



普通高等教育“十五”国家级规划教材配套参考书

A Laboratory for High Level  
Language C++ Program Design

# 高级语言C++ 程序设计实验指导

( 第二版 )

周玉龙 刘 璟



高等教育出版社  
Higher Education Press

普通高等教育“十五”国家级规划教材配套参考书

# 高级语言 C++ 程序设计 实验指导

(第二版)

周玉龙 刘 璟

高等教育出版社

## 内容提要

本书是《高级语言 C++程序设计》(第二版)(高等教育出版社, 2004 年 11 月)的配套教材, 主要目标在于培养与训练学生的自我动手能力, 学习并掌握从程序编写、程序调试到上机运行等一整套的具体实现方法与实验技术。两书内容相互衔接, 相互补充, 使所学理论与实践紧密结合, 以提高解决与处理各种具体问题的实际编程和创新能力。

本书采用一种“示范—帮扶—放手”的章节层次编写方式, 使整个教与学的过程循序渐进, 便于初学者从生疏到掌握, 而后逐步达到熟练。

### 图书在版编目(CIP)数据

高级语言 C++程序设计实验指导/周玉龙, 刘璟.  
—2 版. —北京: 高等教育出版社, 2006.3  
ISBN 7-04-018657-8

I. 高... II. ①周...②刘... III. C 语言-程序设  
计-高等学校-教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 008629 号

策划编辑 何新权 责任编辑 焦建虹 封面设计 于文燕  
版式设计 胡志萍 责任校对 张颖 责任印制 杨明

---

出版发行	高等教育出版社	购书热线	010-58581118
社 址	北京市西城区德外大街 4 号	免费咨询	800-810-0598
邮政编码	100011	网 址	<a href="http://www.hep.edu.cn">http://www.hep.edu.cn</a>
总 机	010-58581000		<a href="http://www.hep.com.cn">http://www.hep.com.cn</a>
		网上订购	<a href="http://www.landaco.com">http://www.landaco.com</a>
			<a href="http://www.landaco.com.cn">http://www.landaco.com.cn</a>
经 销	蓝色畅想图书发行有限公司	畅想教育	<a href="http://www.widedu.com">http://www.widedu.com</a>
印 刷	北京宏伟双华印刷有限公司		
开 本	787×1092 1/16	版 次	2003 年 3 月第 1 版 2006 年 3 月第 2 版
印 张	19.5	印 次	2006 年 3 月第 1 次印刷
字 数	430 000	定 价	24.50 元

---

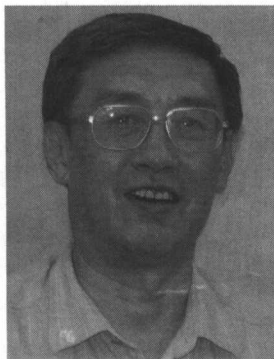
本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究  
物料号 18657-00

## 作者简介



**周玉龙** 南开大学信息技术科学学院计算机科学与技术系教授，兼任教育部理工类计算机基础课程教学指导分委员会委员。自毕业留校任教后一直从事计算机软件与应用方面的教学与科研工作，编写并出版了多本与计算机程序设计有关的教材，主讲多门计算机软件与应用专业的本科生及研究生的课程（例如“计算机引论”、“高级语言程序设计”、“编译方法与技术”、“人工智能”等）。主要研究领域为计算机应用以及面向对象程序设计与编程技术、机器学习与机器翻译及智能软件的研究与开发技术、数据库管理与应用技术等。



**刘璟** 南开大学信息技术科学学院计算机科学与技术系教授，博士生导师，兼任教育部计算机科学与技术教学指导委员会委员，非计算机专业计算机基础课程教学指导分委员会副主任，天津市高等学校计算机基础教学指导委员会副主任，中国计算机学会理论计算机科学分会理事，天津市学位委员会学科评议组成员。

长期讲授“高级语言程序设计”，“算法设计与分析”，“程序设计方法学”，“数据结构”等课程。主要研究领域为并行与分布式系统、算法设计与分析、网络存储系统、计算智能与字符图像识别、面向对象程序设计等。曾主持国家 863、科技攻关、自然科学基金、博士点基金项目等十余项研究课题，在国内外发表论文 60 篇，出版教材《计算机算法引论》、《高级语言 C++ 程序设计》、《C++ 语言速成教程》等。

