国家自然科学基金，四川大学精品课程建设基金资助项目

C++面向对象程序设计

李　涛 主编

游洪跃　陈良银　李　琳 编

高等教育出版社

内 容 提 要

本书全面系统地介绍了面向对象程序设计的基本原理和要素，详细介绍了 C++程序设计技术，并在此基础上对 VC++的程序设计技术等内容做了重点介绍。本书将 C++的基本原理与具体实践相结合，不仅使读者对 C++语言本身有了深该的了解，也为读者迅速掌握目前流行的 C++程序设计工具打下了良好的基础。

本书取材新颖，内容丰富，可读性强，所设计的示例程序立足于方便各位读者观察 C++各机制的运行过程。本书可作为高等院校计算机、信息技术专业本科生和研究生的教材，亦可供其他从事软件开发工作的读者学习参考使用。

附录　C++编程风格指导

一、匈牙利基本命名法

匈牙利命名法是一名匈牙利程序员发明的，他在 Microsoft 工作了多年，此命名法就是通过 Microsoft 的各种产品和文档被传出来的。多数有经验的程序员，不管他们用的是哪种程序设计语言，都或多或少在使用它。

这种命名法的基本原则是：

变量名＝属性＋类型＋对象描述

即一个变量名由 3 部分信息组成，这样命名的好处是程序员很容易理解变量的类型、用途，而且便于记忆。

下面给出一些推荐使用的规则例子，可以选择使用，也可以根据个人喜好作些修改后再使用。

1．属性部分

全局变量（global）：　g_

类成员变量(member)：　m_

静态变量(static)：　s_

2．类型部分

by：无符号字符/字节(BYTE)
b：布尔型(bool)
w：16 位无符号值(WORD)
dw：32 位无符号整型(DWORD)
i：整型(int) (其大小依赖于操作系统)
l：32 位有符号整型(long)
f：浮点型(float)
d：双精度浮点型(double)
str：字符串对象(cstring)
c：字符型(char)
sz：以零结束的字符串(char[])
p：指针(pointer)
lp：长指针(long pointer)
lpsz：32 位字符串指针（char *）指针：
h：句柄（handle）
n：整数（表示一个数 number）
x, y：整数（通常用于 x 坐标和 y 坐标）
v：空类型（void）
fp：函数指针
msg：消息(message)
a：数据组(array)
示例：int iaWidth[7];
o：对象(object)
s：结构(struct)
u：联合(unio)
t：类型(type)

3．描述部分

初始化：Init
临时变量：Tmp
目的对象：Dst
源对象：Src
窗口：Wnd
例如：
hWnd： h 表示句柄，Wnd 表示窗口，合起来为“窗口句柄”。

m_bFlag： m 表示成员变量，b 表示布尔，合起来为：“某个类的成员变量，布尔型，是一个状态标志”。

二、基本规范参考

1．文件命名规则

（1）对每章的例题(sample)的命名规则

主程序文件名为：smain 章_题号.cpp
　　示范：smain5_11.cpp
　　意义：第 5 章例 11 的主程序文件
类声明文件名为：sclass 章_题号_描述.h
　　示范：sclass5_11_bookl.h
　　意义：第 5 章例 11 类 book 的类声明文件
类实现文件名为：sclass 章_题号_ 描述.cpp
　　示范：sclass5_11_book.cpp
　　意义：第 5 章例 11 类 book 的成员函数定义文件

（2）对每章中习题(exercises)的命名规则

主程序文件名为：emain 章_题号.cpp
　　示范：emain5_11.cpp
　　意义：第 5 章习题 11 的主程序文件
类声明文件名为：eclass 章_题号_描述.h
　　示范：eclass5_11_book.h
　　意义：第 5 章习题 11 类 book 的类声明文件
类实现文件名为：eclass 章_题号_描述.cpp
　　示范：eclass5_11_book.cpp
　　意义：第 5 章习题 11 类 book 的成员函数定义文件

