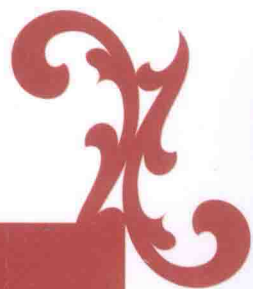




普通高等教育“十一五”国家级规划教材

教育部高等学校电工电子基础课程教学指导委员会推荐教材
电子信息学科基础课程系列教材



程序设计 (C++)

——基础、程序解析与实验指导

姚普选 编著



清华大学出版社



普通高等教育“十一五”国家级

教育部高等学校电工电子基础课程教学指导委员会推荐教材
电子信息学科基础课程系列教材



程序设计 (C++)

——基础、程序解析与实验指导

姚普选 编著

清华大学出版社

北京

内 容 简 介

本书以 C++ 语言为载体,介绍了程序设计的基本思想、常用的程序设计方法以及算法、数据结构的概念等程序设计的相关知识 with 技能。本书的主要内容包括:程序设计基础知识与 C++ 程序设计的一般方法;算法、数据结构的概念及应用;数据类型的概念以及 C++ 中的常用数据类型;面向对象程序设计方法;模板、异常处理等机制的概念及应用;输入输出流与数据文件的概念及应用。

本书将理论知识、程序实例与实验指导整合为一体,尽力为各教学环节的融会贯通创造条件。本书注重程序设计理念的先进性、程序设计方法的实用性以及学习过程中思维的连贯性,对于主要概念、常用方法以及具有递进关系的系列内容,都根据教学活动中的实际需求予以精心的编排与讲解。

本书可用作高等院校计算机程序设计课程的教材,也可与《程序设计教程(C++)》一书一起作为教材,还可作为程序设计工作者的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计(C++):基础、程序解析与实验指导/姚普选编著. —北京:清华大学出版社,2014
电子信息学科基础课程系列教材
ISBN 978-7-302-38095-5

I. ①程… II. ①姚… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 221079 号

责任编辑:文 怡

封面设计:常雪影

责任校对:白 蕾

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:19

字 数:439 千字

版 次:2014 年 12 月第 1 版

印 次:2014 年 12 月第 1 次印刷

印 数:1~2500

定 价:35.00 元

程序设计是一门逻辑性与实践性都很强的课程,学生必须由浅入深地研习其内在逻辑,循序渐进地阅读足量程序并且独立自主地完成相应实验(上机编辑、调试和运行程序)任务,才能在学习和实践中逐步理解程序设计的基础知识、掌握通过特定工具(程序设计语言、软件开发环境等)进行程序设计的基本技能,同时将渐次而来的对于程序设计本质的感悟内化为自己的科学素养。有鉴于此,笔者在《程序设计教程(C++)》^①教材的基础上,根据全国高等院校计算机教学指导委员会的相关文件,结合自己多年来的教学实践以及计算机基础教育的实际需求,精心编写了本书。

本书涵盖高等院校理工科“程序设计”课程的必要内容,这些内容编排在10章(分别对应于《程序设计教程(C++)》的第1~10章)之中:

- 程序设计的概念与C++程序设计的一般方法。
- 数据类型的概念、C++的基本数据类型与表达式。
- 算法的概念、算法的三种基本结构及其C++程序实现。
- 函数与编译预处理机制。
- 构造类型与顺序表操作。
- 指针应用与链表操作。
- 类和对象的概念及应用。
- 类的继承性与多态性的概念及应用。
- 模板、异常处理与命名空间机制。
- 输入输出流的应用与数据文件操作。

本书在选取教学内容时,注重程序设计理念的先进性和程序设计方法的实用性;在编排各部分内容时,尽可能照顾学生在学习过程中的思维连贯性;在讲解核心词语、抽象概念与重要技能时,详细说明其来龙去脉、优点与局限且常以简洁易懂的实例加以佐证。

本书兼顾各教学环节的实际需求,每章都编排了三部分内容:

- 基本知识:介绍程序设计的基础知识、基本技能及其C++程序实现方法。
- 程序解析:讲解相关程序设计任务、解决问题的思路、编程序所依据的算法、程序的运行结果以及修改或扩充程序的思路等。
- 实验指导:包括验证某种概念和方法的基本实验、运用多种概念和方法的综合性实验以及可能会引起思考或研究欲望的“启发性”实验。