# 前 言

计算机编程技术一直受到国内外各高等学校以及计算机界的高度重视。以面向对象技术为基础的 C++ 程序设计方法与技术代表了目前世界上最先进的计算机编程方法与技术。

本书是《高级语言 C++ 程序设计》(第二版)(高等教育出版社, 2004 年 11 月)的配套教材, 两书内容相互衔接, 相互补充, 使所学理论与实践紧密结合。

程序设计课程的实践性非常强, 其教学的主要目的和目标就是教会学生如何编写程序, 使计算机为人们服务。程序设计就像一门艺术, 需要学生动手操练并充分发挥其创造能力。因此, 注重教学实践, 强化实验课与上机实习就显得尤为重要, 所以我们编写了这本具有实验指导性质的教材。

通过分析 & 处理精选的一系列实践性问题, 学习并掌握从问题分析、程序编写、程序调试到上机运行等一整套的具体实现方法与实验技术, 进而提高分析、解决各种实际问题的能力。

本书由十章构成: 第一章 C++ 语言初步、第二章基本数据类型与基本运算、第三章分支与循环、第四章数组与结构、第五章函数及其应用、第六章指针与引用、第七章类与对象、第八章继承与派生、第九章模板、第十章输入输出流。

本书采用一种“示范—帮扶—放手”的章节层次编写方式, 此方式可以形象地比喻为, 我们如何有效而快速地教会一个小孩(学生)去走路或做某件事情的教学方法, 使整个教与学的过程循序渐进, 便于初学者从生疏到掌握, 而后逐步达到熟练。

① “示范”: 给学生做示范, 教他们如何走、如何做。

给出“示范题”的完整求解方法与具体实现, 介绍如何对它们进行调试以及完成整个实验的各种使用技术, 让学生从中观摩、体会、学习, 逐步掌握如何设计与求解同类问题的方法与技术。

② “帮扶”: 帮扶着学生, 让他们练着自己走、自己做。

给出“实践题”的求解方法、实现方案以及实现“框架”, 让学生通过进一步思考, 并对问题进行进一步细化, 模仿“示范题”, 而后自己编制出求解程序, 得到提高。

③ “放手”: 放手让学生自己走、自己做, 但通常事先提示一些重点或注意点。

给出一批“自立题”, 但仅给出某些重点提示或某些注意点, 让学生自己思考, 自己设计并完成求解任务, 提高学生独立设计、编写程序解决问题的能力。

通过“示范”, 而后让学生进行“实践”, 最后达到“自立”的目标。

具体地说, 对每一章, 都按照如下三个层次(即三个节)来进行具体内容的组织与编排:

- 示范题的求解与剖析(“示范”, 1~5 个示范题)。
- 实践题的求解方法与提示(“帮扶”, 2~10 个实践题)。

- 自立题的求解提示与注意（“放手”，3~26个自立题）。

对每一个示范题，均从如下五个方面进行论述与讨论：**【实验目的】**、**【实现方法】**、**【程序编写】**、**【调试运行】**、**【剖析点评】**。

对每一个实践题，从如下三个方面来叙述：**【实验目的】**、**【实现方法】**、**【编程提示】**。

对每一个自立题，则只给出**【实验目的】**和**【要点提示】**。

本书所涉及的程序均在 Visual C++ 6.0 集成环境下调试通过，所介绍使用的上机调试与运行环境也使用 Visual C++ 6.0。

在本书编写过程中，南开大学计算机系的刁成嘉、邵秀丽、刘晓光、李涛等老师提出过诸多有益的建议，在此一并表示衷心的感谢！

书中也许会存在错误与不足，敬请阅读本书的老师和同学们予以批评指正。

编 者

2005 年于南开大学

# 目 录

<b>第一章 C++语言初步</b> .....1	6.1 示范题的求解与剖析..... 153
1.1 示范题的求解与剖析.....1	6.2 实践题的求解方法与提示..... 161
1.2 实践题的求解方法与提示.....5	6.3 自立题的求解提示与注意..... 172
1.3 自立题的求解提示与注意.....8	
<b>第二章 基本数据类型与基本运算</b> .....10	<b>第七章 类与对象</b> ..... 187
2.1 示范题的求解与剖析.....10	7.1 示范题的求解与剖析..... 187
2.2 实践题的求解方法与提示.....15	7.2 实践题的求解方法与提示..... 194
2.3 自立题的求解提示与注意.....17	7.3 自立题的求解提示与注意..... 201
<b>第三章 分支与循环</b> .....20	<b>第八章 继承与派生</b> ..... 214
3.1 示范题的求解与剖析.....20	8.1 示范题的求解与剖析..... 214
3.2 实践题的求解方法与提示.....32	8.2 实践题的求解方法与提示..... 231
3.3 自立题的求解提示与注意.....40	8.3 自立题的求解提示与注意..... 240
<b>第四章 数组与结构</b> .....53	<b>第九章 模板</b> ..... 250
4.1 示范题的求解与剖析.....53	9.1 示范题的求解与剖析..... 250
4.2 实践题的求解方法与提示.....63	9.2 实践题的求解方法与提示..... 254
4.3 自立题的求解提示与注意.....73	9.3 自立题的求解提示与注意..... 259
<b>第五章 函数及其应用</b> .....98	<b>第十章 输入输出流</b> ..... 268
5.1 示范题的求解与剖析.....98	10.1 示范题的求解与剖析..... 268
5.2 实践题的求解方法与提示.....112	10.2 实践题的求解方法与提示..... 278
5.3 自立题的求解提示与注意..... 124	10.3 自立题的求解提示与注意..... 288
<b>第六章 指针与引用</b> ..... 153	<b>参考文献</b> ..... 300

# 第一章 C++语言初步

## 1.1 示范题的求解与剖析

### 一、示范题一

使用 C++ 的输出语句，在屏幕上显示如下 5 行（为 5 个字符串，其中的第 2 行为空行）：  
My name is Lili. My schedule:

```
MON    TUE    WED    THU    FRI    SAT    SUN
-----
Eng     C++    Math   C++    Math
```

#### 【实验目的】

1. 熟悉并使用对字符串进行输出的语句的一般形式：`cout<<"任意字符串"`。掌握转义字符“`\n`”、“`\t`”以及格式控制符“`endl`”的使用方法。
2. 熟悉 Visual C++ 6.0 的集成开发环境。熟悉并掌握从输入到运行一个程序的操作过程：创建工程，输入源程序代码，对程序进行编译、连接后运行（得出结果）。

#### 【实现方法】

1. 可利用如下形式的语句来输出单独占一行的“任意字符串”：

```
cout<<"任意字符串\n";
```

也可将上句改写为：

```
cout<<"任意字符串"<<endl;
```

2. 通过使用转义字符“`\t`”，可使随后的输出在下一个“制表位”处进行（默认情况下，按每 8 个字符位置来划分“制表位”）。

#### 【程序编写】

1. 由于要使用 `cout` 进行输出处理，所以必须使用预处理指令“`#include <iostream.h>`”。
2. 输出具有对齐关系的第 3 行和第 5 行时，可利用“`\t`”来进行位置以及上下行的对齐控制。



3. 转义字符“\n”表示换行,使得随后的输出位置“转移”到下一行的开头处。格式控制符“endl”的使用功能与“\n”相同,也用于实现换行。

```
#include <iostream.h>
void main(){
    cout<<"My name is Lili. My schedule:\n"<<endl;
    cout<<"MON\t"<<"TUE\t"<<"WED\t"<<"THU\t"<<"FRI\t"<<"SAT\t"<<"SUN"<<endl;
    cout<<"-----\n";
    cout<<"Eng\tC++\tMath\tC++\tMath"<<endl;
}
```

### 【调试运行】

首先启动 Visual C++ 6.0,进入到它的集成开发环境窗口。

接着创建工程并输入源程序代码。其操作步骤为:“File”→“New”→“Projects”→“Win32 Console Application”→(右上边第二行)“Location”文本框中填入如“d:\mydir”→(右上边第一行)“Project name”文本框中填入如“proj1”→“OK”→“Finish”→“OK”→“Project”→“Add To Project”→“New”→“C++ Source File”→(右边)“File”文本框中填入文件名如“exe1”(此时系统将使用名为 exe1.cpp 的文件来保存所键入的源程序)→“OK”→键入具体程序。

注意:在“Location”文本框中填入如“d:\mydir”,这是假设准备在 d 磁盘的 mydir 文件夹即子目录下存放与工程工作区相关的所有文件及其相关信息(当 d 磁盘上不存在 mydir 文件夹时,系统将自动创建该名字的文件夹。若准备指定其他位置,则要在此处填入其相关“路径”),当然也可通过单击其右部的“...”按钮来选择并指定其他已存在文件夹的位置。

