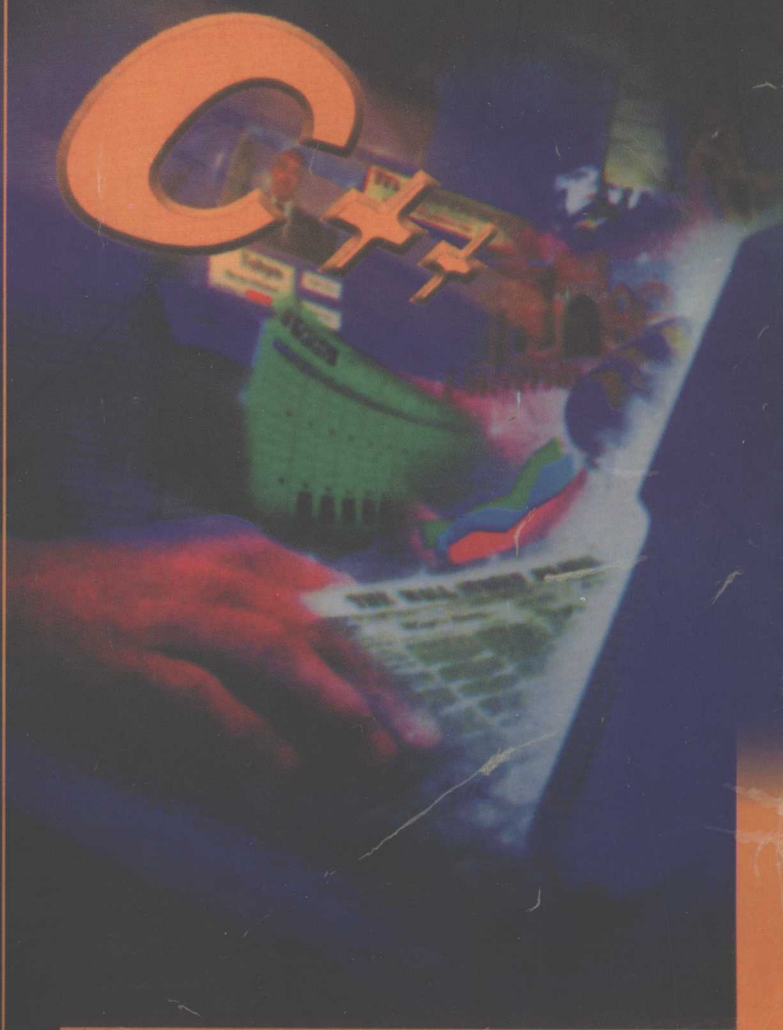


C++语言和 面向对象程序设计 学习指导

瞿裕忠 主审

刘凌波 主编

河海大学出版社



1360089

TP312/1257

C++ 语言和面向对象程序设计 学习指导

主 审 瞿裕忠

主 编 刘凌波

副主编 (以姓氏笔画为序)

万定生 吴 洁 顾韵华

徐州师范大学图书馆



22773953

0089

河海大学出版社

0800261

计算机组成原理与接口技术 + + C
第 2 版

图书在版编目 (CIP) 数据

C++ 语言和面向对象程序设计学习指导 / 刘凌波主编. — 南京:
河海大学出版社, 1999.11

ISBN 7—5630—1095—5

I. C… II. 刘… III. ①语言-程序设计②面向对
象语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 51914 号

河海大学出版社出版发行
(南京西康路 1 号 邮编:210098)

南京金阳彩色印刷厂印刷	江苏省新华书店经销
1999 年 11 月第 1 版	1999 年 11 月第 1 次印刷
开本:787×1092mm 1/16	印张:15
字数:374 千字	印数:1~5050 册
定价:18.00 元	

前 言

近年来,面向对象技术越来越广泛地被人们所接受。面向对象程序设计是在传统的结构化程序设计基础上发展而来的一种新思维、新方法,它以一种更加符合人们思维习惯的方式来解决计算机问题。面向对象方法将数据和有关操作封装在对象中,当调用对象时,只需通过接口完成,而不需要了解其内部的细节,它能够最大限度地促进软件工作者在应用领域中发挥其才能,并提高了软件的可重用性和易维护性。