^① 见《参考文献——1》

丰富多彩的程序解析是本书的一大特色。这些程序都经过了精心的选编、归并和讲解,作为相应章节的程序设计理念和方法的例证,可供学生研读、模仿或者改进和扩充。在确定编程序所依据的算法时,笔者尽量采用那些可以从相应概念或工作原理出发而自行构拟的算法,对于某些必须学习的传统或者经典算法,也尽可能讲清楚其来由、特点与适应范围。例如,在确定求解高次方程的算法时,首先从给定的形如 $f(x)=0$ 的方程式推导出形如 $x=g(x)$ 的迭代式,然后构拟通过这种自行推导出来的迭代式来逐步求得 x 值的算法;在需要使用经典的“牛顿迭代法”时,除了给出其一般形式的迭代式及其使用方法之外,还从其几何意义入手,讲解了构拟这种方法的依据。

每章的实验指导也都是按照教学活动中的实际需求精心编排的。一章中安排两到三个实验,每个实验往往需要编写并运行多个程序。这些实验中的几个程序往往自成一个由浅入深、循序渐进的体系;几个实验之间构成一个紧扣相关学习内容的完整体系;每章的实验又与前后各章互相照应,成为本书构拟的实验体系中不可或缺的一环。一般来说,按部就班地完成本书规定的实验任务,就可以基本掌握相应的知识和技能了。

本书还有三个附录:

附录 1,标准 ASCII 码表。

附录 2,程序调试与纠错的概念及其一般方法。

附录 3,Visual C++ 图形用户界面程序(Windows 程序)的概念及其程序设计方法。

本书可以作为高等院校程序设计课程的教材,也可与《程序设计教程(C++)》一书配套使用。采用本书作为教材的程序设计课程以 56~64(包括上机时数)学时为宜。学时太少时,可以少讲或不讲模板、输入输出流的概念、异常处理的概念等。学时较多时,可以讲解附录 3 的内容,并指导学生做两到三个模仿性的实验。另外,本书的内容选编以及讲解方式也照顾到了非在校大学生的程序设计工作者的需求,可以作为工作或学习过程中的参考书。

程序设计技术博大精深且仍处于不断发展变化之中,受篇幅、时间、读者定位、程序设计语言与环境以及作者水平等种种限制,这本书所涵盖的内容及所表达的思想可能会有所局限。因而,笔者希望传达给读者的信息是否正确或者是否得体,还要经过读者的检验。望广大读者批评指正。

姚普选

2014 年 8 月

第 1 章 程序设计的概念	1
1.1 基本知识	2
1.1.1 C++ 程序中的语句及命令行	2
1.1.2 C++ 程序中的函数	4
1.1.3 Visual C++ 的控制台工程	6
1.2 程序解析	8
程序 1-1 计算并联电阻	9
程序 1-2 求解二元一次方程组	10
程序 1-3 判断考试是否及格	11
1.3 实验指导	13
实验 1-1 C++ 程序的编辑、编译和运行	13
实验 1-2 简单 C++ 程序	18
实验 1-3 程序的调试和运行	23
第 2 章 基本数据类型与表达式	28
2.1 基本知识	29
2.1.1 基本数据类型、常量与变量	29
2.1.2 运算符与表达式	30
2.1.3 数据的输入与输出	33
2.2 程序解析	36
程序 2-1 整数加法练习	36
程序 2-2 输出 5 位整数的反序数	38
程序 2-3 统计参赛选手分数	40
2.3 实验指导	41
实验 2-1 不同类型数据的输入输出	41
实验 2-2 表达式求值	46
实验 2-3 数据的输出格式控制	48
第 3 章 算法与控制结构	50
3.1 基本知识	51
3.1.1 算法的概念与表示	51
3.1.2 算法的 3 种基本结构	52