（3）对模拟试题(test questions)的命名规则如下：

主程序文件名为：tmain 模拟试题号_题号.cpp
　　示范：tmain5_6.cpp
　　意义：模拟试题 5 的第 6 题的主程序文件
类声明文件名为：tclass 模拟试题号_题号_描述.h

示范：tclass5_6_book.h

意义：模拟试题 5 类 book 的第 6 题的类声明文件

类实现文件名为：eclass 模拟试题号_题号_描述.cpp

示范：tclass5_11_book.cpp

意义：模拟试题 5 的第 6 题类 book 的成员函数定义文件

（4）对实验(practice)的命名规则

主程序文件名为：pmain 实验号.cpp

示范：pmain3.cpp

意义：实验 3 的主程序文件

类声明文件名为：pclass 实验号_描述.h

示范：pclass3_book.h

意义：实验 3 类 book 的类声明文件

类实现文件名为：pclass 实验号_描述.cpp

示范：pclass3_book.cpp

意义：实验 3 类 book 的成员函数定义文件

注意：大文件名中描述部分如有多个单词，不同单词之间用下划线分隔。

2．类的名称

类的命名规则为: 以大写字母 C 开头, 后跟类名的首字母要大写, 且类名中每个单词的首字母都要大写, 例如类名: CState

3．头文件

编译一个文件的过程中常会多次包含同一头文件，因此, 定义头文件的结构如下:

对于头文件 s5_6_book.h:

```
#ifndef __ S5_6_BOOK _H__
#define __ S5_6_BOOK _H__
头文件内容 content of header file...
#endif
```

符号常数的名称由前后各两个下划线（'__'）加类的名称再加上'_H' 组成 （如上所示）。类名称中所有字母都应大写。一个类或类库必须提供良好的用户接口, 在不了解该类所引用的其他类或其本身实现细节的情况下仍可使用。VC 将对对每个类的实现文件（如 s5_6_book.cpp）进行 VC 将进行预编译，这就要求在每个类的实现文件文件中要清楚地标识出因被该类引用而所需包含的所有头文件。

一般形式为：

```
#include < *.h >              // 对于系统原有的头文件
```

```
#include " s5_6_book.h"      // 对于用户自定义的头文件
```

头文件的书写顺序为:

（1）文件序言，为一些注释，注释内容见附件 1 格式

（2）所要包含的头文件

（3）符号常数和全局常数的定义

（4）宏定义

（5）类型定义

（6）结构、联合和类的定义

（7）外部变量的宣告

（8）内联函数的定义 （包括成员函数）

注意 ：

（1）一个好的头文件应该对每一相对独立的部分作出注释。

（2）在描述要包含的头文件时，引号中不能出现头文件所在路径。

4．源文件

源文件的结构与 C 语言大致相同。在源文件中，全局常数（即由类型修饰符 const 定义的标识符）的定义与符号常数的定义在同一处。

源文件的书写顺序为:

（1）文件序言，为一些注释行

（2）所要包含的头文件

（3）仅限于在该文件内部使用的符号常数和全局常数的定义

（4）宏定义

（5）文件以外的全局变量的定义

（6）在该文件中定义的静态函数的定义

（7）类的定义

（8）函数定义 （包括成员函数）

5．标识符名

（1）类名

书写形式: C+类名首字母大写+每个单词首字母大写

示　　范: CGState

（2）结构名

书写形式: S+ 其余同上

示　　范: SGstate

（3）联合名

书写形式: U+ 其余同上
示　　范: UGState

（4）类型名

书写形式: T+ 其余同上
示　　范: TGState

6．成员函数名

书写形式: 函数名首字母大写+其后每个单词首字母大写;
注意：
（1）用以一个动词开头的一组词确切地表明函数或方法的功能
（2）合理地使用有意义的前后缀
①常用的后缀
Max —— 最大值
Min —— 最小值
Cnt —— 个数
Key —— 关键值
②常用的前缀
Is —— 提出一个问题
Get —— 取一个值
Set —— 设一个值

