

前言 | PREFACE

C++语言作为编程语言中非常受欢迎的语言，具有 C 语言操作底层的能力，同时还具有提高代码复用率的面向对象编程技术，是一种语句更加灵活、使用更加简捷、技术更加全面的编程“利器”。目前学习和关注 C++的人越来越多，而很多 C++的初学者都苦于找不到一本通俗易懂、容易入门和案例实用的参考书。通过本书的案例实训，初学者可以很快地上手流行的工具，提高职业化能力。

本书特色

由浅入深，编排合理：知识点由浅入深，结合流行有趣的热点案例，涵盖了所有 C++程序开发的基础知识，循序渐进地讲解了 C++程序开发技术。

扫码学习，视频精讲：为了让初学者快速入门并提高技能，本书提供了微视频。通过扫码，可以快速观看视频操作，微视频就像一名贴心教师，解决读者学习中的困惑。

项目实战，检验技能：为了更好地帮助读者检验学习的效果，每章都提供了实战训练。读者可以边学习，边进行实战项目训练，强化实战开发能力。通过实战训练的二维码，读者可以查看训练任务的解题思路和案例源码，从而提升开发技能和编程思维。

提示技巧，积累经验：本书对读者在学习过程中可能会遇到的疑难问题以“大牛提醒”的形式进行说明，辅助读者轻松掌握相关知识，规避编程陷阱，从而让读者在自学的过程中少走弯路。

超值资源，海量赠送：本书还随书赠送大量超值资源，包括精美幻灯片、案例源代码、教学大纲、求职资源库、面试资源库、笔试题库和“小白”项目实战手册。



精美幻灯片



案例源代码



教学大纲



求职资源库



面试资源库



笔试题库



“小白”项目实战手册

名师指导，学习无忧：读者在自学的过程中如果有问题，可以观看本书同步教学微视频。此外，本书设有技术支持 QQ 群（912560309），欢迎读者到 QQ 群获取本书的赠送资源并交流技术。

读者对象

本书是一种完整介绍 C++ 程序开发技术的教程，内容丰富、条理清晰、实用性强，适合以下读者学习使用。

- 零基础的编程自学者。
- 希望快速、全面掌握 C++ 程序开发的人员。
- 高等院校相关专业的教师和学生。
- 相关培训机构的教师和学生。
- 初、中级 C++ 程序开发人员。

鸣谢

本书由云尚科技团队策划并组织编写，主要编写人员为王秀英和刘玉萍。本书的编写虽然倾注了众多编者的努力，但由于编写水平有限，书中难免有疏漏和不足之处，敬请广大读者谅解并予以指正。

编 者

目 录 | CONTENTS

第 1 章 快速步入 C++的世界	001
1.1 C++语言概述	001
1.1.1 C语言与C++语言的关系	001
1.1.2 C++语言的特点	001
1.1.3 C++的发展历程	002
1.2 搭建 C++开发环境	002
1.2.1 安装Visual Studio 2019	002
1.2.2 启动Visual Studio 2019	004
1.2.3 使用Visual Studio 2019建立C++程序	005
1.3 C++的编译过程	009
1.4 新手疑难问题解答	010
1.5 实战训练	010
第 2 章 C++语言基础	011
2.1 C++基本语法	011
2.1.1 C++中的基本概念	011
2.1.2 C++中的分号和空格	011
2.1.3 C++中的语句块	012
2.1.4 C++中的标识符	012
2.1.5 C++中的关键字	013
2.2 C++程序的结构	013
2.2.1 第一个C++程序	013
2.2.2 #include指令	014
2.2.3 iostream标准库	015
2.2.4 命名空间	016
2.2.5 函数main()	016
2.2.6 关于注释	017
2.3 C++数据类型	017
2.3.1 整型数据类型	017
2.3.2 浮点型数据类型	019

