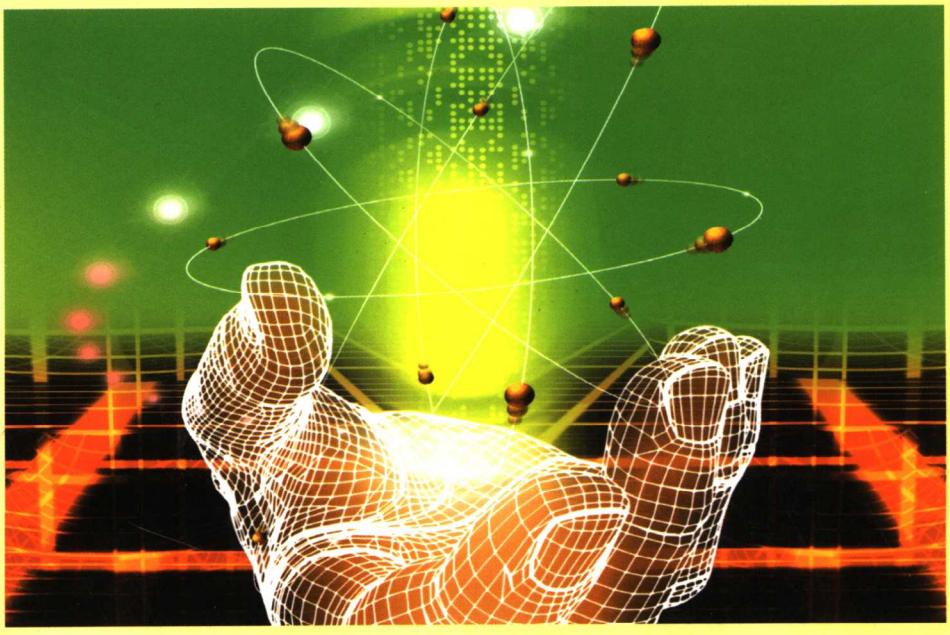
高等院校计算机课程设计指导丛书

C++语言课程设计

刘振安 刘燕君 孙 忱 编著

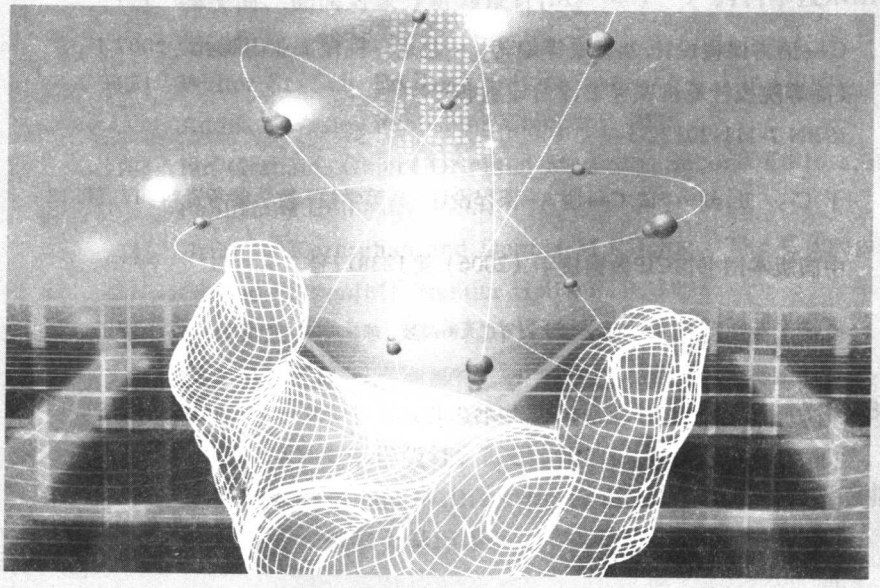


机械工业出版社
China Machine Press

高等院校计算机课程设计指导丛书

C++语言课程设计

刘振安 刘燕君 孙 忱 编著



机械工业出版社
China Machine Press

本书是一本独立于具体的C++语言教材的课程设计辅导，侧重于C++语言面向对象的基本特征，结合实际应用，涵盖C++语言程序设计中面向对象的基本特征，包括：熟悉编程环境和编程规范、动态存储管理和程序调试、多文件与菜单设计、使用组合与派生方法、对象启动程序、模板、循环链表、头文件等。书中给出的实例完整并经测试验证，有的设计还给出测试样例，循序渐进地启发学生完成设计，最后还结合课程设计和实际应用需要进行总结以拓宽知识面。

