

SAMS
**Teach
Yourself**

异步图书
www.epubit.com.cn

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计，**1**小时轻松阅读学习，
24小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务，
通过**实践**提高应用技能，巩固所学知识

C++

入门经典（第6版）

[美] Rogers Cadenhead 著
Jesse Liberty
周进 裴强 译

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS



C++ 入门经典 (第6版)

[美] Rogers Cadenhead 著
Jesse Liberty
周进 裴强 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

C++入门经典 : 第6版 / (美) 罗格斯·卡登海德
(Rogers Cadenhead), (美) 杰西·立波提
(Jesse Liberty) 著; 周进, 裴强译. — 北京: 人民
邮电出版社, 2016. 12
ISBN 978-7-115-44215-4

I. ①C… II. ①罗… ②杰… ③周… ④裴… III. ①
C++语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2016)第296847号

版 权 声 明

Rogers Cadenhead and Jesse Liberty: Sams Teach Yourself C++ in 24 Hours(6th Edition)

ISBN: 0672337460

Copyright © 2016 by Pearson Education, Inc.

Authorized translation from the English languages edition published by Pearson Education, Inc.

All rights reserved.

本书中文简体字版由美国 Pearson 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

-
- ◆ 著 [美] Rogers Cadenhead Jesse Liberty
 - 译 周 进 裴 强
 - 责任编辑 傅道坤
 - 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>
 - 北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
 - 印张: 20.75
 - 字数: 515 千字 2016 年 12 月第 1 版
 - 印数: 1—2 500 册 2016 年 12 月北京第 1 次印刷

著作权合同登记号 图字: 01-2016-2469 号

定价: 45.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

内容提要

本书通过大量短小精悍的程序详细阐述了基本的 C++ 编程概念，包括 C++ 程序的组成部分、变量、常量、表达式、语句、运算符、函数、循环、数组、指针、引用、函数和运算符重载、继承、C++14 新增功能、面向对象分析和设计、模板、错误和异常处理等主题。每章都提供了示例程序清单，并辅以示例输出和代码分析，以阐述该章介绍的主题。为加深读者对所学内容的理解，每章末尾都提供了常见问题及其答案以及练习和测验。

本书是针对 C++ 初学者编写的，不要求读者有 C 语言方面的背景知识。本书既可作为高等院校教授 C++ 课程的教材，也可供初学者自学 C++ 时使用。

作者简介

Rogers Cadenhead 是一位作者、计算机程序员和 Web 开发人员，编著过 25 本与互联网相关的图书，包括 *Sams Teach Yourself Java in 21 Days* 和 *Absolute Beginner's Guide to Minecraft Mods Programming*。他运营的 *Drudge Retort* 等网站每年的访问量超过 2200 万人次。本书的官方站点为 <http://cplusplus.cadenhead.org>

Jesse Liberty 编写了大量有关软件开发的图书，包括 C++ 和 .NET 方面的畅销书。他是 Liberty Associates 公司 (<http://www.libertyassociates.com>) 的总裁，负责编程、咨询和培训工作。

前言^①

祝贺您！当您阅读到这里时，离学习最重要的编程语言之一——C++又近了 20 秒。

如果您再花 23 小时 59 分 40 秒，就将掌握 C++ 编程语言的基本知识。只需 24 个课程（每个课程不超过 1 小时），就将学会重要的 C++ 功能，如管理 I/O、创建循环和数组、使用模板进行面向对象编程以及创建 C++ 程序。

我们将这些主题组织成了结构完美、易于理解的课程。在每章中，都将通过项目、输出和代码分析，演示相关的主题。另外，还清楚地标出了语法示例，以方便参考。

每章末尾还列出了常见问题及其答案，帮助您更深入地掌握学到的知识。

本书针对的读者

通过阅读本书来学习 C++ 时，读者不需要有任何编程经验。本书从基本知识开始，既介绍 C++ 语言，又讨论使用 C++ 进行编程涉及的概念。无论读者是刚开始学习编程还是已经有一些编程经验，本书都将让您能够快速而轻松地学习 C++ 语言。