7．函数参数名

书写形式: 参数类型 + 表明参数含义的名词词组
其中：参数类型采用匈牙利命名法且全部字母小写
示范: ReadString(CString& strText);
注意：匈牙利命名法并没有对函数参数名的命名规则进行规定，MFC 一般都遵循匈牙利命名法，为养成良好习惯，本书内容也遵循此规则。

8．成员函数内部变量

书写形式: 变量类型 + 表明变量含义的名词词组

其中: 变量类型采用匈牙利命名法且全部字母小写，常用的变量类型可以参见成员变量中的变量类型。表明变量含义的名词词组的命名规则和类命名规则相同

示范: int iCnt;

注意：匈牙利命名法并没对成员函数内部变量的命名规则进行规定，MFC 一般都遵循匈牙利命名法，为养成良好习惯，本书内容也遵循此规则。

9．符号常量、宏和枚举量

（1）符号常量

书写形式: 所有字母皆为大写形式, 且每个单词之间以 '_'连接

示　　范: const int SALES_TAX = 1;

（2）宏和枚举量

书写形式: 所有字母皆为大写形式, 且每个单词之间以 '_'连接

示范 1:

```
#define SALES_TAX 1
```

示范 2:

```
enum PinStateType
{
    PIN_OFF,
    PIN_ON
}
```

注意：所有标识符在命名时都应为有对应意义的英文单词, 避免使用单个字母, 如 i, j...

9．类的定义

1）较长类的定义通常遵照下述形式

文件名: Class.h

```
class Class
{
友元函数 friends

public:
构造函数 constructor
析构函数 destructor
```

```
成员函数 member-function-decls

protected:
成员函数 member-function-decls

private:
成员函数 member-function-decls

protected:
数据成员 data-members

private:
数据成员 data-members
};
内联函数定义 inline    member-function-definitions
```

文件名: Class.cpp

```
#include    "Class.h"
按成员函数声明顺序进行定义 member-function-definitions in order declared
 (除内联函数 except inlines)
```

对于类的定义需要注意以下几点:

①public 在文件的最前面, 因为这部分内容是类的使用者最观注的, 其实现细节出现在文件的稍后部分。

②内联函数的定义在类的构造部分之后, 以区别于类的宣告部分,不致在类的定义中产生混乱。

③关键词 public、 protected 和 private 不能省略。

④在类的构造函数中应该只做一些变量初始化和一些不会失败的动作。

2）较短类的定义通常遵照下述形式

文件名: Class.h

```
class Class
{
友元函数 friends

public:
构造函数 constructor
析构函数 destructor
成员函数 member-function-decls

protected:
成员函数 member-function-decls

private:
成员函数 member-function-decls
```

```
protected:
数据成员 data-members

private:
数据成员 data-members
};
```

按成员函数声明顺序进行定义 member-function-definitions in order declared

11．形式参数的宣告

在实际工程中宣告函数的形式参数时，每一个一般形参应单放一行，便于在必要时对其分别加以注释 （通过单行注释符号“//”）。

例如:

```
function(
        int    arg1,                    // comment for arg1
        char *arg2                  // comment for arg2
    )
```

形式参数的命名要有一定的意义。如果参数名提供了足够的信息,可以考虑免去注释,这时可以维持一行一个参数的形式，也可以不维持为种形式。

示例 ：

```
_Rectangle(
        CDC *pDc,       // pointer of CDC
        int    iLeft,           // the left position of rectangle
        int    iTop,            // the top of rectangle
        int    iRight,      // the right position of rectangle
        int    iBottom     // the bottom of rectangle
    )
```

或

```
_Rectangle(CDC *pDc, int iLeft, int iTop, int iRight, int iBottom)
```

12．注释行

适当的注释可以增强程序的可读性，注释的符号分为两种:

“//”和“/*... */”。

使用注释时应注意:

（1）“//”的注释行不可用于以“#”开头的预处理命令行中。

（2）除去文件与函数始端的注释外，对程序中的每一处做过修改之后都要加“//注释”。

（3）在实际工程中，一般每个程序和外部函数的文件始端都有注释，用来指明:

文件名 file name

权限 proprietary　markings

程序的规格 (specification)

维护的历史 history

程序主要的功能及包含的主要函数或类

程序中已知的限制

程序中存在的 bug

注意：在本书中，一般每个程序和外部函数的文件始端都用注释来指明文件名和程序主要的功能及包含的主要函数或类，对于非常简单的程序和外部函数，在程序和外部函数的文件始端可以只注释文件名。

（4）在实际工程中，函数体前必须有对其进行说明的注释，用来指明:

函数名

函数的输入、输出

函数有可能产生的副作用

函数中已知的限制

有关函数作用的简单描述

错误信息返回值

注意：在本书中，一般函数体都用注释来简单描述有关函数作用的简单描述，对于非常简单的函数也可不加注释。

例如：

```
// 已知三边 a, b, c，求三角形的面积
double AreaOf Triangle (double a, double b, double c)
{
    函数体
}
```

（5）较长语句体的中断处必须由注释指明。

例如:

```
while ......
{
    ...
    ...
    if ...
    {
        ...
        ...
    }    // end if
    ...
    ...
}    // end  while
```

13. 空格

（1）在使用括号时其前后无须加空格, 例如:

```
strcpy(data, ch);
```

（2）在使用作用域分辨符::时其前后无须加空格,例如:

```
Classname::membername
```

（3）在使用冒号:时其前后必须加空格，例如:

```
class Derived : public Base {...}
```

以及

```
Derived::Derived() : Base(1)
Derived::Derived() : Base(1)
```

（4）在使用取地址符号&时，标识符名必须紧跟在该操作符后，并且操作符前必须有一空格，例如:

```
int &x, &y;
```

14．修饰符的位置

修饰符 * 和 & 应该靠近数据类型还是靠近变量名，是个有争议的活题。

若将修饰符 * 靠近数据类型，例如 int* x 从语义上讲此写法比较直观，即 x 是 int 类型的指针。

上述写法的弊端是容易引起误解，例如：int* x, y; 此处 y 容易被误解为指针变量。虽然将 x 和 y 分行定义可以避免误解，但并不是人人都愿意这样做。

本书约定:应当将修饰符*和＆近靠变量名

例如：

```
char    *name;
int     *x, y;      // 此处 y 不会被误解为指针
```

注意：上述约定仅限于在用户自定义变量或函数参数等情况，对于 MFC 自动生成的代码，*靠近类型，本书对 MFC 生成的代码不作统一的格式处理，都保持原型。

15．语法标准

（1）在程序源码中，每行的内容不可超出屏幕，也就是说每行不宜超过 78 个字符。

（2）每一行只写一条语句。

（3）在程序的每个片段结束之后要跳过一行再继续。

16．缩排规则

（1）每一个左括号{和右括号}都必须单独占一行，并且同与它相关联的控制结构（如 if, while, for, switch 等）的边界处对齐。右括号}后一般有一个关于已结束的结构体的注释。

注意：该规则同样适用于函数、结构、联合或类的定义。

例 1:

```
String::String(const char *ch)
{
    if (ch)
    {
        data = new char[strlen(ch) + 1];
```

```
            strcpy(data, ch);
        }
        else
        {
            data = new char[1];
            *data = '\0';
        }       // end if
    }
```

（2）上一条规则的唯一例外是 do-while 结构。在该结构中,右括号}必须与 while (condition) 位于同一行中， 以免有人将 while 误解为 "null loop"。