最后对所输入的程序进行编译、连接后运行(得出结果)。此三个步骤所使用的菜单命令依次为:

“Build”→“Compiler ...”;

“Build”→“Build ...”;

“Build”→“Execute ...”。

程序执行结束后,可通过“File”→“Close Workspace”关闭相关的工作区窗口,从而结束了—一个程序从输入到执行的全过程,而重新回到刚刚启动 Visual C++ 6.0 时的那个初始画面(窗口)。

若要打开某个已关闭的工作区 Workspace 以再一次修改或执行其相关程序时,可通过“File”→“Open Workspace”,并在“查找范围”处指定具体的工程文件夹(如“mydir”下的“proj1”),而后通过鼠标单击选定“proj1.dsw”,再单击右下角处的“打开”。

注:打开某个已关闭工作区 Workspace 的另一种常用方法是,通过“资源管理器”找到相关的工程文件夹,如“mydir”下的“proj1”,选定该文件夹下的“proj1.dsw”,双击该文件。

### 【剖析点评】

1. 注意程序的每一条语句均必须以分号作为结束。

2. 所编写的程序中没有使用任何变量, 所以不需要进行变量说明。

3. 往 cout (即屏幕) 上输出信息时, 更具有一般性的语句格式为:

```
cout<< 输出项 1 << 输出项 2 << ... << 输出项 n;
```

其中的各输出项通常为常量、变量、字符串 (更一般地说, 可为一个表达式) 以及格式控制符等。由 cout 负责计算并依次在屏幕上显示它们 (各输出项) 的值。

4. 下面的前两条语句与后两条语句作用完全相同。

```
cout<<"MON\t"<<"TUE\t"<<"WED\t"<<"THU\t"<<"FRI\t"<<"SAT\t"<<"SUN"<<endl;
```

```
cout<<"Eng\tC++\tMath\tC++\tMath"<<endl;
```

```
cout<<"MON\tTUE\tWED\tTHU\tFRI\tSAT\tSUN"<<endl;
```

```
cout<<"Eng\t"<<"C++\t"<<"Math\t"<<"C++\t"<<"Math"<<endl;
```

## 二、示范题二

编写程序, 从键盘输入整数  $x$ 、 $y$ , 计算  $x^2+x-y$  以及  $w=2\pi x+\pi y^2$ , 并将结果显示在屏幕上。

### 【实验目的】

1. 熟悉从 cin (即键盘) 上输入数据时所使用的输入语句格式以及执行到此种语句时的“人机交互”过程。注意 cout 输出项是表达式时系统的处理方式 (cout 将负责计算该表达式的值并显示出来)。

2. 进一步熟悉 Visual C++ 6.0 的集成开发环境。将第二个程序所涉及的所有相关元素与信息仍存放在“d:\mydir”文件夹下, 再单独创建一个叫做 proj2 的工程工作区 (以具体保存相关信息), 并使用 exe2.cpp 文件来存放源程序, 而后对程序进行输入、编译、连接、运行 (以及多次运行)。

### 【实现方法】

1. 为了存放从键盘输入的两个整型 (int 型) 数据  $x$ 、 $y$  以及实型数据  $w$ , 需要使用说明变量数据类型, 并可得到变量数据存储空间的说明语句: “int x,y; double w;”。

2. 从 cin (即键盘) 上输入数据时, 可使用如下格式的输入语句:

```
cin>> 变量名 1 >> 变量名 2 >> ... >> 变量名 n;
```

当执行到此种语句时, 计算机将“暂停”下来, 等候用户从键盘输入多个值 (多个输入值之间通常以“空格”或“换行”符号分隔), 而后 C++ 系统将把这些输入值依次赋值给 cin 中的各个变量 (注意 cin 中必须使用变量而不允许使用常量或非变量的表达式)。

3. 由于 cout 的输出项可以是字符串也可以是表达式, 从而可设计并使用如下语句: “cout<<"x\*x+x-y="<<x\*x+x-y<<endl;”, 其第 1 输出项为字符串 (“原样输出”), 第 2 输出项为表达式 (系统负责计算并输出其结果值), 第 3 输出项为格式控制符 endl (表示换行)。