是否要先学 C 语言

不需要先学 C 语言。Bjarne Stroustrup 创立的 C++ 语言是 C 语言的继任者，功能更强大，用途更广泛。先学 C 语言会让您养成一些编程习惯，在您使用 C++ 编程时，这些习惯容易导致错误。本书不要求读者熟悉 C 语言。

^① 本书由周进、裴强两位老师共同翻译，其中周进老师负责前 12 章、附录的翻译工作以及全书的统稿工作，裴强老师负责后 12 章的翻译工作。在此向两位老师表示感谢！

为何要学习 C++

您可学习众多其他的语言，但 C++ 语言最值得学习，因为它经受了时间的考验，当前仍是一种深受欢迎的编程语言。

虽然面世于 1979 年，但鉴于 C++ 语言的强大功能和灵活性，当前仍被用于开发专业软件。另外，最新版本 C++14 让这种语言更有用。

诸如 Java 等其他语言的灵感来自 C++ 语言，学习 C++ 语言也将让您对这些语言有深入认识。掌握 C++ 语言后，便可将这些技能用于当今的任何平台，包括台式机、Linux 服务器、移动设备、视频游戏控制台和大型机。

目 录

第1部分 C++入门

第1章 编写第一个程序	1
1.1 使用 C++	1
1.2 编译和链接源代码	2
1.3 创建您的第一个程序	3
1.4 总结	4
1.5 问与答	4
1.6 作业	4
第2章 程序的组成部分	6
2.1 使用 C++的原因	6
2.1.1 编程风格	7
2.1.2 C++和面向对象编程	8
2.2 程序的组成部分	9
2.2.1 预处理器编译指令	9
2.2.2 源代码行	10
2.3 注释	11
2.4 函数	11
2.5 总结	13
2.6 问与答	14
2.7 作业	14
第3章 创建变量和常量	16
3.1 变量是什么	16
3.1.1 在内存中存储变量	17

3.1.2 无符号变量和带符号变量	18
3.1.3 变量类型	18
3.2 定义变量	19
3.3 给变量赋值	21
3.4 使用类型定义	22
3.5 常量	22
3.5.1 定义常量	23
3.5.2 枚举常量	23
3.6 自动变量	24
3.7 总结	26
3.8 问与答	26
3.9 作业	27
第4章 使用表达式、语句和运算符	28
4.1 语句	28
4.1.1 空白	28
4.1.2 复合语句	29
4.2 表达式	29
4.3 运算符	30
4.3.1 赋值运算符	30
4.3.2 数学运算符	30
4.3.3 组合运算符	31
4.3.4 递增和递减运算符	31
4.3.5 前缀运算符和后缀运算符	31
4.3.6 运算符优先级	33
4.3.7 关系运算符	35
4.4 if-else 条件语句	35

4.4.1	else 子句	36
4.4.2	复合 if 语句	37
4.5	逻辑运算符	38
4.5.1	与运算符	38
4.5.2	或运算符	39
4.5.3	非运算符	39
4.5.4	关系运算符和逻辑运算符的 优先级	39
4.6	棘手的表达式值	39
4.7	总结	40
4.8	问与答	40
4.9	作业	41
第 5 章	调用函数	42
5.1	函数是什么	42
5.2	声明和定义函数	42
5.3	在函数中使用变量	44
5.3.1	局部变量	44
5.3.2	全局变量	46
5.4	函数形参	47
5.5	从函数返回值	47
5.6	默认函数形参	49
5.7	函数重载	50
5.8	自动确定返回类型	51
5.9	总结	53
5.10	问与答	53
5.11	作业	53
第 6 章	控制程序流程	55
6.1	循环	55
6.2	while 循环	55
6.2.1	退出循环	56
6.2.2	跳到下一次循环	57
6.3	do-while 循环	58
6.4	for 循环	59
6.4.1	高级 for 循环	60
6.4.2	嵌套循环	61
6.5	switch 语句	62
6.6	总结	64
6.7	问与答	64
6.8	作业	65

第 7 章	使用数组和字符串存储信息	66
7.1	数组是什么	66
7.2	写入时超过数组末尾	68
7.3	初始化数组	68
7.4	多维数组	69
7.4.1	初始化多维数组	70
7.4.2	内存简介	71
7.5	字符数组	71
7.6	复制字符串	73
7.7	使用 foreach 循环读取数组	74
7.8	总结	75
7.9	问与答	75
7.10	作业	75

