

21世纪高等院校计算机教材系列

C++语言和面向对象程序设计

教程习题解答及上机实践

●宛延闯 甄炜 李俊 等编著

 机械工业出版社
CHINA MACHINE PRESS



21 世纪高等院校计算机教材系列

C++ 语言和面向对象程序设计 教程习题解答及上机实践

宛延阁 甄 炜 李 俊 等编著



机械工业出版社

本书是与《C++语言和面向对象程序设计教程》(宛延阁等编著)一书配套使用的所有14章习题的详尽解答及上机练习。习题内容丰富、编排精炼准确。习题类型有填空题、选择题、改错题、问答题和编程题等。通过解题和上机可加强读者对C++语言基本概念、面向对象程序设计和面向对象建模内涵的理解。

本书可作为高等院校学生学习C++语言和面向对象程序设计的补充教材,也是广大读者和科技工作者学习C++语言和面向对象程序设计必备的参考书。

图书在版编目(CIP)数据

C++语言和面向对象程序设计教程习题解答及上机实践/宛延阁等编著. —北京:机械工业出版社,2005.6

(21世纪高等院校计算机教材系列)

ISBN 7-111-16159-9

I. C++... II. 宛... 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(2005)第012151号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策 划:胡毓坚 责任编辑:孙 业 版式设计:张世琴

责任校对:刘志文 责任印制:杨 曦

高等教育出版社印刷厂印刷·新华书店北京发行所发行

2005年7月第1版第1次印刷

787mm×1092mm 1/16·8.25印张·198千字

0 001—5000册

定价:14.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68326294

封面无防伪标均为盗版

出版说明

计算机技术是一门迅速发展的现代科学技术，它在经济建设与社会发展中，发挥着非常重要的作用。近年来，我国高等院校十分注重人才的培养，大力提倡素质教育、优化知识结构，提倡大学生必须掌握计算机应用技术。为了满足教育的需求，机械工业出版社组织了这套“21世纪高等院校计算机教材系列”。

在本套系列教材的组织编写过程中，我社聘请了各高等院校相关课程的主讲老师进行了充分的调研和细致的研讨，并针对非计算机专业的课程特点，根据自身的教学经验，总结出知识点、重点和难点，一并纳入到教材中。

本套系列教材定位准确，注重理论教学和实践教学相结合，逻辑性强，层次分明，叙述准确而精炼，图文并茂，习题丰富，非常适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

参加编写本系列教材的院校包括：清华大学、西安交通大学、北京交通大学、北京邮电大学、北京化工大学、北京科技大学、山东大学、首都经贸大学等。

机械工业出版社

前 言

本书是与《C++语言和面向对象程序设计教程》一书配套使用的习题解答及上机练习。

为了使读者更好地掌握并学会 C++ 语言，每章的习题对理解面向对象程序设计基本概念、C++ 编程和面向对象建模都是至关重要的。本书的习题类型有填空题、选择题、改错题、问答题以及编程题等，内容丰富、习题设计精炼准确。这些习题侧重点在于对 C++ 语言基本概念、面向对象程序设计和面向对象建模精髓的理解，以便广大读者和学生尽快地全面掌握。鉴于读者或许不一定能够正确完整地回答书中的每道练习题，所以，我们把《C++ 语言和面向对象程序设计教程》一书中 14 章的所有习题尽量详尽地解答出来，供广大读者和学生学习时参考。

上机实践部分包括了函数应用、类与对象、数组、指针和字符串、多态性、流类库的输入/输出等，一共 26 道上机练习题。读者上机的环境既可以采用 VC++，也可以采用 Borland C++。具体的上机实践环境读者可参照相应的参考书。

参加本书编著的还有石良秀、乔立琴、李保林、代宁、郭应中、高晓丽、王子滨、王浩枫、潘京、季英洁、王玥和王心颖等。

宛延阁

目 录

出版说明

前言

第一部分 习题解答

第1章 绪论	1	第9章 运算符重载	50
第2章 C++ 程序设计初步	4	第10章 包容器类	55
第3章 C++ 语言基础	8	第11章 例外处理和命名空间	59
第4章 C++ 函数	15	第12章 面向对象建模	62
第5章 指针和引用	20	第13章 面向对象设计与实现	67
第6章 类和对象	25	第14章 C++ 面向对象设计与实现的典型实例剖析	72
第7章 继承	31		
第8章 多态、虚拟函数和模板	39		