**【程序编写】**

```

#include <iostream.h>
void main() {
    int x,y;    double w;
    const double pai=3.1416;           //说明 double 型有名常量 pai
    cout<<"Input x,y:";
    cin>>x>>y;
    cout<<"x*x+x-y="<<x*x+x-y<<endl;   //输出表达式的结果值
    w=2*pai*x+pai*y*y;
    cout<<"w="<<w<<endl;             //输出变量 w 的当前值
}

```

**【调试运行】**

若要输入并运行第二个程序（例如，准备在“d:\mydir”文件夹下再单独创建一个叫做proj2的工程工作区，并使用exe2.cpp文件来存放源程序），则要再一次重复类似于示范题一中的操作步骤：“File”→“New”→“Projects”→“Win32 Console Application”→（右上边第二行）“Location”文本框中填入如“d:\mydir”→（右上边第一行）“Project name”文本框中填入如“proj2”→“OK”→“Finish”→“OK”→“Project”→“Add To Project”→“New”→“C++ Source File”→（右边）“File”文本框中填入文件名如“exe2”（使用名为exe2.cpp的文件来保存源程序）→“OK”→从键盘输入第二个具体程序。

之后通过Build菜单的各命令项对程序进行编译、连接而后运行。

当执行到cin语句时，计算机将“暂停”下来，等候用户从键盘输入本程序的x、y值（两个输入值间以“空格”分隔，最后以“换行”结束输入）。第1次执行后，“交互”界面及结果为：

```

Input x,y:2 1
x*x+x-y=5
w=15.708

```

注意：可通过“Build”→“Execute…”来多次执行程序，对不同的输入数据测试各自的执行结果（像示范题一那样的程序，如果其中没有输入的话，则每次都应得到相同的输出结果）。第2次执行后，结果为：

```

Input x,y:5 8
x*x+x-y=22
w=232.478

```

程序执行结束后，通过“File”→“Close Workspace”关闭相关的工作区窗口。

**【剖析点评】**

1. 该程序虽小，但却具有一般通用程序所涉及的 3 个主要部分：
  - 输入初值——如通过 `cin` 为变量 `x`、`y` 赋以初始值。
  - 相关处理——如利用初始值 `x`、`y` 来计算 `w`，或进行某些加工处理。
  - 输出结果——如输出 `w` 以及与 `x`、`y` 相关的表达式值等。
2. 程序中定义并使用了 `double` 型的有名常量 `pai`，其值为 3.1416，表示常数  $\pi$  的一个近似值，该值在程序中不可改变。
3. 为了增强程序的可读性，通常一条 C++ 语句占一行。但 C++ 允许一行出现多条语句，也允许一条语句占用多行。如本程序的第 3 行 “`int x,y; double w;`” 就出现了两条说明语句。

## 1.2 实践题的求解方法与提示

### 一、实践题一

使用 C++ 的输出语句，在屏幕上显示出如下 3 行（3 个字符串）：

Hello! How are you!

I am a student.

C -> C++ -> Visual C++ wonderful!

**【实验目的】**

熟悉如何对字符串进行输出，掌握转义字符 “`\n`” 以及格式控制符 “`endl`” 的使用方法。

熟悉 Visual C++ 6.0 的集成开发环境。熟悉并掌握对一个程序进行输入、编译、连接以及运行的全过程。

**【实现方法】**

1. 使用如下形式的语句来输出单独占一行的 “任意字符串”：

```
cout<<"任意字符串"<<endl;
```

2. 上机对所编写的程序进行输入、编译、连接、运行。

注意：在创建 C++ 工程时，首先要确定准备今后固定在哪一磁盘（如 d 盘）的哪一个文件夹（如 d 盘中的 `mydir`）下存放上机过程中的所有相关信息，而后则在 “交互问答” 过程的 “Location” 文本框中填入该磁盘那一文件夹的 “路径”（如 “`d:\mydir`”）。

**【编程提示】**

由于要往 `cout`（即屏幕上）输出信息，别忘记使用编译预处理指令 “`#include <iostream.h>`”。

## 二、实践题二

在计算机上对《高级语言 C++程序设计》(第二版)(以下简称主教材)中的程序 program2-2.cpp 进行编辑、编译、运行(求解 100 元买 100 只鸡的问题)。

```
//program2-2.cpp
#include <iostream.h>
void main(void) {
    int chicken,hen,cock;
    cout<<"chicken hen  cock"<<endl;
    for(chicken=0; chicken<100; chicken+=3)
        for(hen=0; hen<=33; hen++)
            if((cock=100-chicken-hen)> -1)
                if(chicken/3+hen*3+cock*5==100)
                    cout<<"  "<<<chicken<<"  "<<<hen<<"  "<<<cock<<endl;
}
```