第 2 部分 类

第 8 章	创建基本类	77
8.1	类型是什么	77
8.2	创建新类型	77
8.3	类和成员	78
8.3.1	声明类	78
8.3.2	创建对象	79
8.4	访问类成员	79
8.5	实现成员函数	80
8.6	创建和删除对象	82
8.6.1	默认构造函数	83
8.6.2	编译器提供的构造函数	83
8.7	总结	85
8.8	问与答	86
8.9	作业	86
第 9 章	高级类	88
9.1	const 成员函数	88
9.2	接口和实现	89
9.3	组织类声明和函数定义	89
9.4	内联实现	89
9.5	将其他类用作成员数据的类	91
9.6	总结	95
9.7	问与答	95
9.8	作业	95

第3部分 内存管理

第10章 创建指针 97

- 10.1 理解指针及其用途 97
 - 10.1.1 在指针中存储地址 99
 - 10.1.2 间接运算符（间接地使用指针） 100
 - 10.1.3 指针、地址和变量 101
 - 10.1.4 使用指针操作数据 102
 - 10.1.5 查看存储在指针中的地址 103
 - 10.1.6 为何使用指针 104
- 10.2 栈和堆 104
 - 10.2.1 使用关键字 new 105
 - 10.2.2 使用关键字 delete 106
 - 10.2.3 避免内存泄露 107
- 10.3 空指针常量 108
- 10.4 总结 109
- 10.5 问与答 109
- 10.6 作业 109

第11章 开发高级指针 111

- 11.1 在堆中创建对象 111
- 11.2 删除对象 111
- 11.3 使用指针访问数据成员 112
- 11.4 堆中的数据成员 113
- 11.5 this 指针 115
- 11.6 悬摆指针 116
- 11.7 const 指针 116
- 11.8 const 指针和 const 成员函数 117
- 11.9 总结 118
- 11.10 问与答 118
- 11.11 作业 119

第12章 创建引用 120

- 12.1 什么是引用 120
- 12.2 创建引用 120
- 12.3 将地址运算符用于引用 121
- 12.4 可引用的目标 123
- 12.5 空指针和空引用 124
- 12.6 按引用传递函数参数 124

12.6.1 使用指针实现 swap() 125

12.6.2 使用引用实现 swap() 126

- 12.7 理解函数头和原型 127
- 12.8 返回多个值 127
- 12.9 总结 130
- 12.10 问与答 130
- 12.11 作业 130

第13章 高级引用和指针 132

- 13.1 按引用传递以提高效率 132
- 13.2 传递 const 指针 134
- 13.3 作为指针替代品的引用 136
- 13.4 什么情况下使用引用以及什么情况下使用指针 138
- 13.5 指向对象的引用不在作用域内 138
- 13.6 返回指向堆中对象的引用 139
- 13.7 谁拥有指针 141
- 13.8 总结 141
- 13.9 问与答 141
- 13.10 作业 141

第4部分 高级 C++

第14章 高级函数 143

- 14.1 重载成员函数 143
- 14.2 使用默认值 145
- 14.3 初始化对象 146
- 14.4 复制构造函数 147
- 14.5 编译阶段常量表达式 150
- 14.6 总结 151
- 14.7 问与答 152
- 14.8 作业 152

第15章 运算符重载 154

- 15.1 重载运算符 154
 - 15.1.1 编写递增方法 155
 - 15.1.2 重载后缀运算符 157
 - 15.1.3 重载加法运算符 158
 - 15.1.4 对运算符重载的限制 160
 - 15.1.5 赋值运算符 160

15.2	转换运算符	162
15.3	总结	165
15.4	问与答	165
15.5	作业	165

第5部分 继承和多态

第16章 使用继承扩展类

16.1	什么是继承	167
16.1.1	继承和派生	168
16.1.2	动物和继承	168
16.1.3	派生语法	168
16.2	私有和保护	170
16.3	构造函数和析构函数	171
16.4	将参数传递给基类构造函数	173
16.5	重写函数	177
16.5.1	重载和重写	178
16.5.2	隐藏基类的成员函数	178
16.5.3	调用基类的成员函数	180
16.6	总结	181
16.7	问与答	181
16.8	作业	181