本书不仅适合作为高等院校相关专业C++语言程序设计课程的参考书，对广大工程技术人员也有很高的参考价值。

图书在版编目（CIP）数据

C++语言课程设计/刘振安等编著. －北京：机械工业出版社，2007.1
（高等院校计算机课程设计指导丛书）
ISBN 7-111-20122-1

Ⅰ.C… Ⅱ.刘… Ⅲ.C++语言－课程设计－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123817号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：朱　劼
北京京北制版印刷厂印刷·新华书店北京发行所发行
2007年1月第1版第1次印刷
186mm×240mm · 14.75印张
定价：25.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

高等院校
计算机课程设计指导丛书

专家指导委员会

（以姓氏拼音为序）

丛书序言

近年来，我国在计算机应用、计算机软件和电子类相关专业的人才培养方面，取得了长足的进展，每年的毕业生都有数十万人。但是这些毕业生走进企业、公司、政府机构或研究单位之后，往往深刻地感觉到缺乏实际开发设计项目的经验，不善于综合运用所学理论，对知识的把握缺乏融会贯通的能力。

综合考察目前高等院校教学大纲、课程设置以及内容安排等方面的情况，多数学校还是比较重视训练学生的实际设计能力。但是，从安排设计实践的内容上看，基本上是围绕相关课程教学内容而展开的，不能够构成对实际问题的解决方案；从配套程序的规模上看，一般只是几十行到几百行的源代码，或者是一个单独电路的设计，远远小于一个小型项目的规模；从设计的结构上看，由于设计实践是围绕着课程教学内容而进行的，问题已经高度抽象，学生很难得到有关综合运用所学知识的整体训练机会。而且，这些内容相对简单、问题域已经高度抽象、规模较小的设计实践一人基本上就能完成，学生几乎无法通过这些设计实践，去真正获得有关项目管理和团队协作等方面的基本训练和工作经验。

由此可以看出，大多数学校对学生实际设计能力的训练与国外知名大学和国内精品课程相比较，还是存在一些差距的。为此，机械工业出版社华章分社和一批高等院校的教师，针对当前高等院校计算机硬件、软件和电子类相关课程教学中存在的问题，参考国内外知名大学相关课程成功的教学经验，设计编写了这套“高等院校计算机课程设计指导丛书”，其目的就是通过课程设计的一系列训练，把知识获取和项目实践两个方面有机地结合起来。

在这套“高等院校计算机课程设计指导丛书”中的每一门课程设计里，都安排了由多个子项目组成的一个课程设计项目。学生们可以在教师的指导下，逐步设计实现这些子项目，并最终完成一个功能相对完整，可以运行的系统，其代码可以是数千行，甚至上万行。通过这种设计课程，学生一方面可以结合课程的教学内容循序渐进地进行设计方面的实践训练，另一方面，在参与一系列子项目的实践过程中，还能提高如何综合运用所学知识解决实际问题的能力，以及获得有关项目管理和团队合作等等众多方面的具体经验，增强对相关课程具体内容的理解和掌握能力，培养对整体课程知识综合运用和融会贯通能力。

参加丛书编写的各高等院校的教师都有着丰富的教学、科研，以及与企业合作开发项目等多方面的经验。每个课程设计中的子项目和整体项目，都来自教师们具体的科研和设计开发实践，所选设计项目与教学内容配合紧密，项目的难度与规模适宜。

