语言,应用于多种计算平台,国内很多高校都开设 C/C++ 编程的相关课程,也出现了数百种相关教材。C/C++ 语言在 2011 年之前主要采用 C99 和 2003 标准。2011 年国际标准化组织和国际电工委员会发布了 C++ 11 新标准,推出近百个新语言特征,之后 C++ 14 和 C++ 17 进一步完善了新标准。新标准引入许多新概念、新规则,使得 C++ 编程表达复杂多变,初学者感到学习实践难度较大。本书采用研讨加实践的方式,力图使初学者能熟练掌握新概念、新规则,并增强编程求解能力。

本书共 15 章,主要分为两部分:第 1 部分(前 8 章)主要介绍结构化编程和函数式编程,第 2 部分(后 7 章)主要介绍面向对象编程和泛型编程。

本书可作为大学各学科专业学生学习实践 C++ 的基础教材,也适合作为软件开发人员的自学用书和研究人员的参考用书。

图书在版编目(CIP)数据

新标准 C++ 程序设计 / 严悍,陆建峰,衷宜编著. —南京:
东南大学出版社,2018. 8

ISBN 978-7-5641-7847-5

I. ①新… II. ①严… ②陆… ③衷… III. ①C++ 语
言-程序设计 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字(2018)第 153342 号

新标准 C++ 程序设计

出版发行 东南大学出版社
社 址 南京市四牌楼 2 号(邮编:210096)
出版人 江建中
责任编辑 吉雄飞(联系电话:025-83793169)
经 销 全国各地新华书店
印 刷 虎彩印艺股份有限公司
开 本 787mm × 1092mm 1/16
印 张 29.5
字 数 755 千字
版 次 2018 年 8 月第 1 版
印 次 2018 年 8 月第 1 次印刷
书 号 ISBN 978-7-5641-7847-5
定 价 90.00 元

本社图书若有印装质量问题,请直接与营销部联系,电话:025-83791830。

前 言

C++ 语言体现了当前过程性编程语言的主导思想,并得到广泛应用。C++ 语言表达简洁、灵活多样、计算性能高、平台支持度高,但同时 C++ 语言类型复杂、变化多端、理解较困难,对初学者入门有一定难度。C++ 语言在 2011 年、2014 年和 2017 年经历了三次语言标准升级,核心语言发生巨大变化,在改进传统的结构化编程和面向对象编程基础上引入了函数式编程和泛型编程,强类型弱化为静态类型,而编译器具有编译期运行能力,融合多种语言特征,如 Java, NodeJS/ECMAScript, GO, Python 等。因此,C++ 初学者和程序员都迫切需要重新理解掌握新标准 C++ 语言的新概念和新规则。

本教材编写秉承“内容新颖,概念清晰,规则分明,指导性与实用性并重”的原则,所具特色如下:

- (1) 新概念:涵盖 C++ 11 全部新概念与 C++ 14 部分已实现概念;
- (2) 新平台:支持最新 VS2017 和 DevC++ (GCC) 两大平台;
- (3) 新体系:新概念融入一个整体理论体系,使学生一次性掌握新概念和新规则;
- (4) 新展示:大量图表便于学生理解和教师讲授,且例题丰富,练习题形式多样。

本书共 15 章,主要分为以下两个部分:

第 1 部分(前 8 章),主要介绍结构化编程与函数式编程。其中,第 1 章概括列出新标准语言的新特征,有经验的读者可选择阅读;第 2 章到第 7 章介绍基本类型与变量,运算符与表达式,基本语句,函数和编译预处理,数组与字符串,结构、枚举和联合体;第 8 章介绍指针和引用,也介绍了基于 Lambda 的函数式编程。

第 2 部分(后 7 章),主要介绍面向对象编程与泛型编程。其中,第 9 章到第 12 章介绍新标准面向对象编程新特征;第 13 章介绍基于模板的泛型编程(这是 C++ 难点集中之处);第 14 章介绍输入输出流,不涉及语言特征;第 15 章介绍异常处理。

本书各章后配有小结和练习题,供读者复习和实践。书中所有的编码实例都采用 Visual Studio 2017/C++ 和 DevC++ (GCC) 作为开发环境,前者新标准符合度高但规模庞大,后者短小实用但新标准符合度稍差,运行库支持不足。本书尝试将所有实例在两个平台上运行比较,但略有缺失。附录中给出 ASCII 码表和部分常用函数库,以方便读者查阅。