目录

3.1.3 C++中实现选择结构和循环结构的语句	54
3.2 程序解析	56
程序 3-1 求三角形的面积	56
程序 3-2 确定某年某月的天数	57
程序 3-3 输出 100 以内能同时被 3 和 5 整除的数	59
程序 3-4 穷举法求组合数	61
程序 3-5 计算 $\sin x$ 函数的值	62
3.3 实验指导	64
实验 3-1 3 种基本结构	64
实验 3-2 迭代法与穷举法	66
第 4 章 函数与编译预处理	69
4.1 基本知识	70
4.1.1 函数的定义和调用	70
4.1.2 变量的作用域	72
4.1.3 变量的生存期	73
4.1.4 C++ 程序的多文件结构	75
4.1.5 编译预处理	77
4.2 程序解析	79
程序 4-1 将字符串中的小写字母转换为大写字母	79
程序 4-2 求 3 个实数中的最大数	80
程序 4-3 比较两个分数的大小	81
程序 4-4 求多项式的值	83
程序 4-5 牛顿迭代法求方程的根	84
4.3 实验指导	87
实验 4-1 函数的定义和调用	88
实验 4-2 函数的嵌套与递归调用	90
实验 4-3 编译预处理与多文件结构	92
第 5 章 构造类型与顺序表操作	95
5.1 基本知识	96
5.1.1 一维数组的定义和使用	96
5.1.2 二维数组的定义和使用	97
5.1.3 字符串处理	100

5.1.4	结构体的定义和使用	103
5.1.5	顺序表的概念	105
5.1.6	顺序表的 C++ 实现	106
5.2	程序解析	109
程序 5-1	顺序查找	109
程序 5-2	二分查找	112
程序 5-3	筛法求素数	114
程序 5-4	约瑟夫斯问题	117
程序 5-5	快速排序	119
程序 5-6	计算并输出学生成绩表	121
程序 5-7	枚举型变量的使用	123
5.3	实验指导	126
实验 5-1	数组的使用	126
实验 5-2	字符串的使用	129
实验 5-3	结构体及结构体数组的使用	132
第 6 章	指针与链表	135
6.1	基本知识	136
6.1.1	指针变量	136
6.1.2	动态存储分配	137
6.1.3	指针与数组和字符串	139
6.1.4	指针与函数	142
6.1.5	线性链表	146
6.2	程序解析	149
程序 6-1	三数排序并输出	149
程序 6-2	逆置数组元素	151
程序 6-3	计算圆的周长和面积	152
程序 6-4	按 3 位分节格式输出正整数	154
程序 6-5	线性链表求解约瑟夫斯问题	155
6.3	实验指导	157
实验 6-1	指针变量与取地址运算符	158
实验 6-2	指针与数组和字符串	159
实验 6-3	线性单链表的创建与查找	163

目录

第7章 类和对象	166
7.1 基本知识	167
7.1.1 类与对象的定义和使用	167
7.1.2 构造函数与析构函数	169
7.1.3 对象与指针	172
7.1.4 类的静态成员和友元	174
7.1.5 类的常对象与常成员	175
7.2 程序解析	177
程序 7-1 日期类	177
程序 7-2 日期时间类	179
程序 7-3 学生成绩类	181
程序 7-4 核对密码的学生成绩类	184
程序 7-5 矩阵的加减运算	187
7.3 实验指导	190
实验 7-1 修改并运行程序	190
实验 7-2 人员类及其对象	195
实验 7-3 椭圆类及其对象	196
第8章 类的继承性与多态性	198
8.1 基本知识	199
8.1.1 派生类的定义	199
8.1.2 派生类的继承方式	200
8.1.3 派生类的构造函数和析构函数	202
8.1.4 重载	205
8.1.5 虚函数	207
8.2 程序解析	210
程序 8-1 点类与直线类	210
程序 8-2 点类、圆类与圆柱类	212
程序 8-3 多重继承的工程硕士类	215
程序 8-4 通过函数重载求数组中最大元素	217
程序 8-5 抽象图形类以及圆、矩形和三角形类	219
8.3 实验指导	221
实验 8-1 类的继承性	222

实验 8-2 函数重载与运算符重载	224
实验 8-3 虚函数与抽象类	226
第 9 章 模板、异常处理和命名空间	230
9.1 基本知识	231
9.1.1 函数模板和类模板	231
9.1.2 异常处理	232
9.1.3 命名空间	233
9.2 程序解析	234
程序 9-1 模板函数实现矩阵加法	235
程序 9-2 类模板实现顺序表	237
程序 9-3 类模板实现数组越界报错	240
9.3 实验指导	242
实验 9-1 函数模板和类模板	242
实验 9-2 异常处理	245
第 10 章 输入/输出流	248
10.1 基本知识	249
10.1.1 输入/输出流	249
10.1.2 数据文件的打开与关闭	252
10.1.3 数据文件的读写	254
10.2 程序解析	257
程序 10-1 从输入流中分析出数字串	257
程序 10-2 通信录文本文件	259
程序 10-3 二进制方式打开指定文件	261
10.3 实验指导	262
实验 10-1 输出操作	263
实验 10-2 输入操作	264
实验 10-3 文件读写操作	265
附录 A ASCII 码表	269
附录 B 程序的调试与纠错	272
附录 C 图形用户界面应用程序	279
参考文献	293