例 2:

```
do
{
    ...
} while (expr);
```

（3）控制结构内部的语句体要用大括号括起，甚至当控制体内只有一条语句时也要如此。这可以避免需要添加另外的语句时忘记加大括号。

例 3:

```
if (expr)
{
    return(1);
}
```

当且仅当语句体内容为空时此规则可以例外

例 4:

```
while (expr) ;
```

（4）复合表达式、逻辑与／或表达式需要用括号括起来，以强调运算关系。

例 5:

```
if ((m_wNumColors > 256) && (m_hDlgHiColor || TRUE))
{
    return(1);
}
```

（5）标明程序、类定义及函数定义功能的标题注释必须紧靠左边界书写。

（6）类、结构、联合和数组初始化部分的形式应为每行只有一条语句，而且每条语句也要相对于类、结构、联合和数组的宣告处缩进两个格。

（7）控制结构（如 if, while, do-while, for, switch） 中的判断条件处一般应以注释说明 'true' or 'false' 的动作。

17．运算符和关键字

（1）主要的二元运算符在书写时两边必须加空格。（ +, -, *, /, %, =, <, >, <<=, >>=, &&, ||, ==, !=, +=, -=, *=, /=, %=, &=, |=, <=, >=, <<, >>, &, |）

例 1:

```
if (left < right)
```

```
{
    total    = num1 + num2 - num3 * num4 / 4;
    total2 = num1 + num2 - num3 * num4 / 4 + num5 + num6 - num7 * num8 / 4;
}
```

（2）一元运算符和它的运算对象之间不能有空格。

例 2:

```
sum++;
```

（3）赋值运算符两边必须有空格，且对一组相关内容的赋值语句须按如下形式书写

例 3:

```
bills_due                = true;
my_checkbook   = 0;
bills_paid          = false;
```

（4）条件运算符两边必须有一个空格。

（5）逗号或分号后面必须有一个空格 （甚至空行），定义函数形参时除外。

例 4:

```
int    position1, position2;
char bill_name, check_title;
```

（6）所有其它运算符两边都要有一个空格。

关键字（if, while, for, switch, return）后面须跟一个空格。

当位运算符（&, |, ~, ^, >>,<<）与其他运算符共同使用时,应用括号作为特别区分。

例 5:

```
temp = (cont + (temp1 & temp2));
```

成员选择运算符 （-> , . ）两边不能有空格。

例 6:

```
pt.x = LOWORD(pMsg->lParam);
```

18．数字常量和字符常量

对于程序中有特殊含义的数值常量和字符常量等内容，应该用宏或符号常量来代替

例如：

```
const int MAX_BUF_LEN = 255;
...
if ((GetDataLen() <= MAX_BUF_LEN)
...
```

19．程序实例

```
// 文件名:sclass6_8_circle.h

//程序作者:Mr. You
//本书教学支持站点:http://cs.scu.edu.cn/~litao/
//本书教学支持站点:http://cs.scu.edu.cn/~chenliangyin
```

```
#ifndef __SCLASS6_8_CIRCLE_H__
#define __SCLASS6_8_CIRCLE_H__

const int DEFAULT_RADIUS = 10;              // 缺省圆半径

class CCircle
{
public:
    CCircle(double dRadius = DEFAULT_RADIUS);          // 构造函数
    void SetRadius(double dRadius = DEFAULT_RADIUS); // 设置圆半径
    double GetRadius();                        // 返回圆半径
    double GetArea();                          // 返回求圆面积

private:
    double m_dRadius;                          // 圆半径
};

#endif



// 文件名:sclass6_8_circle.cpp

#include "sclass6_8_circle.h"              //圆的类构造头文件

const int PI = 3.1415926;             //圆周率常数

// 已知半径 dRadius 构造圆
CCircle::CCircle(double dRadius)
{
    m_dRadius = dRadius;
}

// 设置圆半径
void CCircle::SetRadius(double dRadius)
{
    m_dRadius = dRadius;
}

// 返回圆半径
double CCircle::GetRadius()
{
    return m_dRadius;
}

// 求圆面积
double CCircle::GetArea()
{
    double dArea;
```

```
    dArea = PI * m_dRadius * m_dRadius;
    return dArea;
}



// 文件名:smain6_8.cpp

#include <iostream.h>
#include <stdlib.h>
#include "sclass6_8_circle.h"      // 圆的类构造头文件

// 已知半径显示圆面积
void main(void)
{
    double dRadius, dArea;
    CCircle *poCircle;
    cout << "请输入圆半径:";
    cin >> dRadius;

    poCircle = new CCircle(dRadius);
    if (poCircle == NULL)
    {
        cout << "内存耗尽！" << endl;
        exit(1);
    }

    dArea = poCircle->GetArea();
    cout << "圆面积是" << dArea << " ." << endl;
    delete poCircle;
}
```

