



高职高专计算机教育教材研究与编审委员会推荐
21世纪高职高专计算机课程精选教材

全 新 版

C++ 程序设计 基础教程

张军安 编



西北工业大学出版社

21 世纪高职高专计算机课程精选教材

C++程序设计基础教程

张军安 编

西北工业大学出版社

【内容提要】本书为高职高专计算机课程教材。全书共分为 14 章, 主要介绍 C++ 发展史、基本数据类型与表达式、结构控制语句、指针与数组、函数、类与对象、继承与派生、运算符重载、模板、输入/输出流、异常处理。书中配有大量典型实例, 并配有实训, 通过上机操作使读者在学习 C++ 语言的过程中更加得心应手, 做到学以致用。

本书不仅适合高职高专学校作为教材使用, 同时也可以作为各类培训班及高等院校计算机专业的基础课程教材。

图书在版编目 (CIP) 数据

C++ 程序设计基础教程/张军安编. —西安: 西北工业大学出版社, 2005.10

ISBN 7-5612-1986-5

I. C… II. 张… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 091818 号

出版发行: 西北工业大学出版社

通信地址: 西安市友谊西路 127 号 邮编: 710072 电话: 029-88493844 88491757

网 址: www.nwpup.com

印刷者: 陕西友盛印务有限责任公司印刷

开 本: 787 mm×1 092 mm 1/16

印 张: 20.5

字 数: 527 千字

版 次: 2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷

定 价: 27.00 元

21 世纪高职高专计算机教材研究与编审委员会

名 单

(排名不分先后)

主任委员： 夏清国

副主任委员： 刘培奇 刘 晔 刘 黎 刘鹏辉

委 员： 罗 军 任绍辉 孙姜燕 黄伟敏

 韩银锋 封 磊 杨卫社

主 编： 张军安 王 璞

参编人员： 张建林 兰 鑫 赵智勇 高 红

 李学军 马小娟 刘 睿 闫晓敏

 周永红 李 帅

序 言

21 世纪是信息时代,是科学技术高速发展的时代,为了让数以亿计的人跟上时代的步伐,提高全民族的竞争力,积极发展高职高专教育,完善职业教育体系,是我国职业教育改革和发展的一项重要工作。

高等职业教育有其自身的特点。正如教育部“面向 21 世纪教育振兴行动计划”所指出的那样,“高等职业教育必须面向地区经济建设和社会发展,适应就业市场的实际需要,培养生产、管理、服务第一线需要的实用人才,真正办出特色。”因此,不能以本科压缩和变形的形式组织高等职业教育,必须按照高等职业教育的自身规律组织教学体系。为此,我们根据高等职业教育的特点及社会对教材的普遍需求,组织高等职业院校有丰富教学经验的老师,编写了这套《21 世纪高职高专计算机课程精选教材》。

本套教材充分考虑了高等职业教育的培养目标、教学现状和发展方向,在编写过程中突出了实用性。本套教材重点讲述目前在信息技术行业实践中不可缺少的知识,并结合具体实践加以介绍。大量具体操作步骤、众多实践应用技巧、接近实际的实训材料保证了本套教材的实用性。

在本套教材编写大纲的制定过程中,我们广泛收集了高等职业学院的教学计划,调研了多个省市高等职业教育的实际情况,经过反复讨论和修改,使编写大纲能最大限度地符合我国高等职业教育的要求,切合高等职业教育实际情况。

在选择作者时,我们特意挑选了在高等职业教育一线的优秀骨干教师。他们熟悉高等职业教育的教学实际,并有多年的教学经验;其中许多是“双师型”教师,既是教授、副教授,同时又是高级工程师、认证高级设计师;他们既有坚实的理论知识,很强的实践能力,又有较多的写作经验及较好的文字水平。

本套教材是高等职业学院、高等技术学院、高等专科学院计算机课程精选教材,适用于信息技术的相关专业,如计算机应用、计算机网络、信息管理、电子商务、计算机科学技术、会计电算化等,也可供优秀职高学校选作教材。对于那些要提高自己的应用技能或参加证书考试的读者,本套教材也不失为一套较好的参考用书。

最后,希望广大师生在使用过程中提出宝贵意见,以便我们在今后的工作中不断地改进和完善,使这套教材成为高职高专的精品教材。

21 世纪高职高专计算机教材研究与编审委员会

前 言

C++语言是面向对象程序设计语言中应用最为广泛的一种语言。面向对象程序设计方法是将数据和处理数据的过程作为一个封装体。它具有封装性、数据隐藏性、继承重用以及多态性等优点，目前已经成为开发大型应用软件的主要工具。