2.3.3 字符型数据类型	019
2.3.4 布尔型数据类型	021
2.3.5 自定义数据类型	021
2.4 数据的输入与输出	022
2.4.1 认识控制台	022
2.4.2 C++语言中的流	023
2.4.3 认识cout与cin语句	024
2.4.4 流输出格式的控制	026
2.5 新手疑难问题解答	029
2.6 实战训练	030
第3章 使用常量和变量	031
3.1 使用常量	031
3.1.1 认识常量	031
3.1.2 整型常量	032
3.1.3 实型常量	032
3.1.4 字符常量	033
3.1.5 字符串常量	034
3.1.6 其他常量	035
3.2 自定义常量	035
3.2.1 使用#define预处理器	036
3.2.2 使用const关键字	037
3.3 使用变量	038
3.3.1 认识变量	038
3.3.2 变量的声明	039
3.3.3 变量的赋值	040
3.3.4 变量的作用域	041
3.3.5 整型变量	044
3.3.6 实型变量	045
3.3.7 字符型变量	046
3.3.8 布尔型变量	047
3.4 新手疑难问题解答	047
3.5 实战训练	048
第4章 使用运算符和表达式	049
4.1 认识运算符	049
4.1.1 算术运算符	049
4.1.2 自增、自减运算符	050
4.1.3 关系运算符	051
4.1.4 逻辑运算符	052

4.1.5	赋值运算符	054
4.1.6	位运算符	055
4.1.7	杂项运算符	056
4.1.8	逗号运算符	057
4.2	优先级与结合性	058
4.2.1	运算符优先级	058
4.2.2	运算符结合性	059
4.3	使用表达式	061
4.3.1	算术表达式	061
4.3.2	赋值表达式	063
4.3.3	关系表达式	064
4.3.4	逻辑表达式	065
4.3.5	位运算表达式	066
4.3.6	条件表达式	067
4.3.7	逗号表达式	068
4.4	表达式中的类型转换	068
4.4.1	自动转换	068
4.4.2	强制转换	070
4.5	新手疑难问题解答	071
4.6	实战训练	071
第 5 章	程序流程控制结构	072
5.1	顺序结构	072
5.2	选择结构	073
5.2.1	if 语句	073
5.2.2	if...else 语句	074
5.2.3	嵌套 if...else 语句	075
5.2.4	switch 语句	078
5.2.5	嵌套 switch 语句	080
5.3	循环结构	081
5.3.1	循环结构类型	081
5.3.2	循环控制语句	091
5.4	新手疑难问题解答	096
5.5	实战训练	097
第 6 章	函数与函数调用	098
6.1	函数的概述	098
6.1.1	函数的概念	098
6.1.2	函数的定义	098
6.1.3	函数的声明	100

6.2 函数参数及返回值	101
6.2.1 空函数	101
6.2.2 形参与实参	102
6.2.3 函数的默认参数	102
6.2.4 参数的传递方式	103
6.2.5 声明返回值类型	105
6.2.6 函数的返回值	107
6.3 函数的调用	109
6.3.1 函数调用的形式	109
6.3.2 函数调用的方式	109
6.3.3 函数的传值调用	111
6.3.4 函数的嵌套调用	111
6.3.5 函数的递归调用	113
6.4 变量的作用域	116
6.4.1 自动变量	116
6.4.2 静态局部变量	116
6.4.3 外部变量	117
6.4.4 寄存器变量	118
6.5 内联函数	118
6.6 新手疑难问题解答	119
6.7 实战训练	120
第7章 数值数组与字符数组	121
7.1 数组概述	121
7.1.1 认识数组	121
7.1.2 数组的特点	122
7.2 一维数组	122
7.2.1 定义一维数组	122
7.2.2 初始化一维数组	123
7.2.3 一维数组的应用	125
7.3 二维数组	127
7.3.1 定义二维数组	127
7.3.2 初始化二维数组	128
7.3.3 二维数组的应用	129
7.4 多维数组	131
7.5 字符数组	131
7.5.1 字符数组的定义	131
7.5.2 初始化字符数组	132
7.5.3 字符数组的应用	133