本书由南京理工大学计算机科学与工程学院软件工程系 C++ 教学团队集体编写修订,获得南京理工大学“十三五”规划教材出版支持。在本书编写过程中编者得到多方支持,高锦博、高云等参与文字校对工作,在此向他们表示感谢。书中部分内容选自同行专家、学者的教材和专著,参考文献中力求全面列出,如有疏忽和遗漏,编者致以歉意并谨表感谢。本书不足之处,竭诚希望广大读者指正。

编者

2018 年 3 月

目 录

第1章 概述	1
1.1 C++ 语言发展历史	1
1.2 一个简单的 C++ 程序	2
1.3 C++ 程序的开发步骤	3
1.4 开发工具简介	4
1.5 C++ 标准及开发工具	6
1.6 C++ 11 与 C++ 14 新特征	7
1.7 本书组织结构	10
1.8 类型大图及导读	10
第2章 基本类型与变量	13
2.1 关键字和标识符	13
2.1.1 关键字	13
2.1.2 标识符	14
2.1.3 标点符号	15
2.1.4 分隔符与标记	15
2.2 基本类型	15
2.2.1 逻辑型	16
2.2.2 字符型	17
2.2.3 整数型	17
2.2.4 浮点型	18
2.2.5 空类型	19
2.3 字面值	19
2.3.1 逻辑值	19
2.3.2 整型值	19
2.3.3 浮点值	20
2.3.4 字符值	20
2.3.5 字符串值	22
2.4 变量	22
2.4.1 变量的说明	22
2.4.2 变量的初始化	23
2.4.3 auto 初始化	23
2.4.4 变量的赋值	24

2.4.5 变量的输入输出	24
2.4.6 命名常量	27
小结	27
练习题	27
第3章 运算符与表达式	29
3.1 基本运算符	29
3.1.1 算术运算符	31
3.1.2 关系运算符	32
3.1.3 逻辑运算符	33
3.1.4 位运算符	34
3.1.5 条件运算符	36
3.1.6 赋值运算符	36
3.1.7 逗号运算符	37
3.1.8 自增自减运算符	37
3.1.9 sizeof 运算符	39
3.1.10 typeid 运算符	39
3.1.11 其他运算符	40
3.2 表达式	41
3.2.1 左值表达式和右值表达式	41
3.2.2 表达式语句	41
3.2.3 表达式类型与 decltype	42
3.3 类型转换	42
3.3.1 自动类型转换	42
3.3.2 赋值类型转换	44
3.3.3 强制类型转换	45
小结	46
练习题	46
第4章 基本语句	49
4.1 语句分类	49
4.2 结构化编程基本结构	49
4.2.1 顺序结构	50
4.2.2 选择结构	50
4.2.3 循环结构	51
4.3 选择语句	52
4.3.1 条件语句	52
4.3.2 switch 语句	55
4.4 循环语句	56

4.4.1	while 语句	57
4.4.2	do-while 语句	58
4.4.3	for 语句	59
4.4.4	循环语句的比较	60
4.4.5	循环的嵌套	60
4.5	跳转语句	61
4.5.1	break 语句	61
4.5.2	continue 语句	62
4.5.3	goto 语句与标号语句	63
4.6	综合示例	63
	小结	66
	练习题	67
第 5 章	函数和编译预处理	69
5.1	函数基本概念	69
5.1.1	库函数和用户定义函数	69
5.1.2	无参函数和有参函数	69
5.1.3	有返回函数和无返回函数	70
5.2	函数的定义	70
5.2.1	传统函数定义	70
5.2.2	函数定义的要点	71
5.3	函数的调用	72
5.3.1	函数调用的形式	72
5.3.2	函数调用的方式	72
5.3.3	函数调用与以值传递	73
5.4	函数重载	74
5.4.1	重载函数的定义	75
5.4.2	重载函数的调用	75
5.5	嵌套调用和递归调用	76
5.5.1	函数的嵌套调用	76
5.5.2	函数的递归调用	77
5.6	函数原型	82
5.7	auto 函数与尾随返回类型	83
5.8	特殊参数	84
5.8.1	带缺省值的形参	84
5.8.2	可变参数	85
5.9	inline 函数与 constexpr 函数	87
5.9.1	inline 函数	87
5.9.2	constexpr 函数	87