第1章 程序设计的概念

本章的主要学习任务包括：

- 计算机程序设计的基本概念。
- C++程序的一般结构及其各组成部分的一般功能与特点。
- Visual C++^①软件的功能、特点以及控制台工程的创建、编辑和运行方法。

通过本章学习，可以初步理解计算机程序设计的意义、特点以及解决实际问题的一般方法，了解 C++ 程序的一般结构，掌握通过 Visual C++ 软件开发环境来编辑和运行 C++ 程序的一般方法。

^① 为方便起见，本书中将 Microsoft Visual Studio 软件开发环境简称为 Visual C++ 软件。

1.1 基本知识

一个 C++ 程序是由一系列 C++ 语句构成的,这些语句通常分门别类地组织成一个函数,通过函数之间的互相调用而成为一个整体。其中必有一个名为 `main()` 的主函数,是 C++ 程序执行的起点。

Visual C++ 是一种具有程序的编辑、编译、调试、连接装配以及保存和运行等一系列功能的集成开发环境,可在其中完成 C++ 程序的编写和运行过程中的所有工作。

1.1.1 C++ 程序中的语句及命令行

人在日常生活与生产活动中,需要解决各种各样的问题,往往都会经历先收集相关信息,再进行相应处理,最后给出结果这样 3 个阶段。与人处理问题的方式十分相似,在一个 C++ 程序中,往往也包含完成这 3 种任务的语句。

例 1-1^① 分析程序:

```
//例 1-1_求两数之和
#include <iostream>
using namespace std;
int main()
{   int x1,x2,y;
    x1 = 90;
    x2 = 88;
    y = x1 + x2;
    cout << "x1 + x2 = " << y << endl;
    return 0;
}
```

注: C++ 编译器区分字母的大小写,即将大写字母和相应的小写字母当作不同的字符。

该程序的功能为,先准备两个整数,再求两数之和,最后输出和数。

1. 程序中的主函数

程序的主要功能由主函数(名为“main”且由一对花括号“{”和“}”定界)中的一连串 C++ 语句来完成。一个 C++ 程序总是从 `main()` 函数开始执行,当 `main()` 函数执行完毕返回时程序结束,无论 `main()` 函数放在程序的任何位置都会如此。

本程序的主函数中包含多条语句,可以按功能分为 3 部分:准备数据部分、运算部分和输出结果部分。

^① 本书中例题多为《程序设计教程 C++》(姚普选等编著,清华大学出版社 2011 年出版)中的习题,本例就是原教材中第 1 章的第 8 题。

(1) 语句

```
int x1, x2, y;
```

定义了三个整型变量 $x1$ 、 $x2$ 和 y ，分别用于存放两个整数以及和数。语句

```
x1 = 90;
```

```
x2 = 88;
```

给两个整型变量赋值，即将两个整数分别存放到两个变量中（实际上是存放到变量所表示的存储单元中）去。如果将这两条语句改为

```
cin >> x1 >> x2;
```

则可在程序运行后，由用户临时输入两个数来给两个变量赋值。输入时，两个数之间用空格隔开，两个数之后加上回车键。

注：C++ 语言规定：程序中要用到的所有变量都必须先定义，然后才能使用。

(2) 语句

```
y = x1 + x2;
```

进行加法运算，即将右式中两个变量的值相加并将和数赋值给左式的变量（存放到该变量所表示的存储单元中）。

(3) 语句

```
cout << "x1 + x2 = " << y << endl;
```

用于输出和数，其中包含两个输出项 " $x1 + x2 =$ " 和 y ，前者是一个将要原样输出的字符串，后者是将要输出其值的存放了和数的变量。

(4) 语句

```
return 0;
```

比较特殊，这是因为 ANSI/ISO 标准 C++ 要求 $\text{main}()$ 函数定义为 int 型且某些操作系统要求在执行了一个程序之后向操作系统返回一个数值才加入的。