参 考 文 献

1 Stroustrup B, The C++ programming language, Higher Education Press Pearson Education，2002
2 Brian W, Kernighan, Dennis M, Ritchie, The C Programming Language, 1998
3 Dattatri，Kayshav，C++ Effective Object-Oriented Software Construction. Prentice Hall PTR, 1997
4 Koenlg, Andrew and Moo, Barbara E.，Ruminations on C++. AddisonWesley, 1997
5 Koenlg, Andrew and Moo, Barbara E.， Accelerated C++. AddisonWesley 2000
6 Langer, Angellka and Kreft, Klaus, Standard C++ IOStreams and Lippman, Addison Wesley Longmab, 2000
7 Lippman , Stanley B., C++ Gems. SIGS Books & Multimedia，1996
8 Meyers, Scott, Effective C++. Addiso – Wesley, 1992
9 Neyers, Scott, More Effective C++. Addison – Wesley，1996
10 Neyers, Scott, Effective STL．Addison – Wesley，2001
11 Musser, David R., Derge, Gillmer J., Saini，Atul, STL Tutorial and Reference Guide, Second Edition．Addison – Wesley，2001
12 Pohl, Ira, C++ for C Programmers. Addison – Wesley，1999
13 Stroustrup, Bjarne, The Design and Evolution of C++．Addison – Wesley，1994
14 Bjarne Stroustrup. The C++ Programming Language. 3nd edition Addison Wesley Long man, 1997
14 [美]Jon Bates, Tim Tompkins 著，何建辉等译，实用 Visual C++ 6.0 教程，北京： 清华出版社，2001
15 [美]David E. Brumbaugh 著，柏路等译，C++面向对象的程序开发技术--构造 CASE TOOLS，北京：电子工业出版社，1996
16 Cameron Hughes，C++ iostream 面向对象 I/O 程序设计，尤晓东等译．北京：电子工业出版社，1997
17 Kaare Christian, Microsoft C++程序设计指南， 北京：清华大学出版社，1994
19 Greg Perry，C++程序设计教程，北京：清华大学出版社，1994
20 Tom Swan，C++编程秘诀，宋建云译, 北京：电子工业出版社，1994
21 Lafore R, Object-Oriented Programming in C++，北京：中国电力出版社，2004
22 Echel B, Thinking in C++，北京：机械工业出版社，2004
23 Stephen R, Davis，C++编程指南．卢凌云等译, 北京：电子工业出版社，1996
24 Sephch R, Davis，C++编程指南（续）卢凌云等译,北京：电子工业出版社，1997
25 Bruce Eckel，C++深入浅出, 侯雪萍等译, 北京：学苑出版社，1994
26 William H, Murray III，Chris H, Pappas, Borland C/C++从人门到精通，毛选等译．北京：电子工业出版社，1998
27 Scott Meyers 著，C＋+编程技巧， 陈迅雷等译, 上海: 上海科学普及出版社，1994
28 Sephen Blaha，最新 C＋十应用编程技术，孟庆昌等译，北京：国防工业出版社，1997
29 Kris Jamsa, 新编 C++自学教程.，凌涛等译，北京：电子工业出版社，1996
30 Namir C, Shammas, 面向对象的编程指南，宋炎等译，北京：电子工业出版社，1997
31 Jesse Liberty， C++自学通，路明等译， 北京：机械工业出版社，1997
32 Herbet Schildt，C++从入门到精通．马力文等译, 北京：学苑出版社，1994
33 Jesse Liberty，C++程序设计轻松人门．张文旭等译, 北京：机械工业出版社，1996
34 Deitel HM, Deitel PJ，C/C++程序设计大全，薛万鹏等译, 北京：机械工业出版社，1997
35 Michael Hyman Bob Arnson，学用 Visual++5．马岚等译, 北京：电子工业出版社，1998
36 Kris Jamsa， C/C++使用技巧 1001 例, 魏津等译, 北京：电子工业出版社，1996
37 Herbert Schildt，最新 C++语言精华（第 2 版）, 杨长虹等译．北京：电子工业出版社，1997
38 Sieve Oualline，实用 C++编程大全，辛运柿等译，北京：电子工业出版社，1997
39 Peter van der Linden，C 程序设计奥秘．张自力译，云南：云南科技出版社，1998
40 Herbert Schildt，C++学教程．石桥林等译, 北京：学苑出版社，1994
41 [美] Dattatri K，C++面向对象高效编程，潇湘工作室译，北京：人民邮电出版社
42 [美] Alexandrescu A，C++设计新思维（影印版），北京：中国电力出版社，2003