《C++程序设计基础教程》主要讲述了 C++的发展史、基本语法、基本算法、面向对象程序设计的方法及其特性、模板、流、异常处理等知识点。

本书思路新颖、图文并茂，不仅适用于高职高专的学生，也可作为各类培训班学生 C++基础课程的首选学习用书，同时也可作为各类用户的参考工具书。



本书主要内容为：

- ▶ C++的发展与特点概述
- ▶ 数据类型与表达式
- ▶ 程序流程控制语句
- ▶ 数组、指针与引用
- ▶ 函数
- ▶ 结构体与共用体
- ▶ 类与对象
- ▶ 继承与派生
- ▶ 模板
- ▶ 多态性与虚函数
- ▶ 运算符重载
- ▶ C++的 I/O 流
- ▶ 异常的处理
- ▶ 综合实例
- ▶ 实训

由于编者水平有限，疏漏之处在所难免，希望广大读者批评指正。

编 者

目 录

第 1 章 C++的发展与特点概述	1
1.1 C++与 C 语言	1
1.1.1 C++的产生	1
1.1.2 C++与 C 语言的关系	1
1.2 C++的新特性	2
1.2.1 面向对象的概念	2
1.2.2 面向对象的特点	3
1.2.3 C++程序的组成	3
1.3 Visual C++6.0 开发环境	4
1.3.1 Visual C++6.0 开发环境的启动	4
1.3.2 一个简单的 C++程序	6
本章小结	7
习题一	8
第 2 章 数据类型与表达式	9
2.1 基本数据类型	9
2.1.1 整数类型	9
2.1.2 浮点类型	10
2.1.3 字符类型	10
2.1.4 逻辑类型	10
2.1.5 空值类型	10
2.2 常量	11
2.2.1 整型常量	11
2.2.2 浮点型常量	11
2.2.3 字符型常量	12
2.2.4 字符串常量	12
2.2.5 逻辑常量	13
2.2.6 地址常量	13
2.3 变量	13
2.3.1 变量的定义	13
2.3.2 变量赋初值	15
2.4 运算符与表达式	15
2.4.1 算术运算符	15

2.4.2 关系运算符	16
2.4.3 逻辑运算符	17
2.4.4 位运算符	18
2.4.5 特殊运算符	19
2.4.6 运算符小结	20
2.4.7 类型转换	21
2.5 自定义数据类型	22
本章小结	22
习题二	23

第3章 程序流程控制语句

25

3.1 顺序结构语句	25
3.1.1 表达式语句	25
3.1.2 输入输出语句	25
3.1.3 预处理命令语句	27
3.2 选择结构语句	28
3.2.1 if 语句	28
3.2.2 switch 语句	31
3.3 循环结构语句	33
3.3.1 while 语句	33
3.3.2 do...while 语句	35
3.3.3 for 语句	36
3.4 跳转语句	39
3.4.1 continue 语句	39
3.4.2 goto 语句	40
本章小结	40
习题三	41

第4章 数组、指针与引用

43

4.1 数组	43
4.1.1 一维数组的定义与用法	43
4.1.2 二维数组的定义与用法	47
4.2 指针	51
4.2.1 指针的概念	51
4.2.2 一级指针的定义与用法	51
4.2.3 二级指针的定义与用法	53

4.3 指针与数组	54
4.3.1 指针与一维数组	54
4.3.2 指针与二维数组	55
4.4 引用	56
4.4.1 引用的概念	57
4.4.2 引用的用法	57
本章小结	58
习题四	58
 第 5 章 函数	62
5.1 函数的定义	62
5.1.1 声明函数	62
5.1.2 定义函数	62
5.2 函数的调用	63
5.2.1 函数的值传递	64
5.2.2 函数的地址传递	65
5.2.3 函数在排序算法中的应用	65
5.3 函数的重载	69
5.3.1 参数类型不同的重载函数	69
5.3.2 参数个数不同的重载函数	70
5.4 函数的嵌套调用与递归调用	71
5.4.1 函数的嵌套调用	71
5.4.2 函数的递归调用	72
5.5 内联函数	74
5.6 数据存储类型	76
5.6.1 自动存储类型	76
5.6.2 静态存储类型	77
5.6.3 外部存储类型	78
5.6.4 寄存器存储类型	79
5.7 字符串处理函数	80
5.7.1 strlen 函数	80
5.7.2 strcat 函数	80
5.7.3 strcmp 函数	81
5.7.4 strcpy 函数	81
5.7.5 strstr 函数	82
本章小结	82
习题五	83