最后，感谢机械工业出版社华章分社编辑们的大力支持，使出版有关这套丛书的计划，从单纯的构想演化成带有油墨芳香的真实。

丛书写作组

前　言

编程语言课程应注重边学边练，但由于课堂教学和实验的深度和广度有限，练习的深度也受到一定限制。为了弥补这一点，我们特编写了本书。

本书的主要特点如下：

1) 它独立于具体的C++语言教材，侧重讲述C++语言面向对象的基本特征，以“不变”应“万变”，涵盖C++语言的重要基础知识。

2) 结合实际应用的要求，既覆盖知识点，又接近工程实际需要。通过激发学生的学习兴趣，调动学生主动学习的积极性，来引导他们根据实际要求完成编程，训练其实际分析问题的能力及编程能力，并养成良好的编程习惯。

3) 通过详细的实例，循序渐进地启发学生完成设计。本书将要求、算法和源程序分开，为学生创造独立思考的条件。学生在透彻理解要求和算法的前提下，完全可以不按书中提供的参考程序，设计自己的应用程序。

4) 为同一类型的设计题目提供不同的解决方案，以拓宽学生的视野。

5) 课程设计分为基本部分与扩展部分，以满足不同学校和不同学生的要求。

6) 提供综合课程设计，以进一步锻炼学生使用面向对象方法思考问题的能力及动手能力。这些综合实验还可以供学生分工合作，以培养团队协作精神。

7) 对课程设计题目和实际应用的结合进行总结，进一步拓宽知识面。

另外，在实际编程中，为了提高编程质量，对空行、空格和注释均有要求。本书也尽可能根据实际编程要求给出空行、空格和注释，有时因为标题和页码等实际原因，也会适当减少空行、空格和注释，但希望读者在编写代码时，严格按要求处理，以建立良好的编程风格。

本书共分13章。第1章是概述；第2章介绍编程环境和编程规范；第3章介绍动态存储管理和程序调试；第4章介绍多文件与菜单设计；第5章阐述如何通过组合构成新的类；第6章说明通过派生构成新的类的方法；第7章介绍使用对象启动程序；第8章描述如何使用模板并测试程序；第9章介绍设计循环链表和文件；第10章设计供他人使用的头文件；第11章说明使用链表和文件；第12章介绍如何使用向量和文件；第13章对课程设计进行了总结。

我原先所写的《C++程序设计课程设计》一书已被全国许多院校选用，有的学校还将其用作毕业设计的参考资料，均获得可喜成绩。为了满足不同学校的教学需求，又编写了这本书，这本书更注重基础训练。在编写这两本书时，得到许多学校师生的支持和帮助，对他们表示感谢！本书还被许多工程技术人员用作参考书并给出很好的反馈意见，

特此感谢。

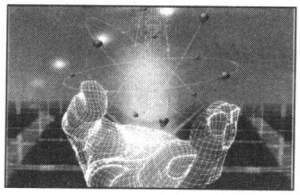
参加本书编写的还有刘燕君和孙忱等，她们放弃暑假休息，不仅参与编写，还仔细调试程序，逐字逐句校对，为本书的出版付出了大量心血。

由于笔者学识有限，书中难免有疏漏之处，请读者指正。

刘振安

中国科学技术大学

目　录



第1章 概　述

由于各院校的情况不同，为了便于教师根据本校的特点和教学计划选择相应的课程设计内容，本章将简要介绍本书的设计题目及其预期目标。

1.1 课程设计目标

一般来讲，课程设计比教学实验复杂一些，深度更广并更加接近实用。目的是通过课程设计的综合训练，培养学生实际分析问题、编程和动手能力，最终帮助学生系统掌握课程的主要内容，更好地完成教学任务。

本书侧重基础训练，通过课程设计的题目明确训练的内容，训练由易而难，逐步深入。有些题目贯穿全书始终，通过不同的方法对其求解（例如出圈游戏），从而让读者体会面向对象编程的风格和特点。本书强调主程序的编制方法，加深对通过类的对象启动程序的理解。