### 【实验目的】

进一步熟悉对程序进行输入、编译、连接,而后运行得到输出结果的全过程。对上述求解比较复杂问题的程序实例及其程序一般结构,以及使用 C++编程求解问题的“广泛性”,可以得到一个直观的感觉。

### 【实现方法】

上机对程序 program2-2.cpp 进行输入、编译、连接、运行。

创建 C++工程进行“交互问答”的过程中,在“Location”文本框内填入存放上机过程中相关信息的磁盘及文件夹的“路径”(如在上一个实验时指定的“d:\mydir”)。

### 【编程提示】

该程序中使用了变量、循环语句、分支语句等一些语法成分。并不期待读者目前可以理解且能够编写此种复杂程度的求解程序。

## 三、实践题三

叙述下述程序的功能,分析它执行后的输出结果,并上机进行测试验证。

```
#include <iostream.h>
void main() {
    int x1,x2,x3,x4,x5,x6,x7,x8,x9,x10;
```

```
cout<<"Input 10 integers:"<<endl;
cin>>x1>>x2>>x3>>x4>>x5>>x6>>x7>>x8>>x9>>x10;
cout<<"---- The result ----"<<endl;
cout<<x10<<" "<<x9<<" "<<x8<<" "<<x7<<" "<<x6<<" ";
cout<<x5<<" "<<x4<<" "<<x3<<" "<<x2<<" "<<x1<<endl;
}
```

### 【实验目的】

读懂程序而后叙述其功能（要实现什么，以及具体做些什么），并上机将程序输入，最后运行得到输出结果，以验证自己的想法。

进一步熟悉如何从 cin（即键盘）上输入数据以及执行到此种语句时的“人机交互”过程，并熟悉 cout 各输出项的使用格式及功能。

### 【实现方法】

1. 说明 10 个 int 型变量存放所输入的数据。
2. 屏幕提示，让用户输入 10 个整数。
3. 通过 cin 输入 10 个整数，依次放入各变量（所占据的存储单元）之中。
4. 使用 cout 按输入的反顺序输出各变量（值），数之间以空格分隔。

### 【编程提示】

1. 程序倒数第 2 条 cout 语句（输出 x10 到 x6）的结尾处没有输出“换行”，从而使下一条 cout 语句（输出 x5 到 x1）输出的结果将紧随上一行的数据，处于同一行上。

2. 本书随后还将设计并提供多种其他实现方式来解决这个问题，本实现是其中最“笨”的一种实现方式。

## 四、实践题四

编写程序，从键盘输入实数  $a$ 、 $b$ 、 $c$ ，计算出  $w = b^2 - 4ac$  以及  $t = \pi w^2 + 2a - bc$ ，并将结果  $w$  与  $t$  显示在屏幕上。

### 【实验目的】

编写具有输入输出并进行简单表达式计算与赋值的程序，上机运行，实现所指定的功能。

### 【实现方法】

1. 说明 double 型变量  $a$ 、 $b$ 、 $c$  以及  $w$ 、 $t$  来存放输入数据及结果数据。
2. 通过 cin 为变量  $a$ 、 $b$ 、 $c$  输入数据。

3. 使用赋值语句来计算  $w$  与  $t$  的具体值。
4. 使用 `cout` 输出  $w$  与  $t$  的值。

### 【编程提示】

通常通过“`const double pai=3.1416;`”来定义 `double` 型的有名常量 `pai`，以表示常数  $\pi$  的一个近似值，常量 `pai` 的值在程序中不可改变。

## 1.3 自立题的求解提示与注意

### 一、自立题一

编写程序，使用 C++ 的输出语句，在屏幕上显示一周的课程表（或工作安排表）。

#### 【实验目的】

使屏幕输出的“信息”呈现某种所设定的样式（如课程表式样等）。

#### 【要点提示】

通过 `cout` 来进行输出，并使用转义字符“`\n`”、“`\t`”以及格式控制符“`endl`”等来控制输出式样。

### 二、自立题二

编写程序，计算如下 12 个连续自然数的累加和  $s$ ，并将结果显示在屏幕上。

$$s = 58 + 59 + 60 + \cdots + 69$$

进一步考虑如何累加更多项的数据。

#### 【实验目的】

通过使用表达式及赋值语句，计算  $s$  并输出。

体会在不使用循环语句的情况下，求出多项数据的累加和并不那么“方便”，尤其当要累加更多项的数据时（如从 58 累加到 158 等），就显得更为突出。

#### 【要点提示】

1. 可考虑使用如下两种赋值语句：

```
s = 58 + 59 + 60 + 61 + 62 + 63 + 64 + 65 + 66 + 67 + 68 + 69;
```

```
s = (58+69)*6; //在能够“提炼”出这一简洁计算公式的情况下使用
```