7.5.4	字符数组的输出	134
7.5.5	字符数组的输入	135
7.6	新手疑难问题解答	137
7.7	实战训练	138
第 8 章	C++中的指针和引用	139
8.1	指针与变量	139
8.1.1	指针变量的定义	139
8.1.2	指针变量的初始化	140
8.1.3	指针变量的引用	142
8.1.4	指针变量的运算	143
8.2	指针与函数	145
8.2.1	指针传送到函数中	145
8.2.2	返回值为指针的函数	146
8.2.3	指向函数的指针	147
8.3	指针与数组	148
8.3.1	数组元素的指针	149
8.3.2	通过指针引用数组元素	149
8.3.3	指向数组的指针变量作为函数参数	152
8.3.4	通过指针对多维数组进行引用	156
8.4	指针与字符串	160
8.4.1	指向字符串的指针变量	160
8.4.2	使用字符指针作为函数参数	161
8.5	指针数组和多重指针	163
8.5.1	指针数组	163
8.5.2	指向指针的指针	164
8.6	C++中的引用	166
8.6.1	认识C++中的引用	166
8.6.2	通过引用传递函数参数	166
8.6.3	把引用作为返回值	167
8.7	新手疑难问题解答	168
8.8	实战训练	168
第 9 章	结构体、共用体和枚举	170
9.1	结构体概述	170
9.1.1	结构体的概念	170
9.1.2	结构体类型的定义	171
9.1.3	结构体变量的定义	172
9.1.4	结构体变量的初始化	173
9.1.5	结构体变量成员的引用	174

9.2 结构体数组	175
9.2.1 结构体数组的定义	175
9.2.2 结构体数组的初始化	175
9.2.3 结构体数组的引用	176
9.3 结构体与函数	177
9.3.1 结构体变量作为函数参数	177
9.3.2 结构体变量的成员作为函数参数	178
9.3.3 结构体变量作为函数返回值	179
9.4 结构体与指针	180
9.4.1 指向结构体变量的指针	180
9.4.2 指向结构体数组的指针	182
9.4.3 结构体指针作为函数参数	183
9.5 共用体数据类型	184
9.5.1 共用体类型的声明	184
9.5.2 共用体变量的定义	184
9.5.3 共用体变量的初始化	186
9.5.4 共用体变量的引用	186
9.6 枚举数据类型	187
9.6.1 枚举类型的定义	187
9.6.2 枚举类型变量的定义	188
9.7 新手疑难问题解答	189
9.8 实战训练	190
第 10 章 C++中的类和对象	192
10.1 C++类	192
10.1.1 类的概述与定义	192
10.1.2 类的实现方法	193
10.1.3 类对象的声明	195
10.1.4 类对象的引用	195
10.2 类访问修饰符	198
10.2.1 公有成员	199
10.2.2 私有成员	200
10.2.3 保护成员	201
10.3 构造函数	201
10.3.1 构造函数的定义	202
10.3.2 带参数的构造函数	203
10.3.3 使用参数初始化表	204
10.3.4 构造函数的重载	205
10.3.5 构造函数的默认参数	206

10.3.6	复制构造函数	207
10.4	析构函数	209
10.4.1	认识析构函数	209
10.4.2	析构函数的调用	210
10.5	C++类成员	211
10.5.1	内联成员函数	211
10.5.2	静态类成员	212
10.5.3	常量类成员	214
10.5.4	隐式/显式的this指针	216
10.6	类对象数组	218
10.6.1	类对象数组的调用	218
10.6.2	类对象数组和默认构造函数	219
10.6.3	类对象数组和析构函数	220
10.7	友元	221
10.7.1	友元函数	221
10.7.2	友元类	222
10.8	新手疑难问题解答	223
10.9	实战训练	224
第 11 章	C++中的继承与派生	225
11.1	C++中的继承	225
11.1.1	什么是继承	225
11.1.2	基类与派生类	226
11.1.3	基类中的构造函数	228
11.1.4	继承中的构造顺序	231
11.2	C++继承方式	233
11.2.1	公有继承	233
11.2.2	私有继承	234
11.2.3	保护继承	235
11.3	派生类存取基类成员	237
11.3.1	私有成员的存取	237
11.3.2	继承与静态成员	238
11.4	多重继承	239
11.4.1	声明多继承	240
11.4.2	多继承下的构造函数	240
11.5	新手疑难问题解答	242
11.6	实战训练	242