C++语言在传统的结构化程序设计语言和面向对象程序设计语言之间起着承上启下的作用,它既保留了结构化程序设计的基本特征,又具有了面向对象技术的新特征。C++语言是在结构化程序设计语言——C语言的基础上发展而来的,它不仅支持和增强了C语言的功能,并且支持面向对象的类、子类、对象、抽象性和继承性等概念,是目前流行的面向对象程序设计语言之一。

本书是在宛延闾编著,清华大学出版社出版的《C++语言和面向对象程序设计》一书的基础上撰写的适合于自学考试的学习指导书,本书内容紧扣大纲,在章节安排上与自学教材同步。书中通过简洁的语言和大量的程序实例深入浅出地阐述了每一节的主要内容,概念清晰明确,通俗易懂,便于理解,在每一节后还附有大量的习题,并配有参考答案,对某些难点作了相关的分析,这为自学者理解并掌握基本概念,巩固对C++语言程序设计的理解以及提高面向对象程序设计技术均具有很好的作用。本书亦适合于大中专院校学生自学使用。

本书中的所有例题和习题中的程序均在 Visual C++ 6.0 版本的编译系统下运行过,在其他版本的 C++ 编译系统中一般也都可运行。

本书由南京经济学院刘凌波主编,河海大学万定生、南京化工大学吴洁、南京气象学院顾韵华为副主编。刘凌波编写了第一章、第三章和第六章,万定生编写了第二章和第八章,吴洁编写了第四章和第七章,顾韵华编写了第五章和第九章,全书由刘凌波统稿。本书由东南大学瞿裕忠老师审稿,在此谨表谢意。在本书编写过程中参考了许多同类书籍,在此表示诚挚的谢意。

由于时间仓促、水平有限,书中错误及不足之处在所难免,恳请广大读者及同行批评指正,以利日后改进。

编者

1999年10月于南京经济学院

目 录

第一章 绪 论	(1)
1.1 面向对象程序设计	(1)
1.2 面向对象问题解	(2)
1.3 类、子类、对象和封装	(2)
1.4 面向对象程序设计的挑战	(3)
1.5 面向对象方法	(4)
第二章 从C语言如何过渡到C++语言	(6)
2.1 C++语言:一个正在演变语言及其发展史	(6)
2.2 在较小范围内如何将C++增强C	(7)
2.3 在较大范围内如何将C++增强C	(10)
第三章 快速掌握C++语言	(12)
3.1 注释行	(12)
3.2 常量、类型和说明	(13)
3.3 C++运算符	(19)
3.4 传递引用	(24)
3.5 指针	(27)
3.6 const 说明符	(30)
3.7 void 类型	(33)
3.8 枚举类型	(37)
3.9 无名共用体(anongmous unions)	(40)
3.10 显式类型转换	(44)
3.11 函数	(48)
3.12 C++系统的文件和物理组织	(55)
第四章 C++环境中的C	(57)
4.1 函数调用	(57)
4.2 函数参数表	(61)
4.3 函数原型	(64)
4.4 引用型参数	(68)
4.5 参数个数不定	(72)
4.6 函数指针	(75)
4.7 const 类型说明符	(77)
4.8 C代码的接口	(80)
4.9 操作符引用返回值	(81)
第五章 数据封装和数据隐藏	(87)

5.1 过程语言、数据抽象、封装和数据隐藏	(87)
5.2 C++ 类	(88)
5.3 类自引用	(94)
5.4 构造函数和析构函数	(95)
5.5 作为成员的类对象	(103)
5.6 对象向量	(106)
5.7 友元(friends)	(110)
5.8 类的静态成员	(117)
5.9 运算符重载	(123)
5.10 几个基本行类	(138)
第六章 导出类和继承性	(140)
6.1 导出类构造	(140)
6.2 父类有构造函数的导出类	(147)
6.3 简单继承和多重继承	(153)
6.4 范围和访问控制	(171)
第七章 多态性和虚拟函数	(181)
7.1 虚拟函数	(181)
7.2 生成链表的一个面向对象解与多态性的异质查询树	(192)
第八章 面向对象模型	(208)
8.1 对象模型	(208)
8.2 动态模型	(210)
8.3 功能模型	(212)
第九章 面向对象设计与实现	(215)
9.1 分析	(215)
9.2 系统设计	(218)
9.3 对象设计	(223)
自学考试大纲	(225)
附录	(232)

