基础教育系列

21世纪高校计算机应用技术系列规划教材

丛书主编　谭浩强

C++面向对象程序设计习题解答与上机指导（第二版）

林小茶　陈维兴　编著

10

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

基础教育系列

21世纪高校计算机应用技术系列规划教材
丛书主编 谭浩强

C++面向对象程序设计习题解答与上机指导（第二版）

林小茶　陈维兴　编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是《C++面向对象程序设计（第二版）》（陈维兴、林小茶编著，中国铁道出版社出版）的配套教材。

书中内容分为三篇：第一篇“《C++面向对象程序设计（第二版）》习题和参考答案”，对主教材中的每道习题都给出了详细的解答，这些习题是作者多年以来在教学中积累、收集并经过验证的，全部上机调试通过；第二篇“C++上机实验环境介绍”介绍了在 Visual C++ 6.0 和 Turbo C++ 3.0 环境下调试与运行程序的方法，以方便读者熟悉上机环境；第三篇“上机实验题与参考答案”中有 10 套上机实验题，每套实验题都给出了实验目的和要求、实验内容、实验步骤以及参考源程序。

本书适合作为高等院校各专业学生学习 C++基础知识的配套教材，也适合单独作为学习 C++ 的辅导书。

图书在版编目（CIP）数据

C++面向对象程序设计习题解答与上机指导 / 林小茶，陈维兴编著. —2 版. —北京：中国铁道出版社，2009.11

（21 世纪高校计算机应用技术系列规划教材. 基础教育系列）

ISBN 978-7-113-10783-3

Ⅰ.C… Ⅱ.①林…②陈… 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（2009）第 214663 号

书　　名：C++面向对象程序设计习题解答与上机指导（第二版）

作　　者：林小茶　陈维兴　编著

策划编辑：秦绪好

责任编辑：崔晓静　　**编辑部电话：（010）63583215**

编辑助理：侯　颖

责任印制：李　佳　　**封面制作：白　雪**

出版发行：中国铁道出版社（北京市宣武区右安门西街 8 号　　邮政编码：100054）

印　　刷：河北省遵化市胶印厂

版　　次：2004 年 5 月第 1 版　2009 年 12 月第 2 版　2009 年 12 月第 6 次印刷

开　　本：787mm×1092mm　1/16　印张：12.75　字数：310 千

印　　数：5 000 册

书　　号：ISBN 978-7-113-10783-3/TP・3652

定　　价：20.00 元

序

PREFACE

21世纪是信息技术高度发展且得到广泛应用的时代，信息技术从多方面改变着人类的生活、工作和思维方式。每一个人都应当学习信息技术、应用信息技术。人们平常所说的计算机教育其内涵实际上已经发展为信息技术教育，内容主要包括计算机和网络的基本知识及应用。

对多数人来说，学习计算机的目的是为了利用这个现代化工具工作或处理面临的各种问题，使自己能够跟上时代前进的步伐，同时在学习的过程中努力培养自己的信息素养，使自己具有信息时代所要求的科学素质，站在信息技术发展和应用的前列，推动我国信息技术的发展。

学习计算机课程有两种不同的方法：一是从理论入手；二是从实际应用入手。不同的人有不同的学习内容和学习方法。大学生中的多数人将来是各行各业中的计算机应用人才。对他们来说，不仅需要“知道什么”，更重要的是“会做什么”。因此，在学习过程中要以应用为目的，注重培养应用能力，大力加强实践环节，激励创新意识。

根据实际教学的需要，我们组织编写了这套“21世纪高校计算机应用技术系列规划教材”。顾名思义，这套教材的特点是突出应用技术，面向实际应用。在选材上，根据实际应用的需要决定内容的取舍，坚决舍弃那些现在用不到、将来也用不到的内容。在叙述方法上，采取“提出问题—解决问题—归纳分析”的三部曲，这种从实际到理论、从具体到抽象、从个别到一般的方法，符合人们的认知规律，且在实践过程中已取得了很好的效果。