第 12 章 C++中的多态与重载	244
12.1 多态概述	244
12.1.1 认识多态行为	244
12.1.2 实现多态性	244
12.2 虚函数与虚函数表	246
12.2.1 虚函数的作用	246
12.2.2 动态绑定和静态绑定	248
12.2.3 定义纯虚函数	249
12.2.4 认识虚函数表	251
12.3 抽象类与多重继承	252
12.3.1 抽象类的作用	253
12.3.2 抽象类的多重继承	253
12.4 认识运算符的重载	254
12.4.1 什么是运算符重载	255
12.4.2 运算符重载的形式	255
12.4.3 可重载的运算符	258
12.5 常用运算符的重载	259
12.5.1 “<” 运算符重载	259
12.5.2 “+” 运算符重载	260
12.5.3 “=” 赋值运算符重载	261
12.5.4 前置运算符重载	263
12.5.5 后置运算符重载	264
12.5.6 插入运算符重载	265
12.5.7 折取运算符重载	266
12.6 新手疑难问题解答	268
12.7 实战训练	268
第 13 章 C++中模板的应用	269
13.1 函数模板	269
13.1.1 函数模板的用途	269
13.1.2 函数模板的定义	270
13.1.3 函数模板的调用	271
13.1.4 函数模板的重载	273
13.2 类模板	274
13.2.1 类模板的定义	274
13.2.2 类模板的实例化	277
13.2.3 类模板的使用	277
13.3 模板的特化	279
13.3.1 函数模板的特化	280

13.3.2 类模板的特化	281
13.4 新手疑难问题解答	282
13.5 实战训练	283
第 14 章 容器、算法与迭代器	284
14.1 认识容器	284
14.2 顺序容器	285
14.2.1 向量类模板	285
14.2.2 链表类模板	290
14.2.3 双端队列类模板	296
14.3 关联容器	297
14.3.1 映射类模板	298
14.3.2 集合类模板	301
14.3.3 多重集合类模板	302
14.4 容器适配器	303
14.4.1 栈类	303
14.4.2 队列类	304
14.4.3 优先级队列类	305
14.5 C++中的算法	307
14.5.1 数据编辑算法	307
14.5.2 查找算法	308
14.5.3 比较算法	310
14.5.4 排序相关算法	311
14.6 C++中的迭代器	312
14.6.1 迭代器的分类	312
14.6.2 迭代器的使用	314
14.7 新手疑难问题解答	315
14.8 实战训练	315
第 15 章 C++程序的异常处理	316
15.1 认识异常处理	316
15.1.1 认识异常处理机制	316
15.1.2 认识标准异常	317
15.1.3 异常处理语句块	317
15.2 异常处理的简单应用	318
15.2.1 抛出异常	318
15.2.2 重新抛出异常	320
15.2.3 捕获所有异常	321
15.2.4 异常的匹配	321