第一章 绪 论

1.1 面向对象程序设计

【本节要点】

面向对象程序设计 (Object _ Oriented Programming, 简称 OOP) 是近几年来迅速发展的一种技术, 已成为当今软件设计的主要方法之一。它综合了功能抽象和数据抽象, 形成以对象为中心的解题过程。

对象是面向对象方法中最基本的元素, 它是分析问题、解决问题的核心。对象就是指现实世界中的一种事物, 它代表着系统中的一个实体, 如在一个教学管理系统中, 学生、教师、课程、教室等都是对象。

每个对象都具有一些属性和与之相关的事件和方法, 这些正是处理对象的途径。

对象的属性是指对象的状态或特征, 如一个学生就是一个对象, 他可以有学号、姓名、性别、出生日期等属性。不同的对象其属性也是不同的, 这是由对象所基于的实体集的性质决定的。

对象的事件是由对象识别的一个动作, 可以编写相应的程序代码来响应该动作。如当你选中某个对象单击或双击鼠标, 这时会自动执行这些事件相应的程序代码。

方法是对类定义的合法的函数, 通过执行该函数来完成一定的功能。某一类的对象可以调用该类定义的方法。对象调用类的方法称为发送一个消息给对象。

【习题及分析】

一、填空题

1. 面向对象程序设计的解题过程是以 对象 为中心。

答案: 对象

2. 对象 是面向对象方法中最基本的元素。

答案: 对象

二、问答题

1. 处理对象的途径是什么?

答案: 处理对象是通过操作处理对象的属性、事件和方法来进行的。

2. 什么是对象的属性、事件和方法?

答案: 对象的属性是指对象的状态或特征, 不同的对象其属性也是不一样的。对象的事件是由对象识别的一个动作, 可以编写相应的程序代码来响应该动作。方法是对类定义的合法的函数, 属于该类的对象可以调用它, 对象调用类的方法称为发送一个消息给对象。

1.2 面向对象问题解

【本节要点】

面向对象的程序设计语言正处于迅速发展的时期，如 C++、Smalltalk、Java 等。在分析问题、解决问题上，它和传统的非面向对象的结构化编程语言相比有着自己独特的方式。

面向对象程序设计是以对象为核心的，在处理过程中将概念和它的实现分离开，功能的实现主要是指方法的调用，方法在调用时是通过方法的接口完成的，至于方法实现的具体细节则无需了解。这样，不仅可增强系统的可靠性，也可提高面向对象软件的可重用性和可扩充性。

【习题及分析】

一、填空题

1. 面向对象程序设计中，在调用方法时无需了解方法实现的具体细节。

答案：方法实现

2. 调用方法是通过方法的接口完成的。

答案：方法的接口

二、问答题

1. 面向对象程序设计语言有哪些？

答案：面向对象程序设计语言有：C++、Smalltalk、Java 等。

2. 面向对象方法的解题方式的优点是什么？

答案：可增强系统的可靠性，可提高面向对象软件的可重用性和可扩充性。

1.3 类、子类、对象和封装

【本节要点】

一、类的概念

类是面向对象技术中的重要概念。类是创建对象的模板，它刻画了一组具有相同的结构、操作和遵守相同规则的对象。对象是类的一个实例，其中定义了对对象的属性、事件和方法。

系统提供了一些类，用户也可以创建自己新的类。

二、封装性

封装性是类的特征之一。类的封装性是指隐藏类中对象的内部数据或操作细节，它隐藏了操作对象内部的复杂性，使之与其他对象之间有着明显的范围和边界。

类的封装性提高了软件的模块化，在操作对象时，可直接使用它提供的接口，而无需了解其内部实现的具体细节。

三、子类

利用类的继承性和多态性产生了一个新的概念——子类。子类是从某一个类的基础上建立的新的类，在 C++ 中，子类又称为导出类，产生子类的类称为父类。

类的继承性使子类可以继承父类的特征，根据需要，子类也可以定义自己的特征。若父

类的某一些特征改变了,则子类相应继承这些改变后的新特征,而不需修改子类。类的继承性增加了资源的共享性,使系统更易于扩充。

类的多态性是指在不同的类中包含着同名的成员函数,这些成员函数的功能可能不同,在调用这个名称的成员函数时,要根据对象所属的类的不同,调用不同的成员函数,相应得出不同的结果。类的多态性使同样的操作可以作用于多种类型的对象上,以增加系统的灵活性。

四、对象

对象是类的一个实例,它具有所在类定义的属性和方法,但同一个类所说明的不同对象可以具有不同的属性值。

【习题及分析】

一、填空题

1. 类的特征有封装性、继承性和多态性。

答案:封装性、继承性、多态性

2. 对象是类的一个实例。

答案:对象

3. 子类是根据类的继承性和多态性特征产生的。

答案:继承性、多态性

二、问答题

1. 什么是类的概念?