第二部分 上机实践

实践1 产生列表表格	102	实践15 继承 String 类	113
实践2 递减赋值运算	102	实践16 寻找由用户提供的多个数值的平均值	114
实践3 华氏温度与摄氏温度相互转换	103	实践17 将指针数组的内容排序到串	115
实践4 只有四则运算的计算器	103	实践18 创建数组类	116
实践5 用结构(struct)来计算房子的体积	104	实践19 重载赋值运算符和拷贝构造函数	117
实践6 用函数计算圆的面积	105	实践20 向数组中写数据	118
实践7 用内联函数完成磅与公斤的互换	105	实践21 模仿 COPY 命令	119
实践8 采用引用传递方式查找实数的整数和小数部分	106	实践22 显示文件的长度	120
实践9 把整数数据类型模型化为类	106	实践23 使用平均值数组的函数模板	121
实践10 将 C 串反转	107	实践24 实现作为一模板的队列类, 并在处理队列错误中使用例外机制	122
实践11 用串作为数据的雇用对象	108	实践25 在数组中存储浮点类型数并用 sort() 函数排序	123
实践12 用运算符 '+' 重载来连接两个串	109	实践26 使用串对象和 push_back() 函数的向量	124
实践13 用重载 '+' 运算符将两个时间(time)相加	110		
实践14 用 Int 类型重载算术运算符	111		

第一部分 习题解答

第1章 绪 论

1.1 填空题

- (1) 计算机软件开发一直受到_____和_____难题的困扰。
- (2) 面向过程语言主要有_____和_____缺陷。
- (3) 无保留库的软件开发方法把_____活动的高墙拆开。它的开发过程是采用_____方法以及_____和_____。
- (4) 面向对象技术的要点是_____、_____、_____、_____和_____。
- (5) 软件的鲁棒性表明了_____的能力。
- (6) 面向对象技术 OMT 包括了刻画一个系统的三种模型是_____、_____和_____。

答：(1) 如何超越程序复杂性障碍，如何在计算机系统中自然地表示客观世界
(2) 数据抽象能力差，模块化能力差
(3) 三个，面向对象分析和设计方法，面向对象程序设计语言，面向对象数据库
(4) 抽象，封装，数据和行为联合，共享，对象结构，协同作用
(5) 系统抵御冲击、维持生存的能力
(6) 对象模型，功能模型，动态模型

1.2 选择题(至少有一个，可以多选)

- (1) 零件、配件与设计图的关系，相当于对象与_____的关系。
a. 成员函数；b. 类；c. 运算符；d. 数据项；e. 基本操作；f. 基本属性。
- (2) 传统的软件开发包含了_____。
a. 两个独立的活动，它们是分析和设计以及程序设计；
b. 三个独立的活动，它们是分析和设计，建立模型以及数据库设计；
c. 三个独立的活动，它们是分析和设计，程序设计以及建立概念模型和存取数据库；
d. 四个独立的活动，它们是分析和设计，程序设计，定义概念模型以及存取数据库。
- (3) 对象是_____。
a. 算法 + 数据结构的累加；
b. 算法 + 数据结构 + 程序设计的累加；

- c. 把数据结构 + 相应的行为封装在一起的实体;
 - d. 把数据结构 + 行为 + 函数运算符封装在一起的实体。
- (4) 以下哪种是我们要使用面向对象语言的正确理由? _____。
- a. 可以自行定义所需的数据类型;
 - b. 面向对象语言的语句比面向过程语言的语句简单易学好用;
 - c. 面向对象语言能够自动地发现并修改已发生的错误;
 - d. 面向对象语言很容易概念化、可重用。

答: (1) b (2) c (3) c (4) a, d

1.3 什么是对象(object)? 什么是面向对象(object-oriented)? 为什么要面向对象?

答: 什么是对象?

在现实生活中, 文档中的一个段落, 工作站上的一个窗口, 国际象棋中的皇后, 一粒米, 二叉树, 小乔的自行车, 小石的汽车, 小李的房子, 策略符号表等等都是对象。

一个对象像一个软件构造块, 它包含了数据结构和提供的相关的行为(操作)。在计算机中表示真实世界模拟思维的抽象。对象本身可以为用户提供一系列服务: 改变对象的状态、测试、传递消息等, 用户无需知道服务的任何实现细节, 操作完全是封闭的。

什么是面向对象?

“面向对象”是把一组相互无关联的对象有机地集成在一起的软件, 而这些对象都是将数据结构和行为紧密结合在一起, 这与传统的将数据结构与行为松散连接的模式完全不一样。

为什么要面向对象?