最后两个课程设计规模较大，分别使用链表和向量实现。在教学中，教师可以根据情况选择其一或全部。这两个课程设计均可以扩充，便于分成几个小项目供学生分工合作，以培养团队协作精神。

1.2 课程设计结构

结构化程序设计使用的是功能抽象，面向对象程序设计不仅能进行功能抽象，而且能进行数据抽象。“对象”实际上是功能抽象和数据抽象的统一。C++语言的“对象”是“类”的实例，程序设计的基础是设计类，所以类的有关概念都是重点，尤其要抓住抽象、封装、继承和多态性等要素。

设计类的重点是选择数据成员和成员函数。成员函数设计的难点是选择函数类型及其参数传递方式。数据类型及程序控制方式仍然是C++语言的基础，数组、指针、类和结构的使用技术是编程的核心技术。学生学习时，常常避开多文件编程和使用文件，但这些都是程序设计员必备的知识，因此本书的课程设计特意加强了这方面的训练。

学习面向对象编程的另一个难点是如何选择一个类，通过这个类的对象用自己的成员函数完成一系列的动作，实现最终目的。为此，本书也特别注意这方面的训练。

本书共有11个设计题目和一个课程设计总结。课程设计不使用一个独立的例子涵盖这些知识点，而是按层次逐步深入。为了使学生理解它们之间如何相互配合，设计要求

使用接近实际需要的方式编程。下面简要说明这些设计题目:

熟悉编程环境和编程规范（第2章）

这一章的目的是熟悉编程环境和编程规范，所以通过给出类的声明，要求读者分别按是否有头文件两种情况来实现程序，从而熟悉编程规范，为以后的学习打下基础。

该章的设计还涉及编制函数及参数传递方式的知识，这些都是编程的基本功。

动态存储管理和程序调试（第3章）

动态内存分配是学习的难点之一，该章将通过为类对象进行动态内存分配演示使用指针的技巧，并熟悉跟踪程序和观察程序中变量和对象值的方法，提高程序调试技能。

这一章仍然使用一个头文件和一个源文件的方法，以便使读者进一步巩固头文件的设计方法并熟悉编程环境及规范。

多文件与菜单设计（第4章）

该章通过设计游戏程序介绍菜单设计方法，并进一步介绍多文件设计和正规的头文件设计方法。

该章使用两种方法设计游戏程序。第一种方法是使用动态数组，目的是进一步加强对动态内存管理的认识。第二种方法是使用向量数组，目的是了解它与普通数组的异同。为了熟悉函数重载的使用方法，这两个游戏程序使用相同的函数名。该章的设计实际上是两个题目，教师可以根据教学要求取舍。

通过组合构成新的类（第5章）

该章的设计任务是使用组合的方法设计新类。这一章给出两个题目，一个题目是使用一个Point类产生Line类，但同时要求给出使用模板实现的程序。另一个题目是使用组合构成新类的出圈游戏。

通过这个项目，目的是使读者掌握在不同的实现方法中，如何设计相应的构造函数和复制构造函数，进一步理解程序如何调用它们及析构函数的执行顺序。

该章的设计实际上是两个题目，教师可以根据教学要求取舍。

通过派生构成新的类（第6章）

该章的设计任务是使用派生的方法设计新类。这一章给出两个题目，一个题目是使用一个Point类派生Line类，但同时要求给出使用模板实现的程序。另一个题目是使用派生构成新类的出圈游戏。