第17章 使用多态和派生类

17.1	使用虚成员函数实现多态	183
17.2	虚成员函数的工作原理	186
17.2.1	不能通过基类指针访问派生类特有的方法	187
17.2.2	切除	188
17.2.3	虚析构函数	189
17.2.4	虚复制构造函数	190
17.2.5	使用虚成员函数的代价	192
17.3	总结	193
17.4	问与答	193
17.5	作业	193

第18章 使用高级多态

18.1	单继承存在的问题	195
18.2	抽象数据类型	198
18.2.1	纯虚函数	201
18.2.2	实现纯虚函数	202
18.2.3	复杂的抽象层次结构	204

18.2.4	哪些类是抽象的	207
--------	---------	-----

18.3	总结	208
18.4	问与答	208
18.5	作业	208

第6部分 特殊主题

第19章 使用链表存储信息

19.1	链表和其他结构	210
19.2	链表案例研究	211
19.2.1	委托	211
19.2.2	链表的组成部分	211
19.3	作为对象的链表	217
19.4	总结	218
19.5	问与答	218
19.6	作业	219

第20章 使用特殊的类、函数和指针

20.1	静态成员数据	220
20.2	静态成员函数	222
20.3	将其他类对象作为成员	223
20.3.1	访问被包含类的成员	228
20.3.2	禁止访问内部类的成员	228
20.3.3	按引用还是按值复制	229
20.4	友元类和友元函数	229
20.5	函数指针	229
20.5.1	函数指针数组	232
20.5.2	将函数指针传递给其他函数	234
20.5.3	将 typedef 用于函数指针	236
20.5.4	成员函数指针	236
20.5.5	成员函数指针数组	238
20.6	总结	240
20.7	问与答	240
20.8	作业	240

第21章 使用 C++14 新增的功能

21.1	最新的 C++ 版本	242
21.2	使用 auto 来指定函数的返回类型	242
21.3	改进的数值字面量	245

第1章

编写第一个程序

本章介绍如下内容：

- C++是如何发明的以及发明它的原因；
- 如何寻找 C++编译器；
- 如何创建并编译第一个程序；
- 如何链接并运行程序。

1.1 使用 C++

1979 年，美国贝尔实验室的一位丹麦计算机科学家开始着手改进 C 编程语言。Bjarne Stroustrup 在其个人网站上解释说，他希望有一种可以高效而优雅地编写程序的语言。

这也是很多人的愿望。

Stroustrup 将其作品取名为 C++，几十年来，它一直占据着世界顶级编程语言的宝座。多年来，很多编程语言如流星般闪过，但对台式机、服务器、手机等嵌入式设备以及众多其他计算环境的软件开发而言，C++始终是不错的选择。

C++是一种可移植的语言，适用于 Microsoft Windows、Apple Mac OS、Linux 和 UNIX 系统。要学习这种语言，最佳的方式是编写程序，而不考虑运行程序的操作系统。

本书从实用的角度介绍 C++，没有对使用的操作系统做任何假设。之所以能够做到这一点，是因为本书介绍的是标准 C++（也被称为 ANSI/ISO C++），这是国际上达成一致的版本，可移植到任何平台和开发环境。

本书的所有代码都是标准 ANSI/ISO C++，可在任何遵循最新 C++标准的开发环境中运行。

另外，本书也介绍了最新版（C++14）的新功能。该版本是 2014 年发布的，因此其名称中包含 14；流行的 C++开发环境都支持这个最新版本。

C++程序是使用一组协同工作的工具开发的，这些工具称为编译器和链接器。

编译器将 C++ 转换成能够运行的形式，它将程序从适合人类阅读的形式（源代码）转换为机器能够运行的形式（机器代码）。编辑器生成目标文件，链接器则根据目标文件生成可执行文件。

有多种 C++ 编程环境深受欢迎，您以前可能使用过或知道如何获取这些编程环境，这包括 GCC（GNU 编译器）、Microsoft Visual Studio、NetBeans 和 Embarcadero C++。