注：C++ 往往这样处理：如果程序正常运行，则向操作系统返回 0，否则返回 -1。

2. 程序中的注释

本程序中的第 1 行

```
//例 1-1_求两数之和
```

是为了便于阅读理解程序而编写的注释。注释可以放在一行中的“//”符号后面或者放在“/*”和“*/”符号之间，不影响程序的执行。

3. 程序中的文件包含命令

本程序中的第 2 行

```
#include <iostream>
```

包含了头文件 `iostream`, 这是 C++ 的标准输入/输出库文件, 其作用是给程序提供与输入/输出操作相关的信息, 当一个程序包含了该文件时, 就会自动定义与键盘输入相关联的输入流对象 `cin` 以及与屏幕输出相关联的输出流对象 `cout` 在内的多个标准流。也就是说, 有了这一行, 其后的程序中就可以使用 `cin` 来输入变量的值并使用 `cout` 来输出变量的值了。

这种文件包含命令是 C++ 的“编译预处理命令”, 用于指示编译器在对本程序进行预处理时, 将另一段 C++ 源程序文件的内容嵌入本程序的相应位置上。当前源程序文件和文件包含命令嵌入的源程序文件在逻辑上看作同一个文件, 经过编译后生成统一的目标文件。

注: 文件包含命令以“#”开头且末尾没有分号。

4. 程序中的指定命名空间语句

本程序中第 3 行

```
using namespace std;
```

的意思是, 使用命名空间 `std`。C++ 标准库中的函数和类是在命名空间 `std` 中定义的, 故当程序中要用到 C++ 标准库时, 就需要这样声明。

1.1.2 C++ 程序中的函数

C++ 语言采用函数的形式来组织程序。一个 C++ 程序常由多个函数组成, 函数之间是互相独立的, 但可以互相调用。

例 1-2 分析程序:

//例 1-2 求分段函数的值

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

//求 y 值的函数

```
float yy(float xx)
```

```
{ if (xx >= 0)
```

```
    return pow(xx, 3) - 9;
```

```
    else
```

```
    return sin(xx) + 3;
```

```
}
```

//主函数

```
int main()
```

```
{ float x, y;
```

```
  cout << "x = ?" << endl;
```

```
  cin >> x;
```

```
  y = yy(x);
```

```

    cout << "y = " << y << endl;
    return 0;
}

```

本程序的功能是,输入变量 x 的值,根据下面的函数计算 y 的值,然后输出 y 值。

$$y = \begin{cases} x^3 - 9 & (x \geq 0) \\ \sin(x) + 3 & (x < 0) \end{cases}$$

1. 程序中的用户自定义函数

本程序的第 6~11 行是一个名为“yy”的用户自定义函数,它由函数头语句

```
float yy(float xx)
```

和一对花括号“{”和“}”括起来的函数体构成。

(1) 头语句的括号中定义了浮点型参数 xx (用 `float` 定义的浮点型自变量),它只有在被其他函数中的相应语句调用时才会有值,因而称为形式参数(或虚拟参数)。函数名 `yy` 之前的 `float` 指定了该函数在被调用而执行后得到的函数值是浮点数,这个值是该函数体中的 `return` 语句赋予函数名的。

(2) 函数体内的 if 语句

```

if(xx >= 0)
    return pow(xx, 3) - 9;
else
    return sin(xx) + 3;

```

根据题目的要求,使用参数 xx 计算函数的值并将结果赋予函数名。if 语句中包含了条件“ $xx \geq 0$ ”并嵌入了两条 `return` 语句。该语句在执行时,根据条件是否成立来确定执行哪条 `return` 语句。

2. 程序中的函数调用

在本程序中定义了 `yy()` 函数之后, `main()` 函数中的语句

```
y = yy(x);
```

调用了该函数。调用时,使用已被赋值的变量 x 代替了函数中原有的形式参数 xx 。调用时使用的变量 x 称为实际参数。

在 `yy()` 函数因被调用而执行时,其中的形式参数 xx 得到了实际参数 x 传递过来的值,计算得到函数的值并将其值赋予“=”左边的 y 变量。

`yy()` 函数也可以在 `main()` 函数之后定义,但当 `main()` 函数需要调用在它后面定义的 `yy()` 函数时,需要先使用语句

```
float yy(float x);
```

或简写为

```
float yy(float);
```

来声明它。

3. 程序中调用的标准函数

在 yy() 函数体内的 if 语句

```
if(xx >= 0)
    return pow(xx,3) - 9;
else
    return sin(xx) + 3;
```

中,嵌入了两条 return 语句。它们分别调用了函数 pow() 和 sin(), 这两个函数都是 Visual C++ 在“cmath”头文件中预先定义好的,称为标准函数。必要时,只需添加一个本程序第 3 行那样的文件包含命令

```
#include <cmath>
```