四十年以来的研究与开发, 突出的发现: 软件开发仅仅把注意力放在功能上是远远不够的。诚然, 软件开发首要的任务是提供正确的功能, 但在功能结构上, 必须能够扩充、删除和修改, 软件必须能够反复使用(可重用)。功能本身是易变的, 不可能达到上述目的。而对象则在客户需求不断变化的情况下相对稳定, 因为无论功能怎么千变万化, 一个问题空间中的对象一般总能保持其稳定不变性。这样, 围绕对象构造的软件系统自然也会有较好的稳定性。所以, 我们的软件系统要采用面向对象技术。

1.4 什么是面向对象程序设计(OOP)? 什么是面向对象程序设计语言(OOPL)?

答: 什么是面向对象程序设计(OOP)?

OOP 中心是围绕抽象数据类型和类、类层次结构、继承和多态进行的程序设计工作。类和继承是符合人们一般思维方式的描述模式。

什么是面向对象程序设计语言(OOPL)?

我们说, 具备以下特征的语言, 就是 OOPL。这些特征有: ①对象; ②可编程性; ③访问控制; ④继承性; ⑤多态性; ⑥可预见性。

1.5 自顶向下设计方法与自底向上设计方法各自有什么特点? 这两种设计方法与面向对象方法有何关联?

答: 自顶向下设计方法的特点: 从问题大的方面入手来寻找解决办法, 然后再解决剩下相对较小的问题。这样, 经过多次迭代最终获得完整明确的解决办法。

自底向上的设计方法的特点: 从解决基本的、简单的问题入手。在此基础上逐步建立解决复杂问题的能力, 直至整个问题得以圆满解决。正好与自顶向下的设计方法相反。

面向对象设计方法是兼有这两者(自顶向下和自底向上)特点的方法。它鼓励人们从问题的基本的简单的方面入手,用对象来描述问题的主要方面(这正是自底向上方法的本质)。同时,面向对象设计方法又要求人们面向目标,考虑为了达到这一目标该如何建立这些基本对象(这也正体现了自顶向下的设计思想)。面向对象设计方法关键的优点在于代码的共享和重用,这是问题的基本方面,正是因为此特点,就有别于自顶向下和自底向上设计方法。

1.6 面向对象方法学包括几个基本重要阶段?如何刻画一个软件系统?

答:面向对象方法学是由应用领域中所创建的模型组成的,然后在系统设计期间添加实现的细节。此方法称作对象模型技术(OMT)方法。有以下几个重要阶段:

- (1) 分析阶段;
- (2) 系统设计阶段;
- (3) 对象设计阶段;
- (4) 实现阶段。

OMT 方法采用三种模型来刻画一个系统:对象模型,它描述系统中的对象和它们之间的关系;动态模型,它描述系统中对象之间的相互作用;功能模型,它描述系统中的数据的变迁。每种模型应用于开发的不同阶段,以几种不同的观点来刻画不同的模型。一个系统的完全描述需要包含这三种模型。

1.7 为什么要学习 C++ 语言?它在企业信息化建设中起什么样的作用?

答:为什么要学习 C++ 语言?

首先,C++ 是一种混合型的面向对象语言,它既保留了高效的 C,又具备了强大的面向对象特征。其次,C++ 是世界流行语言,是世界各大软件公司开发软件的主流语言,功能强大,并能把各大软件公司经过几十年用 C 开发的类库和有用的软件保留下来(减少花费),同时又能在在此基础上开发先进的面向对象软件。所以,C++ 是软件开发的很好的起跳点。

C++ 在企业信息化建设中起什么样的作用?

作用很大。目前,C++ 在尖端信息领域,如金融、航天航空、通信等领域已站稳脚跟,并逐步在网络、移动通信、大型数据处理、石化、电子等企业,为开发人员所接受。我国信息化建设在计算机语言方面,C++ 已处主导作用。

1.8 软件从广义角度可分为哪几种?试举例说明。

答:软件从广义角度可分为系统软件和应用软件。

系统软件包括操作系统,比如 Windows, UNIX 等。

应用软件,种类繁多。比如字处理软件 WordPerfect, Word, AmiPro; 电子表格应用软件有 Execl; 画图的软件有 AutoCAD, Visio 等。