本套教材采取模块化的结构，根据需要确定一批书目，提供了一个课程菜单供各校选用，以后可根据信息技术的发展和教学的需要，不断地补充和调整。我们的指导思想是面向实际、面向应用、面向对象。只有这样，才能比较灵活地满足不同学校、不同专业的需要。在此，希望各校的老师把你们的要求反映给我们，我们将会尽最大努力满足大家的要求。

本套教材可以作为大学计算机应用技术课程的教材以及高职高专、成人高校和面向社会的培训班的教材，也可作为学习计算机的自学教材。

由于全国各地区、各高等院校的情况不同，因此需要有不同特点的教材以满足不同学校、不同专业教学的需要，尤其是高职高专教育发展迅速，不能照搬普通高校的教材和教学方法，必须要针对它们的特点组织教材和教学。

本套教材包括以下五个系列：

- 基础教育系列
- 高职高专系列
- 实训教程系列
- 案例汇编系列
- 试题汇编系列

其中基础教育系列是面向应用型高校的教材，对象是普通高校的应用性专业的本科学生。高职高专系列是面向两年制或三年制的高职高专院校的学生，突出实用技术和应用技能，不涉及过多的理论和概念，强调实践环节，学以致用。后面三个系列是辅助性的教材和参考书，可供应用型本科和高职学生选用。

本套教材自 2003 年出版以来，已出版了 70 多种，受到了许多高校师生的欢迎，其中有多种教材被国家教育部评为**普通高等教育“十一五”国家级规划教材**。《计算机应用基础》一书出版三年内发行了 50 万册。这表示了读者和社会对本系列教材的充分肯定，对我们是有力的鞭策。

本套教材由浩强创作室与中国铁道出版社共同策划，选择有丰富教学经验的普通高校老师和高职高专院校的老师编写。中国铁道出版社以很高的热情和效率组织了这套教材的出版工作。在组织编写及出版的过程中，得到全国高等院校计算机基础教育研究会和各高等院校老师的热情鼓励和支持，对此谨表衷心的感谢。

本套教材如有不足之处，请各位专家、老师和广大读者不吝指正。希望通过本套教材的不断完善和出版，为我国计算机教育事业的发展和人才培养做出更大贡献。

全国高等院校计算机基础教育研究会会长
“21 世纪高校计算机应用技术系列规划教材”丛书主编

谭浩强

第二版前言

FOREWORD

本书是为了配合《C++面向对象程序设计（第二版）》（陈维兴、林小茶编著，中国铁道出版社出版）而编写的辅助教材。

由于第二版教材对第一版教材有一些改进，因此辅助教材也做了相应的改动。

（1）为了适应教学，更换了一些习题。我们认为，这些习题能更好地培养学生的程序设计能力。

（2）主教材中使用标准C++的头文件改写了所有的源程序，本书也同样将习题解答中所有的源程序进行了修改，系统头文件不带扩展名.h，使用系统库时使用命名空间std。

有一点提请读者特别注意：本书保留了对Turbo C++ 3.0环境的描述，但是，该环境并不支持命名空间std。因此，如果在Turbo C++ 3.0环境下对程序进行调试，应该对程序做一定的修改。

```
#include <iostream>
using namespace std;
```

上面两行是使用命名空间的语句，要使程序能在Turbo C++ 3.0环境下执行，应将其修改为：

```
#include <iostream.h>
```

从编者多年的教学经验来讲，面向对象程序设计思想的建立不是一件容易的事情，许多学生都反映，听得懂，看得懂，就是自己一编程序，仍然感觉无从下手。本辅助教材就是想帮助学生解决编写程序难这一问题。因此，在选择例题时，特别注意配合主教材的内容，循序渐进地对面向对象程序设计的能力进行培养。