第 6 章 结构体与共用体	88
6.1 结构体的定义.....	88
6.2 结构体的用法.....	89
6.2.1 结构体成员的引用.....	89
6.2.2 结构体数组.....	89
6.2.3 结构体与指针.....	91
6.3 共用体.....	92
本章小结.....	92
习题六.....	92
第 7 章 类与对象	94
7.1 类.....	94
7.1.1 类的概念.....	94
7.1.2 类的定义.....	95
7.2 对象.....	98
7.2.1 对象的定义.....	98
7.2.2 对象的操作.....	99
7.2.3 this 指针.....	101
7.2.4 对象数组.....	102
7.3 构造函数.....	103
7.3.1 构造函数的定义.....	103
7.3.2 构造函数的重载.....	105
7.3.3 拷贝构造函数.....	106
7.4 析构函数.....	108
7.5 静态成员.....	109
7.5.1 静态数据成员.....	109
7.5.2 静态成员函数.....	110
7.6 友元.....	112
7.6.1 友元函数.....	112
7.6.2 友元类.....	113
本章小结.....	114
习题七.....	115
第 8 章 继承与派生	118
8.1 基类与派生类.....	118

8.2 继承的方式	118
8.2.1 公有继承 (public)	119
8.2.2 私有继承 (private)	120
8.2.3 保护继承 (protected)	122
8.3 单继承	123
8.3.1 成员访问权限的控制	123
8.3.2 派生类与构造函数和析构函数	126
8.4 多继承	132
8.4.1 多继承的概念	132
8.4.2 多继承的应用	134
8.5 赋值兼容原则	137
本章小结	138
习题八	138
 第 9 章 模板	 141
9.1 模板的概念	141
9.2 函数模板	142
9.2.1 函数模板的定义	142
9.2.2 函数模板的用法	143
9.2.3 模板函数的重载	144
9.3 类模板	145
9.3.1 类模板的定义	145
9.3.2 类模板的用法	147
本章小结	150
习题九	150
 第 10 章 多态性与虚函数	 153
10.1 多态性	153
10.1.1 多态性概述	153
10.1.2 多态的分类	153
10.2 虚函数	155
10.2.1 一般虚函数	156
10.2.2 虚函数的继承性	159
10.2.3 虚函数与重载函数	160
10.2.4 虚析构函数	162
10.2.5 构造函数调用虚函数	163

10.3 纯虚函数与抽象类	165
10.4 虚基类	169
10.4.1 虚基类的引入	170
10.4.2 虚基类的构造函数	172
本章小结	174
习题十	174
 第 11 章 运算符重载	 176
11.1 运算符重载的概念	176
11.2 运算符重载的用法	177
11.2.1 实现运算符重载的非成员函数形式	177
11.2.2 实现运算符重载的成员函数形式	179
11.3 特殊运算符的重载	181
11.3.1 自增运算符的重载	181
11.3.2 强制类型转换运算符的重载	187
11.3.3 赋值运算符的重载	188
本章小结	191
习题十一	191
 第 12 章 C++的 I/O 流	 194
12.1 流的概念	194
12.2 格式化输入输出	195
12.2.1 输出宽度设置函数	195
12.2.2 补位填充函数	195
12.2.3 输出精度设置函数	196
12.2.4 输出数据状态设置函数	197
12.2.5 数制设置函数	198
12.3 文件流	199
12.3.1 文件的概念	199
12.3.2 文件流对象	199
12.3.3 文件流状态的判断	201
12.3.4 文本文件的访问	201
12.3.5 二进制文件的访问	208
12.4 字符串流	214
本章小结	215
习题十二	215

第 13 章 异常的处理	219
13.1 异常处理的概念	219
13.2 如何实现异常处理	219
13.3 关于异常处理的若干规定	221
13.4 异常处理的综合应用	223
本章小结	228
习题十三	228
第 14 章 综合实例	231
实例 1 同学录管理系统	231
实例 2 学生成绩管理系统	252
本章小结	284
实训	285
实训 1 数据类型与表达式	285
实训 2 顺序结构语句	286
实训 3 选择结构语句	287
实训 4 循环结构语句	288
实训 5 数组	289
实训 6 函数	291
实训 7 类与对象	292
实训 8 继承与派生	294
实训 9 多继承	296
实训 10 模板	298
实训 11 多态性与虚函数	299
实训 12 运算符重载	301
实训 13 文件流	303
附录	306
附录 I C++语言常用语法概要	306
附录 II 常用字符与 ASCII 码对照表	312
附录 III 运算符和结合性	313