15.3 异常处理的高级应用·····	324
15.3.1 自定义异常类·····	324
15.3.2 捕获多个异常·····	325
15.3.3 异常的重新捕获·····	327
15.3.4 构造函数的异常处理·····	328
15.4 新手疑难问题解答·····	329
15.5 实战训练·····	330
第 16 章 C++中文件的操作·····	331
16.1 文件 I/O 操作·····	331
16.1.1 输入文件流·····	331
16.1.2 输出文件流·····	332
16.1.3 输入/输出文件流·····	333
16.2 文件的打开与关闭·····	334
16.2.1 文件的打开·····	334
16.2.2 文件的关闭·····	336
16.3 文本文件的处理·····	337
16.3.1 将变量写入文本文件·····	337
16.3.2 将变量写入文件尾部·····	337
16.3.3 从文本文件中读取变量·····	338
16.4 使用函数处理文本文件·····	339
16.4.1 使用函数get()读取文本文件·····	339
16.4.2 使用函数getline()读取文本文件·····	340
16.4.3 使用函数put()将记录写入文本文件·····	340
16.5 新手疑难问题解答·····	341
16.6 实战训练·····	341

第1章

快速步入 C++的世界



本章内容提要

C++是由 Bjarne Stroustrup 于 1979 年在贝尔实验室开始设计、开发的，是一种面向对象的程序设计语言。C++可运行于多种平台上，如 Windows、Mac 操作系统及 UNIX 操作系统的各种版本。本章介绍 C++语言的基础知识，主要内容包括 C++语言的起源和特色、C++语言开发环境搭建、C++语言的编译过程等。

1.1 C++语言概述



微视频

C++是一种静态类型的、通用的不规则编程语言，支持过程化编程和面向对象编程，它综合了高级语言和低级语言的特点。

1.1.1 C 语言与 C++语言的关系

C++语言最初的开发宗旨是作为 C 语言的继任者，可以说，C++语言进一步扩充和完善了 C 语言。但不同于 C 语言，C++是一种面向对象的语言，实现了继承、抽象、多态和封装等概念。C++语言还支持类，而且类包含成员数据及操作成员数据的成员方法。其结果是，程序员只需要考虑数据及要用它们来做什么。一直以来，很多 C++编译器都支持 C 语言。

☆**大牛提醒**☆

虽然 C++是在 C 语言的基础上发展起来的一种语言，但它不是 C 语言的替代品或 C 语言的升级。C++语言和 C 语言是兄弟关系，故没有谁比谁先进的说法。

1.1.2 C++语言的特点

相比 C 语言而言，C++语言是一种年轻的语言。C 语言曾以其简洁明了的结构化编程成为主流编程语言，当时很多程序员从事 C 语言程序开发。C++以 C 语言为基础，加入面向对象概念，不仅顺应了当时的潮流，还简化了 C 语言程序设计到 C++语言程序设计的转变过程。

如今，C++仍然是一种主流编程语言，这足以证明 C++是一种优秀而又强大的编程语言。其具有以下几个优点。

(1) C++应用范围十分广泛。C++的应用几乎无所不包，从科学计算到网络应用程序、从分布式应用到移动设备应用、从系统级软件到计算机游戏应用都有 C++靓丽的身影。

(2) C++在硬件级编程方面。由于 C++包含了 C 语言特性，因此对于硬件驱动的开发，自然也游刃有余。

(3) C++编程的高效性。相比其他面向对象编程语言，C++的执行效率更高。

(4) C++类库执行。C++的标准库包含了大量模板、通用算法，能够大大提高开发效率。

(5) C++遵循 ANSI 标准。标准的建立不仅仅是所有程序员的福音，也使 C++的进一步发展成为可能。标准化使 C++编写的程序从一台计算机上移植到另一台计算机上成为可能。

1.1.3 C++的发展历程

要想学好 C++编程，了解 C++的历史演变过程是一个必需的前提。而 C++是从 C 语言发展来的，所以首先从 C 语言的历史讲起。

C 语言是由计算机科学家丹尼斯·里奇（Dennis Ritchie）创造的。在 1967 年，丹尼斯·里奇进入著名的贝尔实验室（C 语言、C++语言和 UNIX 操作系统都在此诞生）工作。在贝尔实验室工作的过程中，里奇为了解决在工作中遇到的问题，创造了 C 语言。如图 1-1 所示，即为 C 语言发展历程示意图。