请读者正确地使用本书，不要在没有经过任何思考的情况下，直接阅读答案，这样做一点好处也没有，并且违背了我们编写本书的初衷。

本书第一篇的内容由陈维兴和林小茶共同编写，附录由陈维兴编写，第二篇和第三篇由林小茶编写。

本书适合作为高等院校各专业学生学习C++基础知识的配套教材，也适合单独作为学习C++的辅导书。

由于作者水平有限，书中难免还存在疏漏，殷切希望广大读者批评指正。

编 者

2009年10月

第一版前言 FOREWORD

学过程序设计的人，都有一个体会，看别人编写的程序，好像挺明白的，但是一旦要自己编写一个程序，就感觉无从下手。这是因为程序设计是一门对实践环节要求很高的课程，初学者要想真正学会C++语言程序设计，最重要的要抓住两个关键环节：一个是多做程序设计的习题，多编程，另一个就是多上机。写在纸上的程序是否正确，最好的办法就是上机验证一下。为此，我们编写了这本习题解答与实验指导。本书在对教材中的习题进行解答的同时，也对一些基本的程序算法和规则进行了详细的分析，希望能帮助学习者尽快掌握C++语言程序设计的基本规则与编程规律，并能够熟练运用这些规则与技巧，编制出具有良好风格的应用程序，最终能够顺利地通过上机调试。

本书的主要内容分为3篇：第一篇“《C++面向对象程序设计》习题和参考答案”是对教材中的习题的详细解答；第二篇“C++语言上机实验环境介绍”介绍了C++程序设计调试环境Visual C++ 6.0和Turbo C++；第三篇“上机实验题与参考答案”安排了10套精心设计的实验，每个实验都给出了详细的实验目的、实验要求和实验步骤，帮助学习者掌握C++程序设计的方法，并进一步加深对课程相关内容的理解与掌握。考虑到学生在上机操作中，经常遇到一些错误信息，但看不懂，不知如何修改，因此我们把一些常用的错误信息注释放在附录中，以供学生上机实验时使用。

提供习题解答和实验参考源程序的主要目的是给学习者一个参考和借鉴，作者在这里要强调一点，程序设计是一个创作的过程，解决一个实际问题的程序肯定不是唯一的，因此，在阅读本书的参考答案之前，希望读者已经独立思考过教材中的习题以及实验题目，这样才有助于程序设计水平的提高，并且不要把本书的参考源程序作为唯一的答案。本书中所有程序都经作者在Visual C++6.0或Turbo C++3.0上调试通过。

本书第一篇的内容由陈维兴教授和林小茶副教授共同编写，附录由陈维兴编写，第二篇和第三篇由林小茶编写。

在本书编写和出版过程中，全国计算机基础教育研究会会长谭浩强教授给予了指导和把关，在此表示最衷心的感谢。

由于水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编 者

2004年4月

目录

CONTENTS

第一篇 《C++面向对象程序设计（第二版）》习题和参考答案

第二篇 C++上机实验环境介绍

第三篇 上机实验题与参考答案

第一篇 《C++面向对象程序设计（第二版）》习题和参考答案

第 1 章 面向对象程序设计概述

【1.1】什么是面向对象程序设计?

【解】面向对象程序设计是一种新型的程序设计范型。这种范型的主要特征是：

程序=对象+消息

面向对象程序的基本元素是对象，面向对象程序的主要结构特点是：第一，程序一般由类的定义和类的使用两部分组成，在主程序中定义各对象并规定它们之间传递消息的规律。第二，程序中的一切操作都是通过向对象发送消息来实现的，对象接收到消息后，启动有关方法完成相应的操作。

面向对象程序设计方法模拟人类习惯的解题方法，代表了计算机程序设计新颖的思维方式。这种方法的提出是对软件开发方法的一场革命，是目前解决软件开发所面临的困难最有希望、最有前途的方法之一。

【1.2】在面向对象程序设计中，什么是对象？什么是类？对象与类的关系是什么？

【解】在面向对象程序设计中，对象是描述其属性的数据以及对这些数据施加的一组操作封装在一起构成的统一体。对象可以认为是：数据+操作。对象所能完成的操作表示它的动态行为，通常也把操作称为方法。

在面向对象程序设计中，类就是具有相同的数据和相同的操作的一组对象的集合。也就是说，类是对具有相同数据结构和相同操作的一类对象的描述。

类和对象之间的关系是抽象和具体的关系。类是对多个对象进行综合抽象的结果，一个对象是类的一个实例。

【1.3】现实世界中的对象有哪些特征?