第 1 章 C++的发展与特点概述

C++语言是当今世界功能最强大、应用最广泛的程序设计语言之一。它是在原有 C 语言的基础上发展起来的，与 C 语言相比增加了类的概念。类是一组对象的统一描述体。数据用来描述对象的状态，函数用来描述对象的行为，类是数据与函数的结合体，而 C 语言中的结构仅仅是数据的集合。

本章主要内容：

- ◆ C++与 C 语言
- ◆ C++的新特性
- ◆ Visual C++6.0 开发环境

1.1 C++与 C 语言

C++与 C 语言有着密切的联系。C++是在 C 语言的基础上发展起来的，C++继承了 C 语言的优点，极大地扩充了 C 语言的功能。C++包括 C 语言的全部特征、属性和优点，既面向过程又面向对象，对面向对象完全支持。

1.1.1 C++的产生

1972 年美国的 Dennis Ritchie 教授设计发明了 C 语言，并首次应用在 UNIX 系统上，此后出现了许多 C 语言版本，它们之间由于没有统一的标准，因此存在许多不同之处，使程序的可移植性受到很大的限制，制约了其发展和推广。美国标准研究所 ANSI (American National Standard Institute) 为了改进此现状制定了一套 ANSI 标准，也就是现行的 C 语言标准。

随着计算机应用领域的扩大，计算机软件系统的规模成几何型递增，给程序编写者带来极大的压力，因此许多专业人士都在探索，是否有一种方法可以大幅度提高代码的重用性，并能够事例化一些具体的事务。

1980 年贝尔实验室 Bjarne Stroustrup 教授开始对原有 C 语言进行了扩充和改进，引入了类(class)的概念。1983 年正式给这种带类的 C 命名为 C++。以后经过 3 次修订终于在 1994 年制定了 ANSI C++ 标准草案。C++继承了 C 语言功能强大、使用灵活、运算符和数据结构丰富以及使用结构化编程语言等特点。同时增加了面向对象编程 (OPP) 的完全支持，并最终成为应用最广泛的编程语言之一。

1.1.2 C++与 C 语言的关系

C++是 C 语言的一个超集。C++不但可以兼容所有的 C 语言语句，而且具有自己的新特性。如类、对象、面向对象程序设计以及对输入输出语句的改进，因此，把 C++称为 C 语言的一个超集，而 C 语言仅仅是 C++的一个子集。其关系如图 1.1.1 所示。

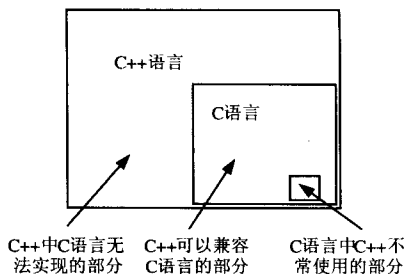


图 1.1.1 C++与 C 语言的关系

1.2 C++的新特性

C++不但可以完全兼容 C 语言的编程规则，而且还在原有 C 语言的基础上引入了类的概念，使其成为真正的面向对象程序设计语言。

1.2.1 面向对象的概念

面向对象程序设计相对于结构化程序设计增加了许多新的概念。透彻理解和掌握这些概念是学习面向对象程序设计思想最基本的要求，本小节将重点讲解对象、类和消息。

1. 对象

在现实世界中，可以将客观存在的任何实体看做一个对象，如一个家庭、一个学校、一个社会、一个人、一粒沙子等。各个对象之间都存在着一定的不同点，也正是这种不同点使我们很容易区分每一个对象，同时它们之间也存在着联系，虽然有些联系看起来很牵强。如一个家庭与一个社会，一个个家庭组成了一个社会大家庭。

对象的属性有静态的如一个人的性别，从出生开始其性别就是不可改变的。还有一些属性是动态的，即随着条件的变化对象的这种属性也在变化。如一个人的身高随着年龄的增长，身高也在不断的增长。动态属性与静态属性之间在一定条件下可以转换，如人的身高在达到一定年龄时，就成为了一个定值，即人的身高作为动态属性转换成静态的属性。

现实世界具体实体抽象化便构成了计算机世界里的对象，即抽象的对象。如利用计算机画板工具画一幅画，这幅画也属于一个对象，具有一定的属性，但只能看到，摸不着。还有一些更为抽象的对象如数据结构、计算机算法中的进制等。

现实世界中的对象和抽象的对象存在一定的联系，如把利用计算机画的一幅画打印出来，抽象的一幅画就变成实实在在的一幅画了。

总之，无论是现实世界的对象，还是计算机中抽象的对象。它们都具有以下特点：

- (1) 每个对象的名称具有惟一性，用于惟一标志一个对象。
- (2) 每个对象都包含一组操作，每个操作与对象的一种行为或功能相对应。
- (3) 每个对象都包含一组状态，对象的特征由它们来描述。
- (4) 每个对象所包含的操作可能是自身的也可能是外来的。