通过这些项目的练习，使学生掌握在不同的实现方法中如何设计相应的构造函数和复制构造函数，进一步理解程序如何调用它们及析构函数的执行顺序。

该章的设计实际上是两个题目，教师可以根据教学要求取舍。

使用对象启动程序（第7章）

前面几章主要是设计特定的函数，这些函数使用类作为参数完成预定任务。这一章的目的是产生一个类的对象，通过对象自己的函数成员完成设计任务。

该章的设计项目是设计职工信息表，并由此产生一个信息简表。要求使用数组，利

用赋值兼容规则实现简表，并使用虚函数实现多态性，完成显示不同简表信息的任务。

该章还给出使用菜单和向量实现的设计方案，以便读者进一步理解向量的使用方法，这个要求可以作为选做项目，教师也可以根据教学需要加以取舍或增加新的要求。

使用模板并测试程序（第8章）

该章的设计任务是定义一个计算器模板类，模拟后缀表达式的计算过程，目的是进一步熟悉模板及模板之间的关联方法。设计本身并不难，但涉及堆栈的知识，所以该章还会简要介绍测试与调试的知识，希望增加测试要求，进一步理解后缀表达式及堆栈的应用。

设计循环链表和文件（第9章）

该章将使用循环链表再次设计出圈游戏程序，目的是让读者接触简单的链表概念并进一步熟悉面向对象的程序设计思想。另外，该章还涉及简单的读取文件操作，以便理解字符流的概念。

设计供他人使用的头文件（第10章）

该章的目的是设计一个头文件，使用户可以像使用系统头文件那样，通过包含它就可以使用在头文件中定义的功能。为此，该章设计一个有理数(Rational)类的头文件，供用户来求解有理分式方程。

这个类的实现涉及大量的运算符重载，包括四则运算、关系运算和输入输出流等。通过这个设计，读者可以更加理解重载的重要性。

使用链表和文件（第11章）

该章的重点是使用链表和文件操作，同时进一步熟悉运算符重载。一般来说，编制实用程序都离不开文件存取和运算符重载，该章的内容应该给予足够的重视。

该章将使用链表实现学生成绩的统计、查询、删除以及文件的存取等操作。

使用向量和文件（第12章）

该章要求设计一个实用的小型学生成绩管理程序，它要求不用链表，而是用向量来设计这个程序。该程序有查询和检索等功能，并且能够对指定文件操作，也可将多个文件组成一个文件。

该章涉及的知识面较多，由于各校教学要求不一样，教师也可以根据实际教学情况决定取舍。

因为向量的内容比较好理解，前面的设计也使用过它，所以建议学生将本课程设计列为必做或自学的内容，以便增加对STL库的了解。读者也可以在熟悉这个程序的基础上，改用数组来实现这个设计。

课程设计总结（第13章）

在实际的应用中，重点是保证程序的设计质量。只有掌握头文件和多个源程序文件的编制及制作工程文件等方法，才能提高实际应用的能力。该章结合前面的设计实例和具体的Visual C++ 6.0编程环境，有的放矢地总结使用C++语言设计面向对象实用程序时

经常用到的主要技术。

1.3 评价标准

因为本书提供参考程序，所以一般情况下读者都能完成预定设计。如果学生只是按照程序去做，其分数只能在85分以下。为了证明学生已经掌握设计所涵盖的知识点，应该向学生提一些问题，例如程序如何实现及其原理等问题。考虑到各校情况不一，本书没有在各章的评价中规定必须提问题才算完成设计任务。

另外，程序的可读性均作为正确性的一个方面进行记分，不再单列考核标准。

一般遵循如下规律加以评价:

1) 严格控制90分，其标准是有创意。

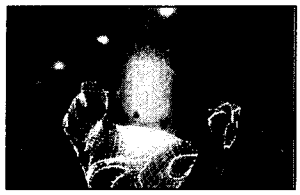
2) 程序必须全部正确，并有一定改进或者能正确回答设计中的问题才能给予85分以上的成绩。

3) 有少许失误可给75 ~ 83分。

4) 错误不多可给60 ~ 73分。

5) 没有完成规定的要求，则不予及格。

因为课程设计主要是锻炼学生的探索能力，所以应该鼓励他们不要将这项工作作为负担，以便提高钻研问题的兴趣，放手去做。另外，评分标准也可以只设“通过”、“没通过”和“优秀”3档以激发学生的学习兴趣。



第 2 章
熟悉编程环境和编程规范

学习C++语言，首先要熟悉编程环境,以便能熟练使用编程环境编制和调试程序,得到正确的可执行文件。另外，应熟悉编程规范，并在实际编程中不断加深认识，不断积累经验，才能提高自己的实际编程能力。