【解】现实世界中的对象具有以下特性：

（1）每一个对象必须有一个名字，以区别于其他对象。

（2）用属性来描述它的某些特征。

（3）有一组操作，每个操作决定对象的一种行为。

（4）对象的操作可以分为两类：一类是自身所承受的操作，另一类是施加于其他对象的操作。

【1.4】什么是消息？消息具有什么性质？

【解】在面向对象程序设计中，一个对象向另一个对象发出的请求称为消息。当对象接收到发向它的消息时，就调用有关的方法，执行相应的操作。消息是一个对象要求另一个对象执行某个操作的规格说明，通过消息传递才能完成对象之间的相互请求或相互协作。

消息具有以下3个性质：

（1）同一个对象可以接收不同形式的多个消息，做出不同的响应。

（2）相同形式的消息可以传递给不同的对象，所做出的响应可以是不同的。

（3）对消息的响应并不是必须的，对象可以响应消息，也可以不响应。

【1.5】什么是方法？在C++中它是通过什么来实现的？

【解】面向对象程序设计中的消息传递实际是对现实世界中的信息传递的直接模拟。调用对象中的函数就是向该对象传送一个消息，要求该对象实现某一行为（功能、操作）。对象所能实现的行为（操作），在程序设计方法中称为方法，它们是通过调用相应的函数来实现的，在C++中方法是通过成员函数来实现的。

【1.6】什么是抽象和封装？

【解】抽象是人类认识问题最基本的手段之一。抽象是对有关事物的共性进行归纳、集中的过程。抽象是通过特定的实例（对象）抽取共同性质后形成概念的过程。面向对象程序设计中的抽象包括两个方面：数据抽象和代码抽象（又称行为抽象）。前者描述了某类对象的属性或状态，也就是此类对象区别于彼类对象的特征物理量；后者描述了某类对象的共同行为特征或具有的共同功能。

在现实世界中，所谓封装就是把某个事物包裹起来，使外界不知道该事物的具体内容。在面向对象程序设计中，封装是指把数据和实现操作的代码集中起来放在对象内部，并尽可能隐蔽对象的内部细节。

封装是面向对象程序设计方法的一个重要特性。封装具有两方面的含义：一是将有关的数据和操作代码封装在一个对象中，各个对象相对独立、互不干扰；二是将对象中某些数据与操作代码对外隐蔽，即隐蔽其内部细节，只留下少量接口，以便与外界联系，接收外界的消息。这种对外界隐蔽的做法称为信息隐蔽。信息隐蔽有利于数据安全，防止无关人员访问和修改数据。

【1.7】什么是继承？请举例说明。

【解】继承所表达的是对象类之间的相关关系，这种关系使得某类对象可以继承另外一类对象的特征和能力。在现实生活中，继承是很普遍和容易理解的。例如，我们继承了父母的一些特征，如种族、血型、眼睛的颜色等，父母是我们所具有的属性的基础。

图1-1-1是一个继承的典型例子：汽车的继承层次。

以面向对象程序设计的观点，继承所表达的是对象与类之间相关的关系，这种关系使得某一类可以继承另外一个类的特征和能力。

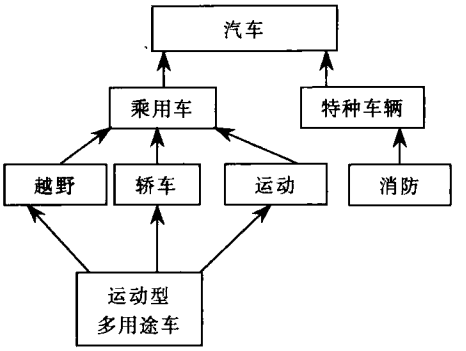

图 1-1-1　汽车的继承层次图

【1.8】若类之间具有继承关系，那它们之间具有什么特征？

【解】若类之间具有继承关系，那它们之间具有以下几个特性：

（1）类间具有共享特征（包括数据和操作代码的共享）。

（2）类间具有差别或新增部分（包括非共享的数据和操作代码）。

（3）类间具有层次结构。

假设有两个类 A 和 B，若类 B 继承类 A，则类 B 包含了类 A 的特征（包括数据和操作），同时也可以加入自己所特有的新特性。这时，我们称被继承类 A 为基类或父类，而称继承类 B 为类 A 的派生类或子类。同时，我们还可以说，类 B 是从类 A 中派生出来的。