第2章 C++ 程序设计初步

2.1 填空题

- (1) C++ 程序通常使用_____程序进入计算机。
- (2) 在 C++ 系统中, 编译开始之前执行的是_____程序。
- (3) _____程序把编译器的输出与不同的库函数组装在一起产生可执行代码。
- (4) _____程序把内存中各部分 C++ 可执行代码组装在一起。
- (5) 每个 C++ 程序开始执行的函数是_____。
- (6) 每个函数体由_____符号开始, 又以_____符号结束函数体。每个语句结束用_____符号。
- (7) 将数据和函数捆绑在一起称为_____。
- (8) 对象的两个主要成份是_____和_____函数。

答: (1) 编辑(editor) (2) 预处理 (3) 连接(linker)
(4) 装入(loader) (5) main() (6) {},,;
(7) 封装 (8) 数据, 对数据的操作

2.2 选择题(至少选一个, 可以多选)

- (1) 对象和类的关系, 好像是_____。
 - a. 变量和数据类型的关系;
 - b. 变量和成员函数的关系;
 - c. 成员函数和基本操作的关系;
 - d. 成员函数和非成员函数的关系。
- (2) 如果一种语言能够产生新的数据类型, 那么我们就称为_____。
 - a. 是错误的;
 - b. 是可封装的;
 - c. 是可继承的;
 - d. 是可扩展的。
- (3) endl 是流操作符, 它是输入/输出流库的一部分, 其含义为_____。
 - a. end linker(结束连接);
 - b. end loader(结束装入);
 - c. end line(输出一个新行);
 - d. end level(结束层次)。
- (4) 注释是程序员理解阅读程序的良好工具。C++ 注释行有以下几种: _____。
 - a. 单行注释行//一种;
 - b. 单行注释行//和多行注释行/*...*/两种;
 - c. 没有相应的注释行;
 - d. 有三种注释行: //(单行注释), /*...*/(多行注释), /*...*/(多行注释)。

答: (1) a (2) d (3) c (4) b

2.3 C++ 对 C 作了哪些重要扩充?

答: 主要有以下 16 条重要扩充, 它们是:

- ① 分程序内的说明: 允许在分程序内和在可执行代码之后出现变量声明。

- ② `const` 说明符：既可用于一个实体在其作用域内的值说明为常量，也可用于对指针指向的数据。函数的参数也可以说明为 `const`。
- ③ `sizeof` 运算符，扩展了使用范围。以下都是合法的：`sizeof(数据类型)`，`sizeof(变量)`，`sizeof 数据类型`，`sizeof 变量`。
- ④ 强制类型转换，支持两种格式：
 类型(表达式)；
 (类型)表达式；//C 语言支持的
- ⑤ 内联(line)函数：把指定的函数直接插入到每个调用语句处，以增加内存为代价，加快程序执行速度。
- ⑥ 缺省参数：函数说明时，可以把参数中的后面 1 个或几个赋以缺省值。调用时也可以不提供这些缺省的自变量。
- ⑦ 引用参数：在参数类型后加一“&”，表示引用参数——即成为自变量的别名。
- ⑧ 函数名重载：体现了多态性。
- ⑨ `new` 和 `delete` 运算符进行内存动态分配和释放。
- ⑩ `void` 指针和 `void` 函数，表示不返回任何值。
- ⑪ 对象(object)：略。
- ⑫ 类(class)：略。
- ⑬ 成员函数，它是类在数据上的操作。
- ⑭ 构造函数和析构函数：前者完成对象的初始化工作，后者释放该对象所占的内存资源。
- ⑮ 继承：有单一继承和多重继承。
- ⑯ 命名空间和例外处理：是 C++ 国际标准中重要组成部分，扩大了应用范围和出错处理机制。

2.4 试编写一个 C++ 程序。要求从键盘输入三个整数，并打印这些数的和(sum)以及这三个整数的平均值(average)。屏幕对话应出现以下格式的画面：

Input three different integers: 13 27 14

Sum is 54

Average is 18

答：本题的 C++ 源程序如下：

```
#include <iostream>
using namespace std;
int main() {
    int x, y, z, sum, average;
    cout << "Input three different integers:";
    cin >> x >> y >> z;
    cout << "Sum is" << x + y + z << endl;
    cout << "Average is" << (x + y + z) / 3 << endl;
    return 0;
}
```