本章的设计还涉及编制函数及参数传递方式的知识，以便为以后的编程打下基础。

2.1 设计要求

本课程设计的目的是熟悉编程环境和编程规范，所以本设计给出类的声明，读者只需要按要求实现类，并根据是否有头文件来实现程序。

1. 类的设计要求

下面是一个bird类:

```
class bird{
      private:
          int IsFly;
      public:
          bird(int=0);
          int Get();
};
```

如果IsFly为0，表示是一只会飞的鸟，例如喜鹊。如果IsFly为1，表示是一只不会飞的鸟，例如企鹅。

在程序中实现这个类。

2. 主函数的设计要求

主函数的要求如下:

```
int main()
{
      bird bird1(0), bird2(1), bird3;
      bird bird4(4);

      answer(bird1);
```

```
    answer(bird2);
    answer(bird3);
    answer(bird4);
    return 0;
}
```

3. answer函数的设计要求

根据对象的属性给出是否能飞的答案。如果能飞，输出“I can fly!”；否则输出“I can not fly!”。如果IsFly不是0或者1，则输出“I do not know it!”。

4. 单文件构成方式

工程名为bird，整个程序使用一个文件bird.cpp实现。

5. 两个文件的构成方式

要求如下：

1) 工程名为bird。

2) 将类和answer函数的原型声明放在头文件bird.h中。

3) 程序文件为bird.cpp。

6. 演示两种程序的构造过程

使用Visual C++ 6.0产生项目和相应文件，编制并运行程序。

2.2 类的实现

在类的外部实现成员函数的定义。

1. 构造函数

首先读懂给出的函数原型

```
bird(int=0);
```

的含义。这是具有默认参数的构造函数，默认值是IsFly=0，即一般的鸟会飞。主程序中

```
bird bird4;
```

创建的鸟bird4是一种会飞的鸟。使用参数列表的构造函数定义如下：

```
bird::bird(int a):IsFly(a)
{}
```

也可以写成如下的等效形式：

```
bird::bird(int a)
{IsFly=a;}
```

2. 成员函数

成员函数是属于类的，即“bird::Get()”。成员函数的返回类型是int，应返回类的IsFly属性。由此可以写出它的定义：

```
int bird::Get()
{ return IsFly;}
```

2.3 函数answer的设计

根据主程序使用answer函数的方法，可以写出函数的原型。因为不需要answer函数返回值，所以参数可以使用传值的方式。但函数需要判别对象的属性IsFly是否合乎要求。如果IsFly的值既不是0，又不是1，就要另行处理。可以写出函数的原型如下：

```
void answer(bird);
```

函数的定义如下：

```
void answer(bird a)
{
     int x=a.Get();
     if(x<0||x>2)
     {
          cout<<"I do not know it! "<<endl;
          return;
     }

     if(x==0)
          cout<<"I can fly!"<<endl;
     else
          cout<<"I can not fly!"<<endl;
}
```

虽然可以在判断语句中直接使用成员函数Get，例如：

```
if(a.Get()==0)   cout<<"I can fly!"<<endl;
```

但是函数中多次使用成员函数会降低程序效率，所以先将它的值赋给变量x，然后使用x进行判别。

answer函数的另一种参数传递方式是传引用。它的函数原型是

```
void answer(bird&);
```

在多文件结构中，将使用传引用的函数定义。

2.4 使用单文件构成模式

单文件构成模式是指程序只使用一个后缀为cpp的文件。这种模式是将类的定义和预处理命令以及其他函数原型声明放在主函数之前，然后书写主函数及其他函数。下面将该文件分为三个部分，这也是一般的程序结构。

第1部分：预处理命令
　　类的声明和定义
　　其他函数原型声明
第2部分：主函数
第3部分：其他函数

采用这种结构的好处是，不必考虑其他函数的定义的先后顺序。当然，也可以将主函数放在最后。