【1.9】什么是单继承、多继承？请举例说明。

【解】从继承源上分，继承分为单继承和多继承。单继承是指每个派生类只直接继承了一个基类的特征。图 1-1-2 表示了一种单继承关系，它表示 Windows 操作系统中窗口间的继承关系。

多继承是指多个基类派生出一个派生类的继承关系。多继承的派生类直接继承了不止一个基类的特征。例如，小孩喜欢的玩具车既继承了车的一些特性，又继承了玩具的一些特征，如图 1-1-3 所示。

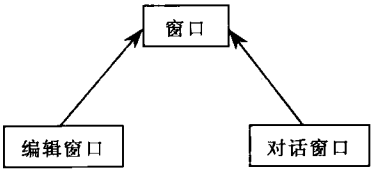

图 1-1-2　单继承关系

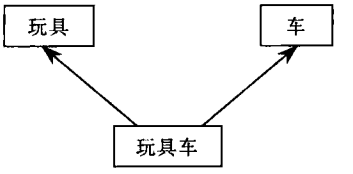

图 1-1-3　多继承关系

【1.10】什么是多态性？请举例说明。

【解】多态性也是面向对象程序的重要特征。它是指不同的对象收到相同的消息时产生不同的行为方式。例如，我们同样双击 Windows 系统桌面上的图标，有的是打开多媒体播放器，有的是打开资源管理器。

利用多态性，用户只需发送一般形式的消息，而将所有的实现留给接收消息的对象。对象根据所收到的消息而做出相应的动作。

【1.11】传统程序设计方法的局限性主要有哪些？

【解】传统程序设计方法的局限性主要有 3 个方面：

（1）传统程序设计开发软件的生产效率低下。

传统程序设计仍是采用较原始的方式进行程序开发，程序设计基本上还是从语句一级开始。软件生产中缺乏大粒度、可重用的构件，软件的重用问题没有得到很好的解决，从而导致软件生产的工程化和自动化屡屡受阻。传统程序设计的特点是数据与操作分离，而且对同一数据的操作往往分散在程序的不同地方。这样，如果一个或多个数据的结构发生了变化，那么这种变化将波及程序的很多部分，甚至遍及整个程序，致使许多函数和过程必须重写，严重时会导致整个软件结构的崩溃。传统程序设计是面向过程的，数据和操作相分离的结构使得维护数据和处理数据的操作过程要花费大量的精力和时间，严重地影响了软件的生产效率。

总之，要提高软件生产效率，就必须很好地解决软件的重用性、复杂性和可维护性问题。但是传统的程序设计是难以解决这些问题的。

（2）传统程序设计难以应付日益庞大的信息量和多样的信息类型。

当代计算机的应用领域已从数值计算扩展到了人类社会的各个方面，所处理的数据已从简单的数字和字符发展为具有多种格式的多媒体数据，如文本、图形、图像、影像、声音等，描述的问题从单纯的计算问题到仿真复杂的自然现象和社会现象。随着计算机处理的信息量与信息类型的迅速增加，程序的规模日益庞大、复杂度不断增加，传统程序设计方法是无法应付的。

（3）传统程序设计难以适应各种新环境。

当前，并行处理、分布式、网络和多机系统等已经或将是程序运行的主流方式和主流环境。这些环境的一个共同的特点是具有一些有独立处理能力的结点，结点间有通信机制，即以消息传递进行联络。显然，传统程序设计很难适应这些新环境。