只要在计算机上安装了最新的 C++ 编译器并知道其基本用法，您就能毫无困难地完成本书的编程项目。

如果您没有 C++ 编译器，不知道如何使用编译器或不知道如何寻找编译器，也不用担心，附录 D 介绍了如何下载并安装 GCC——一款卓越的免费编译器。如果您还没有安装 C++ 编译器，现在就请按该附录说的做。

Microsoft Visual Studio 也支持 C++ 编程，您可从网站 www.visualstudio.com 下载其免费版 Visual Studio Community。为方便使用 Visual Studio 学习 C++ 的读者，本书提供了一些相关的指南。

1.2 编译和链接源代码

创建您的第一个 C++ 程序之前，有必要了解整个创建过程。

C++ 程序一开始为源代码，即在 Windows “写字板”、Gedit、Emacs 或 Vi 等编辑器中输入的文本。虽然 Microsoft Word 和其他字处理器能够将文件存储为纯文本，但编程时应使用更简单的编辑器，因为您不需要字处理器的任何格式和显示功能。源代码由没有特殊格式的纯文本组成。

**By the
Way**

注意： NotePad++ 是一款免费的 Windows 文本编辑器，提供了卓越的 C++ 编程支持。拥有一定的程序创建经验后，您将喜欢它提供的一些功能，如代码高亮显示，即以不同的颜色显示不同的语言元素。这个文本编辑器可从 <https://notepad-plus-plus.org> 下载。

对于您创建的 C++ 源代码文件，可使用扩展名 .cpp、.cxx、.cp 或 .c。在本书中，所有源代码文件都使用扩展名 .cpp，这是 C++ 程序员最常见的做法，也是有些编译器的默认做法。大多数 C++ 都不关心源代码文件的扩展名，但使用 .cpp 有助于您识别源代码文件。

源代码是供人类阅读的 C++ 程序，必须经过编译和链接才能运行。

编译源代码时，将生成一个目标文件，链接器将把它转换为可执行的程序。

创建 C++ 程序时，将链接一个或多个目标文件以及一个或多个库。库是一系列可链接的文件，提供了有用的函数和类，可供您在程序中使用。函数是一个执行任务的代码块，如将两个数相乘或显示文本。类定义了一种新数据类型和相关的函数。

创建 C++ 程序的步骤如下。

(1) 使用文本编辑器创建源代码。

- (2) 使用编译器将源代码转换为目标文件。
- (3) 使用链接器链接目标文件和必要的库，生成可执行的程序。
- (4) 输入可执行文件的名称以运行它。

GCC 编译器将编译和链接合而为一。

1.3 创建您的第一个程序

介绍完程序的创建过程后，该创建您的第一个程序并尝试使用编译器了。

运行您用于创建程序的文本编辑器并新建一个文件。您将创建的第一个程序会在屏幕上显示文本。

在编辑器中输入程序清单 1.1 所示的文本，但不要包含左边的行号和它后面的冒号。这些行号旨在方便引用代码。

输入时，务必正确地输入标点，如第 5 行的字符::和<<。

输入完毕后，将文件保存为 `Motto.cpp`。

程序清单 1.1 `Motto.cpp` 的完整代码

```
1: #include <iostream>
2:
3: int main()
4: {
5:     std::cout << "Solidum petit in profundis!\n";
6:     return 0;
7: }
```

这里的目标是熟悉创建 C++ 程序的步骤。如果您不知道每行代码的功能，也不用担心，第 2 章将介绍这些代码。

保存文件后，需要进行编译和链接。如果您使用的是 GCC，可使用如下命令完成这两项任务：

```
g++ Motto.cpp -o Motto.exe
```

这个命令让编译器 G++ 编译文件 `Motto.cpp`，并将其链接为可执行文件 `Motto.exe`。如果编辑成功，将不会显示任何消息。该编辑器仅在有问题时出声：显示一条错误消息以及错误出现在哪行。

如果发生编译器错误，请逐行检查程序，确保没有遗漏任何标点，尤其是第 5 行和第 6 行末尾的分号。

解决潜在的问题后，尝试再次编译。如果仍有问题且找不出原因，可从本书配套网站 (<http://cplusplus.cadenhead.org>) 下载该程序。

成功编译程序后，就可在计算机上运行 `Motto.exe`，方法与运行其他程序一样：将其名称 `Motto.exe` 作为命令输入，再按回车键。

程序 `Motto` 的输出如下：

```
Solidum petit in profundis!
```

这是奥尔胡斯大学（Aarhus University）的校训。该大学位于丹麦的奥尔胡斯市，是一座公立大学，在校学生 3.8 万人，在丹麦排名第二。该校训是 Seek a firm footing in the depths 的拉丁版。