5.10 作用域	88
5.10.1 局部作用域	88
5.10.2 文件作用域与全局作用域运算符	90
5.10.3 函数原型作用域	90
5.10.4 函数作用域	91
5.11 程序运行期存储区域	91
5.12 存储类	91
5.12.1 static 变量与多文件项目	92
5.12.2 extern 变量	94
5.12.3 thread_local 变量	95
5.12.4 存储类小结	96
5.13 编译预处理	96
5.13.1 包含文件	97
5.13.2 无参宏	98
5.13.3 有参宏	100
5.13.4 条件编译	103
5.13.5 条件编译示例	104
5.13.6 其他预处理指令	105
小结	106
练习题	107
第 6 章 数组与字符串	111
6.1 一维数组	111
6.1.1 一维数组的定义	111
6.1.2 一维数组的初始化	112
6.1.3 一维数组的访问	113
6.1.4 基于范围 for 语句	114
6.1.5 一维数组的应用	115
6.1.6 调用标准算法简化数组编程	119
6.2 二维数组	121
6.2.1 二维数组的定义	121
6.2.2 二维数组的初始化	121
6.2.3 二维数组的应用	122
6.3 数组与函数	124
6.4 容器 vector 与 map	127
6.4.1 vector	127
6.4.2 map	129
6.4.3 初始化列表与统一初始化	131
6.5 字符数组与字符串	132

6.5.1	字符数组的定义	132
6.5.2	字符数组的初始化	133
6.5.3	字符数组的输入输出	134
6.5.4	字符数组的操作	136
6.6	字符串处理函数	137
6.6.1	字符数组处理函数	137
6.6.2	string 类型	139
6.6.3	字符串应用示例	140
	小结	142
	练习题	142
第7章	结构、枚举、联合体	146
7.1	结构	146
7.1.1	结构类型的定义	146
7.1.2	说明结构变量	149
7.1.3	结构变量的初始化	149
7.1.4	结构变量的使用	150
7.1.5	结构的数组	152
7.1.6	结构中的静态成员	155
7.1.7	结构的嵌套定义	157
7.1.8	C++ 结构的构造函数与成员函数	157
7.2	位域	159
7.2.1	位域的定义	159
7.2.2	位域的使用	161
7.3	枚举	162
7.3.1	枚举类型及枚举变量	162
7.3.2	枚举的使用	163
7.3.3	强类型枚举	165
7.4	联合体	166
7.4.1	联合体类型的定义	166
7.4.2	联合体变量的说明及使用	167
7.4.3	非受限联合体	169
7.5	类型别名 typedef 与 using	170
	小结	171
	练习题	172
第8章	指针和引用	175
8.1	指针及指针变量	175
8.1.1	指针概念与求址运算	175

8.1.2	指针变量的说明与初始化	177
8.1.3	指针的运算	178
8.1.4	用 <code>nullptr</code> 替代 <code>NULL</code>	182
8.2	指针与结构	182
8.2.1	结构的指针	182
8.2.2	指针作为结构成员	184
8.3	指针与数组	186
8.3.1	用指针访问数组	186
8.3.2	指针与字符串	189
8.3.3	指针的数组	191
8.4	指针与函数	198
8.4.1	指针作为形参	198
8.4.2	函数返回指针	199
8.4.3	函数的指针	200
8.5	<code>void</code> 指针与 <code>const</code> 指针	203
8.5.1	<code>void</code> 指针	204
8.5.2	<code>const</code> 指针	205
8.6	动态使用内存	206
8.6.1	<code>new</code> 运算符	206
8.6.2	<code>delete</code> 运算符	208
8.6.3	智能指针与垃圾回收 GC	211
8.7	引用	213
8.7.1	左值引用	213
8.7.2	左值引用与数组、指针的关系	214
8.7.3	左值引用与函数	216
8.7.4	指针与左值引用的对比	220
8.7.5	右值引用 <code>&&</code>	221
8.7.6	引用类型绑定关系	223
8.7.7	<code>auto</code> 推导与 <code>decltype</code> 推导规则	224
8.8	Lambda 表达式	226
8.8.1	语法构造	226
8.8.2	简单用法	228
8.8.3	嵌套 L 式与高阶函数	229
8.8.4	调用 STL 算法	230
	小结	231
	练习题	232
第 9 章	类和对象	239
9.1	类	239