[美] Deitel HM, Deitel PJ 等编著，施平安译，C++程序设计教程——习题解答，北京：清华大学出版社，2004
[美] Dale N, Weems C, Headington M 等编著，C++程序设计(第二版，影印版)，北京：高等教育出版社，2001
[美] Zaratian B 著，Visual C++6.0Programmer’s Guide，北京：北京希望电脑公司，1998
[美] Bates J, Tonpkins T 著，何健辉等译，实用 Visual C++6.0 教程，北京：清华大学出版社，2000
[美] Krusw RL，Ryba AJ，C++数据结构与程序设计，钱丽萍译，北京：清华大学出版社，2004
[美] Josuttis NM 编著，C++ Standard library，Publisher: Addison Wesley，1999
[美] Savitch W 著，C++面向对象程序设计——基础、数据结构与编程思想，周靖译，北京：清华大学出版社，2004
[美] Overland B 著，C++语言命令详解，董梁等译，北京：电子工业出版社，2000
[美]Stevens AI 著，C++大学自学教程，林瑶等译，北京：电子工业出版社，2004
郑莉，董渊，张瑞丰，C++语言程序设计，北京：清华大学出版社，2003
刁成嘉，面向对象 C++程序设计，北京：机械工业出版社，2004
刘瑞新，曹建春，沈淑娟，张连堂等，Visual C++面向对象程序设计程，北京：机械工业出版社，2004
钱能，C++程序设计教程，北京：清华大学出版社，2003
温秀梅，丁学均，C++语言程序设计，北京：清华出版社，2004
谭浩强，C++程序设计，北京：清华大学出版社，2004
杨淑莹，VC++图像处理程序设计，北京：清华大学出版社，北方交通大学出版社，2003
刘斌，王忠，面向对象程序设计——Visual C++，北京：清华大学出版社，2003
甘玲 邱劲,面向对象技术与 Visual C++, 北京：清华大学出版社,2004
Stroustrup: C++, http://www.research.att.com/~bs/C++.html
C++ Home. http://www.cpp-home.com/
Boost C++ Libraries, http://www.boost.org/
C++ FAQ LITE, http://geneura.ugr.es/~jmerelo/c++-faq/
Cprogramming.com: Your Resource for C++ Programming, http://www.cprogramming.com/
cplusplus.com - The C++ resources network, http://www.cplusplus.com/
C++学习资源网，http://www.rayoko.com/
程序员大本营，http://www.csdn.net/
编程爱好者，http://www.programfan.com/