就可以在后面的程序中调用它们了。

1.1.3 Visual C++ 的控制台工程

Visual C++ 为用户提供了可视化的集成开发环境,其中包括了所有设计、调试、配置应用程序所用到的工具。通过这些工具,可以很容易地创建程序中的代码和可视化部分,及时地观察界面设计过程中的变化,并利用调试功能来查错和纠错,从而快速地设计出符合要求和使得用户满意的应用程序。

1. 工程的概念

在 Visual C++ 软件中,一个程序称为一个工程或项目(Project)。通过 Visual C++ 进行 C++ 程序设计时,需要创建工程并在其中完成 C++ 源代码的编辑以及程序的编译、调试、连接、保存和运行等一系列工作。

工程又称为项目。它有两层含义:一是指最终生成的应用程序;二是指为了创建这个应用程序所需要的全部文件的集合,包括各种源程序文档、资源文件和配套的文档等。目前大多数流行的软件开发工具都利用工程来对软件开发过程进行管理。

Visual C++ 中编写并处理的任何程序都与工程有关(都要创建一个与之相关的工程),工程用于组织应用程序的资源。一个工程对应一个文件夹,工程名就是文件夹名。一个工程中的所有文件都存放在相应的文件夹中,包括源程序代码文件(.cpp、.h)、工程文件(.dsp)、工程工作区文件(.dsw)以及工程工作区配置文件(.opt)等,可能还会包含 Debug(调试)、Release(发行)或者 Resource(资源)等子文件夹。

2. 工程与工作区

Visual C++ 中的每个工程都与一个工程工作区相关联。Visual C++ 通过对工程工作

区的操作,可以显示、修改、添加、删除资源文件。最简单的情况下,一个工作区中存放一个工程,代表着某个要进行处理程序。必要时,也可以把多个工程存放在一个工作区中,其中可以包含某个工程的子工程或者与其相互依赖的其他工程。也就是说,Visual C++允许用户在一个工作区内添加多个工程,其中有一个是活动的(默认的),每个工程都可以独立地编译、连接和调试。

注:本书中的例子都是在一个工作区中存放一个工程。

可见,工程工作区就像是一个“容器”,由它来“盛放”相关工程的所有信息,在创建一个新工程时,同时要创建这样一个工程工作区,然后通过相应的窗口来观察与存取该工程的各种元素及其相关信息。

创建了工程工作区之后,系统自动创建一个相应的工作区文件来存放该工作区的的相关信息。另外,还会创建其他几个相关文件,如工程文件以及选择信息文件等。

3. Visual C++的控制台工程

Visual C++中预置了多种类型的工程。用户在编程序时,首先要选定合适的工程类型,然后创建工程并在其中完成程序的编辑、编译、连接、调试与运行等一系列工作。选定不同的工程类型意味着通知 Visual C++系统帮助自己提前做好预期的准备工作(初始化工作)。例如,事先自动生成一个底层程序框架(可称为框架程序)或者进行某些隐含设置(如隐含位置、预定义常量、输出结果类型等)。

在诸多不同类型的工程中,“控制台工程(Win32 Console Application)”是一种用于C++程序设计的最简单的工程类型。本书中大多数程序都是通过这种工程来编写和处理的。

创建了一个控制台工程或者打开了一个已有的控制台工程之后,Visual C++软件就会在自己的窗口中显示相应的源代码编辑窗口,用户可在此编辑C++程序的源代码并通过Visual C++窗口中的其他功能来完成一个C++生命周期中的所有工作。

注:Visual C++的控制台工程小巧而简单,但足以解决并支持本课程中涉及的绝大多数程序设计内容和技术,使我们把关注的重心放在程序本身而非界面处理等其他方面。

4. C++程序设计的一般过程

如果已经确定了解决问题的“算法”并且确定了要在Visual C++中编写C++程序来实现这种算法,则可以按照以下步骤来完成程序设计任务:

(1) 启动 Visual C++,在其中创建一个工程。

(2) 编辑 C++程序,其中包括用于输入数据、进行相应处理以及用于输出处理结果的C++语句。

(3) 编译 C++程序,就是调用编译程序来把 C++语言的源程序翻译成计算机可以识别的目标程序(由二进制代码构成)。在编译 C++程序的过程中,要进行词法分析、语法分析、语义检查(生成中间代码)、代码优化以及生成目标代码并给出相应的提示信息。