【1.12】面向对象程序设计的优点主要有哪些？

【解】面向对象程序设计的优点主要体现在以下几个方面：

（1）可提高程序的重用性。

重用是提高软件开发效率的最主要的方法，面向对象程序设计能较好地解决软件的重用问题。对象所固有的封装性和信息隐藏等机理，使得对象内部的实现与外界隔离，具有较强的独立性，它可以作为一个大粒度的程序构件，供同类程序直接使用。

（2）可控制程序的复杂性。

面向对象程序设计采用了数据抽象和信息隐藏技术，把数据及对数据的操作放在一个个类中，作为相互依存、不可分割的整体来进行处理。这样，在程序中任何要访问这些数据的地方都只需简单地通过传递消息和调用方法来进行，这就有效地控制了程序的复杂性。

（3）可改善程序的可维护性。

在面向对象程序设计中，对对象的操作只能通过消息传递来实现，所以只要消息模式即对应的方法界面不变，方法体的任何修改都不会导致发送消息的程序的修改，这显然给程序的维护带来了方便。另外，类的封装和信息隐藏机制使得外界对其中的数据和程序代码的非法操作成为不可能，这也就大大减少了程序的错误率。

（4）能够更好地支持大型程序设计。

面向对象程序设计在数据抽象和抽象数据类型之上又引入了动态连接和继承性等机制，进一步发展了基于数据抽象的模块化设计，使其更好地支持大型程序设计。

（5）增强了计算机处理信息的范围。

面向对象程序设计方法模拟人类习惯的解题方法，代表了计算机程序设计新颖的思维方法。用类来直接描述现实世界中的类型，可使计算机系统的描述和处理对象从数据扩展到现实世界和思维世界中的各种事物，这实际上大大扩展了计算机系统处理的信息量和信息类型。

（6）很好地适应新的硬件环境。

面向对象程序设计中的对象、消息传递等思想和机制，与分布式、并行处理、多机系统及网络等硬件环境恰好相吻合。面向对象的思想也影响到计算机硬件的体系结构，现在已在研究直接支持对象概念的面向对象计算机。这样的计算机将会更适合于面向对象程序设计，能更充分地发挥面向对象技术的威力。

第 2 章 C++基础

【2.1】简述 C++的主要特点。

【解】C++最主要的特点有以下几点：

（1）C++保持与 C 兼容，这就使许多 C 程序代码可不经修改就为 C++所用，用 C 编写的众多的库函数和实用软件可以用于 C++中。

（2）C++是一个更好的 C，它保持了 C 语言简洁、高效和接近汇编语言等特点，并对 C 语言的功能做了不少扩充。用 C++编写的程序比用 C 语言编写的程序更安全，可读性更好，代码结构更为合理。

（3）用 C++编写的程序质量高，从开发时间、费用到形成的软件的可重用性、可扩充性、可维护性和可靠性等方面有了很大的提高，使得大中型的程序开发变得更加容易。

（4）增加了面向对象的机制，C++几乎支持所有的面问对象程序设计特征，体现了近 20 年来在程序设计和软件开发领域出现的新思想和新技术。

C++最有意义的方面是支持面向对象的特征，然而，由于 C++与 C 保持兼容，使得 C++不是一个纯正的面向对象的语言，C++既可用于面向过程的结构化程序设计，也可用于面向对象的程序设计。

【2.2】下面是一个 C 程序，改写它，使它采用 C++风格的 I/O 语句。

```
#include <stdio.h>
int main()
{ int a,b,d,min;
  printf("Enter two numbers: ");
  scanf("%d%d",&a,&b);
  min=a>b?b:a;
  for(d=2;d<min;d++)
    if(((a%d)==0)&&((b%d)==0))  break;
  if(d==min)
  { printf("No  common  denominators\n");
    return  0;
  }
  printf("The lowest common denominator is %d\n",d);
  return 0;
}
```

【解】

```
#include <iostream>
using namespace std;
int main()
{ int a,b,d,min;
  cout<<"Enter two numbers: ";
  cin>>a;
  cin>>b;
  min=a>b?b:a;
  for(d=2;d<min;d++)
    if(((a%d)==0)&&((b%d)==0))  break;
  if(d==min)
  { cout<<"No common denominators\n";
    return 0;
  }
  cout<<"The lowest common denominator is "<<endl<<d;
  return 0;
}
```