9.1.1	类的定义	239
9.1.2	类成员的可见性	241
9.1.3	类的数据成员	242
9.1.4	类的成员函数	243
9.1.5	类与结构的区别	246
9.2	对象	247
9.2.1	对象的创建	247
9.2.2	访问对象的成员	248
9.2.3	类与对象的关系	249
9.3	this 指针	249
9.4	类中的其他内容	250
	小结	251
	练习题	251
第 10 章	类的成员	253
10.1	构造函数	253
10.1.1	构造函数的定义	253
10.1.2	缺省构造函数	254
10.1.3	委托构造函数	254
10.2	析构函数	255
10.3	拷贝构造函数与拷贝赋值函数	257
10.3.1	拷贝构造函数	257
10.3.2	拷贝赋值函数	259
10.3.3	浅拷贝与深拷贝	261
10.3.4	用 string 替代 char *	263
10.3.5	转换构造函数	263
10.4	移动构造函数与移动赋值函数	265
10.4.1	移动语义	265
10.4.2	移动构造函数	266
10.4.3	移动赋值函数	267
10.4.4	移动实例分析	267
10.5	特殊成员函数及其显式控制	270
10.5.1	特殊成员函数总结	270
10.5.2	特殊成员函数的显式控制	272
10.6	复合对象与成员对象	274
10.6.1	复合类的构造与析构	274
10.6.2	复合对象设计要点	275
10.7	对象数组	276
10.7.1	定义和使用	276

10.7.2 对象数组作为成员	277
10.8 静态成员	277
10.8.1 静态数据成员	277
10.8.2 静态成员函数	279
10.9 限定符	280
10.9.1 限定符 const	280
10.9.2 限定符 volatile	282
10.9.3 引用限定符	283
10.10 类成员的指针	284
10.10.1 数据成员的指针	284
10.10.2 成员函数的指针	286
10.11 线程对象 thread	287
小结	290
练习题	290
第 11 章 类的继承	295
11.1 继承与派生	295
11.1.1 基类与派生类	295
11.1.2 派生类的定义与构成	296
11.1.3 继承方式与访问控制	297
11.2 派生类的构造和析构	299
11.2.1 派生类的构造函数	299
11.2.2 派生类继承构造函数	301
11.2.3 派生类的析构过程	302
11.3 二义性问题与支配规则	303
11.3.1 多继承造成的二义性	303
11.3.2 支配规则	304
11.3.3 导入基类成员	305
11.4 虚基类	306
11.4.1 共同基类造成的二义性	306
11.4.2 虚基类的说明	307
11.4.3 虚基类的例子	309
11.5 子类型关系	310
11.6 虚函数	313
11.6.1 虚函数定义和使用	313
11.6.2 成员函数中调用虚函数	316
11.6.3 构造函数中调用虚函数	317
11.6.4 虚析构函数	318
11.6.5 纯虚函数与抽象类	319

11.6.6 final 函数与类	321
11.7 标量、平凡、标准布局与 POD	322
11.8 字面类型与 constexpr 对象	324
11.9 继承性设计要点	325
小结	327
练习题	327
第 12 章 运算符重载	332
12.1 一般运算符重载	332
12.1.1 运算符重载函数	332
12.1.2 双目运算符的重载	333
12.1.3 单目运算符的重载	334
12.2 友元函数实现运算符	336
12.2.1 友元 friend	336
12.2.2 友元运算符函数	337
12.2.3 用户定义字面值 UDL	339
12.3 特殊运算符重载	341
12.3.1 类型转换函数	341
12.3.2 下标运算符	343
12.3.3 函数调用运算符	344
12.3.4 new/delete 运算符	344
小结	345
练习题	346
第 13 章 模板与 STL	348
13.1 模板的概念	348
13.2 函数模板	349
13.2.1 函数模板的定义	349
13.2.2 函数模板的使用	350
13.2.3 函数模板的显式特例化	353
13.2.4 函数模板与有参宏的区别	354
13.2.5 函数模板重载与 SFINAE 规则	354
13.2.6 模板正确实例化与静态断言	356
13.2.7 带缺省实参的函数模板	357
13.2.8 可变参量的函数模板	359
13.2.9 完美转发与引用折叠规则	360
13.2.10 auto 函数推导返回类型	362
13.3 类模板与别名模板	364
13.3.1 类模板的定义	364