答案:类是创建对象的模板,它刻画了一组具有相同的结构、操作和遵守相同规则的对象。

2. 什么是类的封装性和继承性?

答案:类的封装性是指隐藏了类中对象的内部数据或操作细节,使操作对象的内部复杂性与其他部分隔离开来。类的继承性是指子类能够继承父类的属性和功能,父类发生改变,子类也继承这些改变。

1.4 面向对象程序设计的挑战

【本节要点】

面向对象程序设计与传统的结构化程序设计有很大的区别。面向对象程序设计首先是将软件系统划分成类,这些类具有某些共同的特性,对于一些具体的不同点,则通过建立子类来进行处理。根据需要,可以对已存在的软件系统增加新的功能,但要处理好将新功能增加到面向对象系统什么地方的问题。

建立类和子类有两种方法:(1)自顶向下分解方法;(2)自底向上分解方法。自顶向下分解方法是将系统划分成多个类,每个类再划分成更特殊的子类,这样一层层划分下去,直到子类不可被划分为止。自底向上分解方法是将多个子类抽象提取成一个父类,再将各个父类抽象成更一般的父类,直至成为整个系统。在实践中,通常是两种方法共同使用。

典型的软件开发包含三个独立活动:(1)分析和设计活动;(2)程序设计;(3)用另外的语言和概念模型定义和存取数据库。它们三者之间有阻隔。采用无保留库的软件开发过程

则可以改变这种现状，从而产生高生产率、高质量和高灵活性。

传统的结构化程序设计遵循的是“算法+数据结构=程序设计”的思想，面向对象程序设计则是对这种思想的挑战，它是以封装和数据隐藏为基本要素，使数据更易于操作、维护和扩充。

【习题及分析】

一、填空题

1. 建立类和子类的两种方法是____和____。

答案：自顶向下分解方法、自底向上分解方法

2. 传统的结构化程序设计遵循的思想是“____”。

答案：算法+数据结构=程序设计

3. 面向对象程序设计是以封装和数据隐藏为基本要素。

答案：封装、数据隐藏

二、问答题

1. 什么是自顶向下分解方法？

答案：自顶向下分解方法是將系统划分成多个类，每个类再划分成更特殊的子类，这样一层层划分下去，直至子类不可被划分为止。

2. 典型的软件开发包含哪三个独立活动？

答案：典型的软件开发包含以下三个独立活动：（1）分析和设计活动；（2）程序设计；（3）用另外的语言和概念模型定义和存取数据库。

1.5 面向对象方法

【本节要点】

一、面向对象的概念

面向对象系统是以对象为基本元素，现实世界中，一个人、一棵树、一篇文章等都可以是一个对象，它代表了系统中的—个实体，并同时封装了该实体的状态和操作，与外部的联系是通过操作接口进行的。

通过对对象的数据加以抽象，形成类，类是对象的抽象，对象是类的实例。同一类的对象具有一些相同的属性，但每一个对象又具有其自身独有的属性，以区别于其他对象。

类的继承性使在类的基础上又可定义新的子类，子类继承了该类的属性，但它又具有自己的特性。

二、面向对象开发方法

面向对象开发是在传统的面向过程程序开发的基础上的一个重大发展。

面向对象方法的基本特征有：

（1）对象是数据和有关操作的封装体。

（2）面向对象方法的继承性体现了概念分离抽象。

（3）面向对象方法用消息将对象动态链接在一起。

（4）面向对象方法具有信息隐藏性。

三、面向对象方法学

对象模型技术 (Object Modeling Technique, 简称 OMT) 是一种典型的面向对象开发方法, 它分为以下几个阶段:

- (1) 分析阶段: 提取一个系统的需求的过程。
- (2) 系统设计: 对全局结构的高层进行决策。
- (3) 对象设计: 建立基于分析模型的设计模型。
- (4) 实现阶段: 进行具体的程序设计和数据库或硬件的实现。