2. 在学习了“循环”之后，试着改进此求解方法。

### 三、自立题三

编写程序，任意输入 5 个整数，求出它们的累加和  $s$ ，并将结果显示在屏幕上。  
进一步考虑，在输入更多数据（如 100 个）时，又该如何处理。

#### 【实验目的】

通过说明若干个变量来存放输入数据，而后使用表达式及赋值语句，计算  $s$  并输出。

体会在不使用循环语句的情况下，求解该问题时的麻烦程度，尤其当具有更多输入数据的时候（例如有 100 个数据时），矛盾就更突出，也体现出后文学习编程技巧的重要性。

#### 【要点提示】

1. 可考虑使用如下两种方式来实现：

● 说明 5 个变量  $x_1$ 、 $x_2$ 、 $x_3$ 、 $x_4$ 、 $x_5$  来存放输入数据，最后通过 “ $s = x_1 + x_2 + x_3 + x_4 + x_5$ ；” 来实现累加。

● 说明 1 个变量  $x$  来存放输入数据，最后通过连续写 5 次 “ $\text{cin} \gg x; s = s + x;$ ” 来实现累加（注意，作为累加和的  $s$  变量的初值要设置为 0）。

2. 在学习了“循环”之后，试着改进此求解方法。

### 四、自立题四

设有一个底圆半径为  $R_1$ 、高为  $H_1$  的圆柱体，一个底圆半径为  $R_2$ 、高为  $H_2$  的圆锥体。编写程序，从键盘输入  $R_1$ 、 $H_1$ 、 $R_2$ 、 $H_2$ ，而后计算出这两个物体的底圆面积之和  $sa$  以及它们的体积之和  $sv$ ，并将结果显示在屏幕上。

#### 【实验目的】

编程求解具有输入输出的一般计算问题。

#### 【要点提示】

计算底圆面积的公式： $s = \pi R^2$ 。

圆柱体体积：底圆面积  $s$  乘以高  $H$ 。

圆锥体体积：底圆面积  $s$  乘以高  $H$ ，再除以 3。



## 第二章 基本数据类型与基本运算

### 2.1 示范题的求解与剖析

#### 一、示范题一

编写程序，对各表达式进行计算并输出指定结果。

(1) 计算下述各表达式的值并输出：

```
'A' + 3           '6'-48           'a'-'A'
2*'0' + 'a'/10    30/8 + 30%8 + 30/8.0    1>2 || 2>1
9>5 && 5>2        9>5>2           !!!25
false || !(1!=2 && false)
```

(2) 计算下列各表达式，并将表达式的值以及执行表达式后被改变的那些相关变量的值一并输出。注意，计算每个表达式前，总假定各相关变量的初值为：

```
int a=1, b=2, c=3; char ch='F'; double x=1.1;
x*+=a+19           (及改变后的变量 x)
c++*b+++*a++      (及变量 a、b、c)
++c*++b*++a       (及变量 a、b、c)
a=b--c+ch--++x    (及变量 a、b、c、ch)
c=(a=5,b+=3,a+b)  (及变量 a、b、c)
b!=c && a>c || b==1
```

#### 【实验目的】

熟悉表达式的求值规则以及各相关运算符的功能、优先级、结合性等。注意表达式求值计算的“副作用”，即表达式计算完成后，有可能同时改变了某些相关变量的当前值。

#### 【实现方法】

1. 通过使用 cout 和输出运算符“<<”将各表达式的值以及被改变的相关变量的值（若有的话）显示在屏幕上。

2. 注意：当要直接输出表达式“1>2||2>1”的值，使用“cout<<1>2||2>1;”时会出现语法错误。这是因为比较运算符“>”和“<”以及逻辑运算符“||”都比被重载了的移位运算符“<<”