当输入 13 27 14 之后，就在屏幕上出现以下画面：

Input three different integers: 13 27 14

Sum is 54

Average is 18

2.5 试编写一个产生以下输出的 C++ 程序：

20

40

39

要求：使用一个值为 20 的整型常量并施加算术赋值运算符产生 40，然后使用递减运算符产生 39。

答：本题的 C++ 源程序如下：

```
//算术赋值和递减
#include <iostream>
using namespace std;
int main() {
    int Variable = 20;
    cout << Variable << endl;    //Variable 是 20
    Variable *= 2;                //Variable 变成 40
    cout << Variable -- << endl; //先显示 40,然后再减掉 1
    cout << Variable << endl;    //Variable 现在是 39
    return 0;
}
```

2.6 试用 C++ 程序，产生如下格式表格：

2 0 0 5 1 3 5

2 0 0 6 8 2 8 0

2 0 0 7 1 1 3 3 0

2 0 0 8 3 2 3 2 0

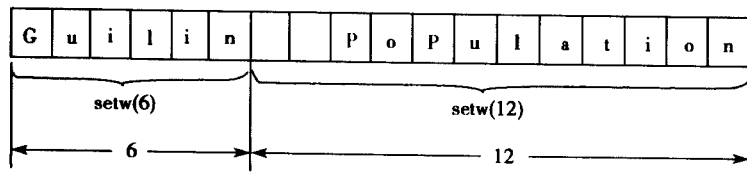
 |←8→|

要求：按上述格式表格输入、对齐；所有的输出只能用一条 cout 语句实现。

提示：为了完成上述要求，就必须用到 setw 操作符。setw 操作符可使得流中的数字（或字符串）以 n 个字符的宽度输出。setw(n) 中的 n 为参数，值在字段中是右对齐的。用 setw(n) 可以改变输出的字段的宽度。例如，下面语句

```
cout << setw(6) << "Guilin" << setw(12) << "Population" << endl;
```

其输出为：



答：本题 C++ 源程序如下：

//生成规定格式的表格

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main() {
```

```
    cout <<2005 <<setw(8) <<135 <<endl
```

```
        <<2006 <<setw(8) <<8280 <<endl
```

```
        <<2007 <<setw(8) <<11330 <<endl
```

```
        <<2008 <<setw(8) <<32320 <<endl;
```

```
    return 0;
```

```
}
```

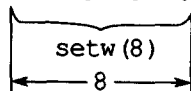
输出结果为

```
2005      135
```

```
2006      8280
```

```
2007     11330
```

```
2008     32320
```



第3章 C++ 语言基础

3.1 填空题

- (1) enum 类型中前三个枚举元的值通常是____、____和____。
- (2) C++ 编译器会忽略掉空白符这一规则的两个例外情况是____和____。
- (3) 告诉计算机该做什么的 C++ 命令, 称之为_____。
- (4) 库函数的实际代码包含在_____文件中。
- (5) 数组中的所有元素必须是_____数据类型。
- (6) C++ 字符串是一个_____类型的_____。
- (7) 在源文件中使用的数组名称表示数组的_____。
- (8) 所有程序都是根据三种类型的控制语句编制的, 它们是____、____和____。
- (9) 当条件为真时执行一个动作, 否则执行另一个动作, 是用_____选择结构。
- (10) 指定具体次数的循环称之为_____循环。

答: (1) 0, 1, 2 (2) 字符常量, 预处理指令
(3) 语句 (4) 库 (5) 相同的
(6) 数组, 字符 (7) 内存地址
(8) 顺序, 选择, 循环 (9) if/else
(10) 计算控制或有限

3.2 选择题(至少选一个, 可以多选)

- (1) 关系运算符_____。
 - a. 是逻辑运算两个操作数, 仅此而已;
 - b. 是将一个操作的结果赋予另一个操作数;
 - c. 比较两个操作数的大小, 要么相等, 要么不等;
 - d. 比较的值的类型是任何 C++ 内置类型, 并产生一个布尔结果。
- (2) 在一个包含多个语句的循环体的 for 循环中, 分号应该出现在_____后面。
 - a. for 语句自身;
 - b. 循环体中的每个语句;
 - c. 循环体的终边花括号;
 - d. 测试表达式。
- (3) 访问数组元素要采用_____。
 - a. 先进后出的方法;
 - b. 先进先出的方法;
 - c. 点运算符;
 - d. 数组下标;
 - e. 成员名称。
- (4) 访问多维数组时, 每个数组下标_____。
 - a. 用分号分隔开;
 - b. 用方括号围住再用分号分隔开;
 - c. 用分号分隔开再用方括号围住;

- d. 用方括号围住;
 - e. 用逗号分隔开。
- (5) 试问 `intArray[8]` 是指的数组中哪一个元素? _____。
- a. 第 8 个元素;
 - b. 第 7 个元素;
 - c. 第 9 个元素;
 - d. 不确定。
- (6) `string` 类的对象 _____。
- a. 以空字符作为终止标志;
 - b. 无需进行内存分配管理;
 - c. 可使用赋值运算符进行拷贝;
 - d. 它没有成员函数。
- (7) `exit()` 库函数退出 _____。
- a. 它所在的函数;
 - b. 它所在的循环;
 - c. 它所在的块;
 - d. 它所在的程序。
- (8) 逻辑运算符 `&&` 和 `||` : _____。
- a. 比较两个数字值的关系;
 - b. 运算两个数字值;
 - c. 比较两个布尔值的关系;
 - d. 运算两个布尔值。
- (9) `break` 语句退出 _____。
- a. 仅仅是最内层的循环;
 - b. 仅仅是最内层的 `switch` 语句;
 - c. 所有循环和 `switch` 语句;
 - d. 最内层的循环或 `switch` 语句。
- (10) 块内定义的变量其可见范围是 _____。
- a. 从定义点向前看的函数部分;
 - b. 从定义点向前看的程序部分;
 - c. 从定义点向前看的块的部分;
 - d. 整个函数区域。

答: (1) c, d (2) b, d (3) d (4) d (5) c
 (6) b, c (7) d (8) d (9) d (10) c

3.3 试写出 4 种不同的 C++ 语句, 使其对整型变量 `x` 都加 1。

答: 这 4 种不同的 C++ 语句是:

```
x = x + 1;
x += 1;
++x;
```

x++;

3.4 判断下面程序片段是否正确。如果错误，则要说明错在哪里，并用正确方式加以更正。

- (1) `while(c <= 5) {`
 `product *= C;`
 `++C;`
(2) `cin << value;`
(3) `if (gender == 1)`
 `cout << "People" << endl;`
 `else;`
 `cout << "Student" << endl;`
(4) `x = 1;`
 `while (x <= 10);`
 `x++;`
 `}`
(5) `for (y = 0.1; y != 1.0; y += 0.1)`
 `cout << y << endl;`
(6) `switch (n) {`
 `case 1:`
 `cout << "The number is 1" << endl;`
 `case 2:`
 `cout << "The number is 2" << endl;`
 `break;`
 `default:`
 `cout << "The number is not 1 or 2" << endl;`
 `break;`
 `}`
(7) 下面的代码应当打印 1 到 10。
 `n = 1;`
 `while (n < 10)`
 `cout << n++ << endl;`

答: (1) 错误: 丢了 while 循环的右花括号“}”。

更正: 在 ++C; 之后添上右花括号}。

(2) 错误: 流输入符号 << 用错了。

更正: 把 << 改为 >>。

(3) 错误: 在 else 后面用了分号“;”产生了逻辑错误。这样第二个输出语句永远在执行着。

更正: 去掉 else 后面的分号“;”。

(4) 错误: 在 while 头语句后面的分号“;”将产生无限循环。

更正: 去掉 while 语句后面的分号“;”，加上}。或者去掉分号“;”，并去掉

“x++;”后面的右花括号}。

(5) 错误：使用了浮点数去控制一个 for 循环结构。

更正：应使用整型数来控制执行 for 循环。

正确的语句为：

```
for (y=1;y!=10;y++)
    cout << ((float)y/10) << endl;
```

(6) 错误：在 case 1: 处丢了一个 break 语句。

更正：添加上 break 语句。请读者注意：这不是必须的。如果程序员要求每次执行完 case 1: 语句后都要执行 case 2: 语句的话，那就不是一个错误。

(7) 错误：在 while 循环中的关系运算符不合适。

更正：应该把 < 改为 <=，并把 10 改为 11。

3.5 试用穷举法找出 1 至 100 之间的质数并把它们显示出来。要求分别使用 while, do - while 和 for 循环语句加以实现。

答：1. 用 while 循环语句实现

```
#include <iostream>
using namespace std;
void main() {
    int i, j, k, signal;
    i = 2;
    while (i <= 100) {
        signal = 1;
        k = i / 2;
        while (j <= k) {
            if (i % j == 0) {
                signal = 0;
                break;
            }
            j++;
        }
        if (signal)
            cout << i << "    ";
        i++;
    }
}
```

2. 用 do[CD * 3]while 循环语句实现

```
#include <iostream>
#include <math>
using namespace std;
void main() {
```