奥尔胡斯大学的校友包括丹麦女王玛格丽特二世、诺贝尔化学奖得主 Jens Christian Skou、丹麦王子 Fredrik，还有一个名为 Bjarne Stroustrup 的家伙。

1.4 总结

祝贺您现在可将自己称为 C++ 程序员了，但如果您到此止步不前，那么没有人会认为您是那种值得炫耀的 C++ 程序员。

30 多年来，C++ 一直是流行的软件开发语言。它有其独特之处，但熟悉程序的结构后，将很容易创建复杂的程序。

在接下来的几章中，您将学习 C++ 的基本成分。在每章中，您都将创建多个程序，它们演示了 C++ 语言和编程技巧。

1.5 问与答

问：文本编辑器和字处理器有何不同？

答：文本编辑器生成纯文本文件，只包含字母、数字、空格和标点。它没有设置格式的命令，如粗体、斜体、行对齐、边距等。C++ 源代码不需要这些格式，如果使用字处理器，它可能在文件中存储编译器无法识别的东西。如果您无法编译程序 `Motto` 且使用的是字处理器，请尝试使用更简单的编辑器，如 Windows “写字板”或 Notepad++，看看能否解决问题。

问：我使用的编译器内置了编辑器，可使用它吗？

答：听起来您使用的是集成开发环境（IDE）——一种提高程序编写、调试和测试速度的图形界面工具。诸如 Microsoft Visual Studio 等复杂编译器包含完整的 IDE，让程序员能够访问帮助文件、就地编辑和编译代码以及在不离开 IDE 的情况下解决编译和链接错误。这是一种更好的 C++ 程序编写方式，但仅当您知道如何使用 IDE 时才如此。在学习 C++ 的同时，还要学习 IDE 的方方面面太难了。这就是本书推荐使用 GCC 的原因，它简单、功能强大，还是免费的。

问：能否不理睬编译器发出的警告消息？

答：绝对不能。当编译器发现代码的功能可能并非您的本意时，将发出警告。最好留意这些警告，并通过必要的修改消除它们。出现错误消息意味着编译器不知道如何将您编写的代码转换为机器语言，而警告意味着可以转换，但转换方式可能与您期望的不同。

1.6 作业

输入、编译、链接并执行您的第一个程序后，您将能够回答几个问题并完成两个练习，

以巩固编译器方面的知识。

1.6.1 测验

1. 哪种工具将 C++源代码转换为目标代码？

- A. 编译器
- B. 链接器
- C. 集成开发环境

2. 最常用的源代码文件扩展名是什么？

- A. .cpp
- B. .c
- C. .h

3. 可使用哪些工具来编辑源代码？

- A. 文本编辑器
- B. 字处理器
- C. 两者都行

1.6.2 答案

1. A。编译器接受 C++源代码文件，并将其转换为目标代码。链接器将该目标文件和其他必要的目标文件链接起来，创建一个可执行程序。
2. A。编译器可处理任何源代码文件，而不管其扩展名是什么，但.cpp 被广泛用作 C++代码的文件扩展名。使用这种扩展名让您以后能更容易识别程序的源代码。
3. C。可使用任何可将代码存储为纯文本的工具。您可使用操作系统自带的简单编辑器，如“写字板”、Vi、Gedit 或 Emacs，还可使用其他编辑器，如前面提及的 Notepad++。

1.6.3 练习

1. 修改程序 Motto，使其显示文本“Saluton Mondo!”，这是问候语“Hello world!”的世界语版本。
2. 如果您没有 C++ IDE 且不习惯使用命令行，请看看 NotePad++、NetBeans (<http://netbeans.org>) 或 Code::Blocks (<http://codeblocks.org>)。它们都是免费的 IDE，可进行配置使其与 GCC 协同工作。阅读本书时，您可能发现它们使用起来更容易。这些练习的答案请参阅本书配套网站，其网址为 <http://cplusplus.cadenhead.org>。