不过，在 1979 年，Bjarne 博士为了分析 UNIX 的内核，苦于当时没有合适的工具将 UNIX 的内核模块化，于是他为 C 加上了一个类似 Simula 的机制，并为此专门成立了开发小组。这就是 C++最初的萌芽状态。当时，这个语言并不是叫作 C++，而是叫作 C with class，它仅仅被当作 C 语言的一种补充。不过，随着事态的发展，C++逐渐成熟起来。

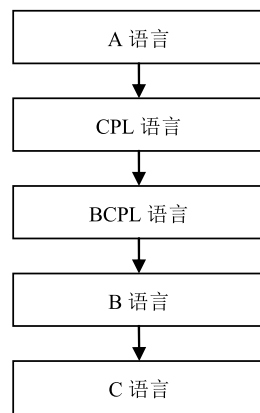


图 1-1 C 语言发展历程



微视频

1.2 搭建 C++开发环境

随着 C++语言的不断发展，C++语言的集成开发环境也有了长足的发展。对于使用 Windows 平台的 C++语言开发人员来讲，使用 Visual Studio (VS) 进行开发比较普遍。所以本书以 Visual Studio 2019 为主进行讲解。

1.2.1 安装 Visual Studio 2019

本小节介绍 Visual Studio 2019 的安装方法，具体操作步骤如下。

步骤 1：下载 Visual Studio 2019 程序安装包，如图 1-2 所示。

步骤 2：双击下载好的安装程序文件，进入安装界面，如图 1-3 所示。

步骤 3：单击“继续”按钮，会弹出“Visual Studio 2019 程序安装加载页”界面，显示正在加载程序所需的组件，如图 1-4 所示。

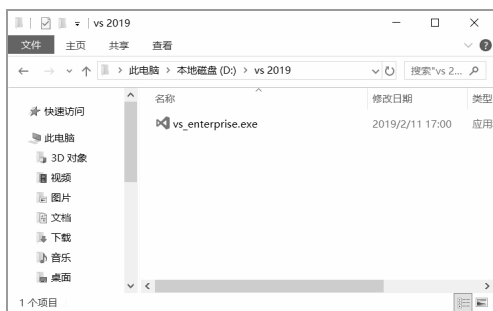


图 1-2 Visual Studio 2019 程序安装包

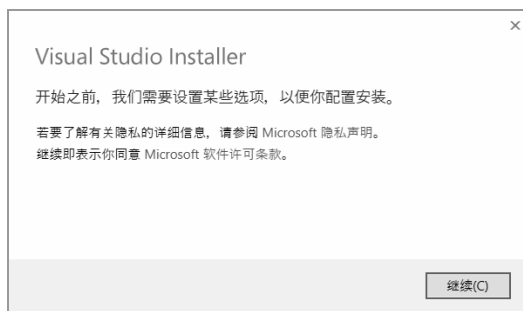


图 1-3 Visual Studio 2019 安装界面

步骤 4: 加载完成后, 会自动跳转到“Visual Studio 2019 程序安装起始页”的界面, 如图 1-5 所示。该界面提示有三个版本可供选择, 分别是 Visual Studio Enterprise 2019 Preview、Visual Studio Professional 2019 Preview、Visual Studio Community 2019 Preview, 用户可以根据自己的需求选择。对于初学者而言, 一般推荐安装 Visual Studio Community 2019 Preview。



图 1-4 安装加载页



图 1-5 安装起始页

步骤 5: 单击“安装”按钮后, 弹出“Visual Studio 2019 程序安装选项页”的界面。在该界面中, 选中“通用 Windows 平台开发”复选框和“使用 C++ 的桌面开发”复选框, 也可以在“位置”处更改程序安装路径, 如图 1-6 所示。



图 1-6 Visual Studio 2019 程序安装选项页

步骤 6: 在图 1-6 中选择好要安装的功能后, 单击“安装”按钮, 进入如图 1-7 所示的“Visual Studio 2019 程序安装进度页”界面, 显示安装进度。安装程序自动执行安装过程, 直至该安装过程执行完毕。