2. 类

类是一组具有相同属性和方法的对象的集合，即定义了其所属对象的共有属性与方法，而对象是类的一个实例。

3. 消息

消息用于实现对象与对象之间数据共享和信息传递。消息具有下面 3 个性质：

- (1) 同一个对象可以接受多条不同的消息。
- (2) 同一条消息可以发送给多个不同的对象。
- (3) 消息可以发送给任意对象，但对象可以不响应消息。

1.2.2 面向对象的特点

C++是一种真正意义上的面向对象的编程语言，具有面向对象编程语言的共性。

1. 封装性与隐蔽性

类是实现数据封装的工具，而对象是数据封装的实现。用户可以通过自定义类型支持数据的封装与隐蔽。在 C++面向对象程序设计中，数据与对数据的操作封装在一起，形成一个类。对象是一个具体类的变量。每个类可以包含自定义的公有成员 (public)、私有成员 (private) 和保护成员 (protected)。类建好后，使用者不必知道类中数据之间算法的具体实现，只须知道用它可以做什么即可。

2. 继承与重用性

C++语言允许在一个类的基础上声明一个新类，这种操作称为继承与重用，该类被称为基类，声明的新类被称为派生类。派生类可以继承其基类的公有成员与保护成员，作为自己的成员使用。

3. 多态性

所谓多态性是指当类的一个对象接受到一条信息时，对象的表现是动态的、可变的。

1.2.3 C++程序的组成

一个完整的 C++程序包含预处理命令、表达式语句、函数、输入输出语句和注释语句等。

例 1.1 C++程序的应用。

程序

```
#include<iostream.h>
void main()
{
    int a=0;           //定义一个整型变量 a 并给其赋初值 0
    a=a+1;             //把 a 的值加 1 后的结果重新赋给 a
    cout<<a<<endl;    //输出 a 的值
}
```

1. 预处理命令

预处理命令包括：宏定义命令、文件包含命令和条件编译命令。预处理命令是以“#”开头的，例 1.1 中的预处理命令属于文件包含命令。关于预处理命令的使用方法将在第 3 章详细介绍。

2. 表达式语句

C++程序的基本单位是语句，程序是按语句编译的。“;”是语句的结束标志符。只有语句结束标志符，没有表达式的语句，称为空语句。空语句也参与编译，但不实现任何操作。

3. 函数

C++程序中的函数包括两种：一种是主函数，即用关键字 `main` 修饰的函数；另一种是子函数，即由用户编写的除系统函数外的函数。主函数可以调用任何一个子函数，子函数之间也可以相互调用，但子函数不能调用主函数。

函数的定义格式为：

<函数返回值类型><函数名>()

函数的返回值类型可以是任意基本数据类型；函数名用于标志一个函数，使函数具有惟一性。

4. 输入输出语句

输入输出语句用于实现人机交互，即人与计算机之间相互传递信息。具体使用规则将在第 12 章详细介绍。

5. 注释语句

注释语句是为了增加源程序的可读性，但不参与程序编译。

注释有两种形式：一种是块注释，其标志符是“/*”与“*/”，处在两个标志符之间的部分是程序的注释；另一种是行注释，其标志符是“//”，处在该标志符后的部分是注释。

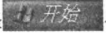
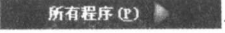
注意：注释部分在 Visual C++6.0 开发环境中以绿色显示。

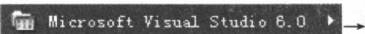
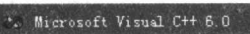
1.3 Visual C++6.0 开发环境

C++开发环境有多种版本，目前最流行的有 Borland 公司（Turbo C 的开发者的）开发的 BorlandC++ 和微软公司开发的 Microsoft Visual C++6.0 版本。

本节将以 Microsoft Visual C++6.0 版本为例介绍 C++程序的开发流程。

1.3.1 Visual C++6.0 开发环境的启动

启动 Visual C++6.0 开发环境，通常可采用两种方式：一是选择  开始 →  所有程序 (P) →

 Microsoft Visual Studio 6.0 →  Microsoft Visual C++ 6.0 命令；二是首先在桌面上创建 Visual C++6.0



的快捷方式，然后双击  图标。采用以上任意两种格式即可打开如图 1.3.1 所示的 Visual C++6.0 的工作界面。