13.3.2	类模板的使用	365
13.3.3	显式特例化与部分特例化	368
13.3.4	友元模板	371
13.3.5	类模板的继承	374
13.3.6	带缺省实参的类模板	376
13.3.7	可变参量的类模板	376
13.3.8	嵌套类模板	377
13.3.9	别名模板	378
13.4	标准模板库 STL	379
13.4.1	容器概念	379
13.4.2	迭代器	381
13.4.3	容器的共同成员类型和操作	382
13.4.4	算法	383
13.4.5	基于 C++11 简化编程	385
13.4.6	函数对象	386
13.4.7	vector, deque 和 list	387
13.4.8	set 和 multiset	390
13.4.9	map 和 multimap	393
13.5	命名空间	398
13.5.1	命名空间的定义	398
13.5.2	空间中成员的访问	399
13.5.3	inline 命名空间	400
	小结	401
	练习题	401
第 14 章	输入输出流	404
14.1	概述	404
14.1.1	流	404
14.1.2	文件	405
14.1.3	缓冲	405
14.2	基本流类	405
14.2.1	基本流类体系	405
14.2.2	预定义标准对象	406
14.2.3	流的格式控制	407
14.2.4	流的错误处理	410
14.3	标准输入/输出	411
14.3.1	cin 输入要点	411
14.3.2	输入操作的成员函数	412
14.3.3	cout 输出要点	414

14.3.4 输出操作的成员函数	415
14.3.5 重载 << 和 >> 运算符	415
14.4 文件流	416
14.4.1 文件概述	416
14.4.2 文件处理的一般过程	416
14.4.3 文件的打开与关闭	417
14.4.4 文本文件的使用	419
14.4.5 二进制文件的使用	422
14.4.6 文件的随机访问	425
小结	427
练习题	427
第 15 章 异常	429
15.1 异常的概念	429
15.2 异常类型的架构	431
15.3 异常处理语句	432
15.3.1 throw 语句	432
15.3.2 try-catch 语句	434
15.3.3 异常处理的例子	436
15.3.4 无异常 noexcept	439
15.4 终止处理器	440
15.5 通用属性	441
小结	442
练习题	442
附录 A ASCII 码表	445
表 A.1 常用 ASCII 码表	445
表 A.2 ASCII 控制字符	446
附录 B 常用库函数	447
表 B.1 运行库的功能分类	447
表 B.2 运行库头文件	448
表 B.3 标准 C++ 头文件	449
表 B.4 string 类型 <string>	450
表 B.5 数学函数 <math.h>	452
表 B.6 C 标准库 <stdlib.h>	453
表 B.7 内存函数 <memory.h>	454
表 B.8 时间函数 <time.h> 与 <sys/timeb.h>	454
参考文献	456

第 1 章 概述

本章介绍 C++ 语言的起源、发展概况及其特点, C++ 程序的基本结构, 面向对象程序设计的基本概念, 简单的上机操作过程。

1.1 C++ 语言发展历史

C++ 语言是在 C 语言的基础上逐步发展和完善的, C 语句吸收了其他高级语言的优点逐步成为实用性很强的语言。

20 世纪 60 年代, Martin Richards 开发了 BCPL 语言(Basic Combined Programming Language)。1970 年, Ken Thompson 在 BCPL 语言的基础上发明了实用的 B 语言。1972 年, 贝尔实验室的 Brian W. Kernighan 和 Dennis M. Ritchie 在 B 语言的基础上进一步对其充实和完善, 设计了 C 语言, 并在 1978 年出版了著名的《C 编程语言》一书, 史称“K&R”。此后 C 语言多次改进, 得到广泛应用。Unix/Linux 操作系统就是基于 C 语言开发的。

C 语言具有以下特点:

(1) 结构化编程语言。以函数作为基本模块, 语法简洁, 使用灵活方便。

(2) 具有一般高级语言的特点, 又具有汇编语言的特点。除了提供对数据进行算术运算、逻辑运算、关系运算之外, 还提供了二进制整数的位运算。用 C 语言开发的应用程序, 不仅结构性较好, 且程序执行效率高。