【2.3】函数重载是指（　　）。

A．两个或两个以上的函数取相同的函数名，但形参的个数或类型不同

B．两个以上的函数取相同的名字和具有相同的参数个数，但返回值的类型不同

C．两个以上的函数名字不同，但形参的个数或类型相同

D．两个以上的函数取相同的函数名，并且函数的返回类型相同

【解】A

【2.4】声明或定义一个内联函数时，必须在函数开始使用关键字（　　）。

A．static　　B．inline　　C．const　　D．extern

【解】B

【2.5】一个函数功能不太复杂，但被频繁调用，应选用（　　）。

A．内联函数　　B．重载函数　　C．递归函数　　D．嵌套函数

【解】A

【2.6】在C++中，下列关于设置参数默认值的描述中，正确的是（　　）。

A．程序中有函数重载，就不能设置参数默认值

B．设置参数默认值，只能在函数定义时进行

C．设置参数默认值时，应该是先设置右边的再设置左边的

D．设置参数默认值时，应该全部参数都设置

【解】C

【2.7】已知 int m=10，在下列表示引用的方法中，（　　）是正确的。

A．int &x=m;　　B．int &y=10;　　C．int &z;　　D．float &t=&m;

【解】A

【2.8】在C++中，下列关于设置函数默认参数值的描述中，（　　）是正确的。

A．不允许设置默认参数值

B．在指定了默认值的参数右边，不能出现没有指定默认值的参数

C．只能在函数的定义中指定参数的默认值

D．设置默认参数值时，必须全部都设置

【解】B

【2.9】下列描述中关于引用调用的是（　　）。

A．形参是指针，实参是地址值　　B．形参是引用，实参是变量

C．形参和实参都是变量　　D．形参和实参都是数组名

【解】B

【2.10】下列语句中错误的是（　　）。

A．int *p=new int(10);　　B．int *p=new int[10];

C．int *p=new int;　　D．int *p=new int[40](0);

【解】D

【说明】

“int *p=new int(10);”表示动态分配1个整型内存空间，初值为10。

“int *p=new int[10];”表示动态分配10个整型内存空间。

“int *p=new int;”表示动态分配1个整型内存空间。

“int *p=new int[40](0)”想给一个数组分配内存空间时，对数组进行初始化，这是不允许的。

【2.11】假设已经有定义“const char *const name="chen";”，下面的语句中正确的是（　　）。

A．name[3]='a';　　B．name="lin";

C．name=new char[5];　　D．cout<<name[3];

【解】D

【说明】name被定义为指向常量的常指针，所以它所指的内容和本身的内容都不能修改。“name[3]='a';”修改了name所指的常量，“name="lin";”和“name=new char[5];”修改了常指针，只有D输出一个字符是正确的。

【2.12】假设已经有定义“char *const name="chen";”，下面的语句中正确的是（　　）。

A．name[3]='q';　　B．name="lin";

C．name=new char[5];　　D．name=new char('q');

【解】A

【说明】name被定义成常指针，所以它所指的内容能改变，但指针本身的内容不可以修改。“name[3]='q';”修改了name所指的内容，是正确的。而“name="lin";”、“name=new char[5];”和“name=new char('q');”以不同的方法修改了常指针，都是错误的。

【2.13】假设已经有定义“const char *name="chen";”，下面的语句中错误的是（　　）。

A．name[3]='q';　　B．name="lin";

C．name=new char[5];　　D．name=new char('q');

【解】A

【说明】name被定义指向常量的指针，所以它所指的内容不能改变，但指针本身的内容可以修改。“name[3]='q';”修改了name所指的内容，是错误的。“name=="lin";”、“name=new char[5];”和“name= new char('q')”以不同的方法修改了常指针，都是正确的。

【2.14】重载函数在调用时选择的依据中，（　　）是错误的。

A．函数名字　　B．函数的返回类型　　C．参数个数　　D．参数的类型

【解】B