(3) 可移植性较好。在某一种计算机上用 C 语言开发的应用程序, 源程序经少许更改或不用更改, 可以在其他型号和不同档次的计算机上重新构建运行。

(4) 编程自由度大, 运行错误比较多而且较难解决。指针是 C 语言的灵魂, 精通指针的程序员可以编写非常简洁、高效的程序, 但却不易理解, 而且出错难以排除。初学者掌握 C 语言指针并不容易。

随着 C 语言的不断推广, C 语言存在的一些不足也开始显露出来。例如, 数据类型检查机制比较弱; 缺少代码重用机制; 以函数为模块的编程不能适应大型复杂软件的开发与维护。

1980 年, 贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充, 把 Simula 67 语言中的类引入到 C 中, 称为“带类的 C”。1983—1984 年间 C 语言进一步被扩充, Rick Maseitti 提议将改进后的语言命名为 C++ 语言(称为 C Plus Plus, 这两个加号应书写为上标, 分别表示虚函数和运算符重载)。之后 C++ 语言又扩充了模板、异常等概念, 使功能日趋完善。

C++ 语言除继承了 C 语言的特点外, 还具有以下特点:

(1) C++ 是 C 的一个超集, 它基本上具备了 C 语言的所有功能。一般情况下 C 语言源代码不做修改或略做修改, 就可在 C++ 环境下构建运行。

(2) C++ 是一种面向对象编程语言。面向对象编程的特性是封装性、继承性和多态性。类作为程序的基本模块，封装性隐藏模块内部的实现细节，而使外部使用更方便、更安全；继承性提高了模块的可重用性，而且使程序结构更贴近现实概念的描述；多态性使行为的抽象规范与具体实现相互协调，使行为的一致性和灵活性得到统一。抽象编程和模板提供更好的可重用性，而异常处理则增强了编程的可靠性。这些特征都非常适合大型复杂软件的编程实现。

(3) C++ 语言程序可理解性、可维护性更好。对于大型软件的开发维护而言，这一特点非常重要。

1.2 一个简单的 C++ 程序

开发工具为了区分 C 语言和 C++ 语言程序，约定当源程序文件的扩展名为“.c”时，为 C 语言程序；文件的扩展名为“.cpp”时，则为 C++ 程序。本书中除作特殊说明外，所有源程序文件扩展名均为“.cpp”。

下面通过一个简单的实例，说明 C++ 程序的基本结构及其特点。

例 1.1 根据输入的半径，求出一个圆的面积，并输出计算结果。

```
#include <iostream>
using namespace std;
int main(void) {
    float r, area; //说明两个变量:半径 r 和圆面积 area
    cout << "输入半径 r = "; //显示提示符
    cin >> r; //从键盘上输入半径变量 r 的值
    area = 3.1415926 * r * r; //计算圆面积
    /* 输出半径和圆面积 */
    cout << "半径 = " << r << '\n'; //输出变量 r 的值
    cout << "圆面积 = " << area << '\n'; //输出圆面积的值
    system("pause"); //让控制台暂停, 让用户能看到最后的输出
    return 0; //main 返回, 停止程序
}
```

使用 C++ 集成环境，先创建一个空项目，然后添加一个源文件，再将以上内容输入到源文件中。“生成解决方案”，构建可执行程序，成功之后再“开始执行”（不调试，Ctrl + F5）该程序。

该程序在执行时出现一个 DOS 命令窗口，并显示提示信息，假设键盘输入 3.5，显示结果如下：

```
输入半径 r = 3.5
半径 = 3.5
圆面积 = 38.4845
```

下面对程序的基本结构及各语句进行说明。

(1) 包含文件

第 1 行是 #include，称为包含指令，指定一个文件，<iostream> 是标准输入输出流文件。如果要从键盘上输入数据 cin，或将要在显示器上输出结果 cout，就应包含该文件。一般程序都要这个指令。有关编译预处理将在第 5 章介绍。第 2 行的 using 指令用于导入 std 命名空间，以简化 cin，cout 使用。前 2 行对于大多数程序都一样。

(2) 注释

注释是一段文本信息，用来说明程序功能或方法。在 C++ 程序任何位置都可插入注释。