四、三种模型

在 OMT 方法学中描述一个系统采用以下三种模型:

- (1) 对象模型: 描述了系统中的对象和它们之间联系的静态结构。
- (2) 动态模型: 描述了一个系统的对象的事件、状态和响应的时序性质。
- (3) 功能模型: 描述了系统内部数据的操作变化, 它是对对象模型和动态模型的补充。

【习题及分析】

一、填空题

1. OMT 方法分为以下几个阶段: (1) ____; (2) ____; (3) ____; (4) ____。

答案: 分析阶段、系统设计、对象设计、实现阶段

2. 一种典型的面向对象开发方法是____, 简称____。

答案: 对象模型技术、OMT

3. OMT 方法用____、____和____三种模型来描述一个系统。

答案: 对象模型、动态模型、功能模型

二、问答题

1. 简述面向对象方法的基本特征。

答案: 面向对象方法的基本特征有:

- (1) 对象是数据和有关操作的封装体。
- (2) 面向对象方法的继承性体现了概念分离抽象。
- (3) 面向对象方法用消息将对象动态链接在一起。
- (4) 面向对象方法具有信息隐藏性。

第二章 从 C 语言如何过渡到 C++ 语言

2.1 C++ 语言：一个正在演变的语言及其发展史

【本节要点】

本节介绍了 C++ 的发展历史，主要了解：

1. C++ 的基本特点

- (1) C++ 是 C 的超级集合；
- (2) C++ 是一种混合性语言；
- (3) C++ 是面向对象的程序设计语言。

2. C++ 的发展

- (1) C++ 是由 BCPL 和 Simaula67 的某些灵感而导致产生的；
- (2) 1980 年由贝尔实验室创建；
- (3) C++ 语言标准化工作从 1989 年开始。

3. 自顶向下的程序设计方法和自底向上的程序设计方法

(1) 自顶向下的程序设计方法是把大问题分解成许多相对小的问题来求解，使用同样的方法分解小的问题，直到完全了解问题的解决方法。

(2) 自底向上的程序设计方法是从基本的、简单的问题入手，在此基础上建立解决复杂问题的能力，直到整个问题的全面解决。

【习题及分析】

一、选择题

1. C++ 支持过程程序设计方法和 A 设计方法。

- A. 面向对象 B. 面向函数 C. 逻辑程序 D. 面向问题

答案：A

2. C++ 语言最有意义的方面是支持 C 特征。

- A. 面向事件 B. 面向程序 C. 面向对象 D. 面向用户

答案：C

二、填空题

1. C++ 语言是基于 C 语言，是该语言的一个 超级 集合。

答案：C、超级

2. 面向对象的语言中，可扩充性 是关键。

答案：可扩充性

三、判断题

1. C++ 语言程序员只能采用自顶向下的程序设计方法。

答案：错误

四、问答题

1. 在面向对象程序设计时, 为什么鼓励程序员采用自顶向下和自底向上相结合的程序设计方法?

答案: 在面向对象程序设计时, 要求程序员从问题的基本的简单的方面入手, 用对象的概念来描述问题的基本特征, 这时需要采用自底向上的程序设计方法。同时, 面向对象程序设计时, 又要求程序员面向问题的目标, 为了实现目标, 该如何建立基本对象, 这又体现了自顶向下的程序设计方法。

2.2 在较小范围内如何将 C++ 增强 C

【本节要点】

本节介绍了 C++ 的非面向对象的特点, 主要了解:

- (1) 注释行;
- (2) 枚举名;
- (3) 结构体或类名;
- (4) 块内说明;
- (5) 作用域限定运算符;
- (6) const 说明符;
- (7) 无名共用体;
- (8) 显式类型转换;
- (9) 函数原型;
- (10) 函数名重载;
- (11) 函数参数的缺省值;
- (12) 具有不确定的参数个数的函数;
- (13) 函数中引用参数;
- (14) inline 说明;
- (15) new 和 delete 运算符;
- (16) 指向 void 的指针和返回 void 函数。

【习题及分析】

一、选择题

1. 在结构体或类名前, C++ 中使用限定词 struct 或 A 是不必要的。

- A. class B. public C. rem D. void

答案: A

2. C++ 语言中允许同一程序中有相同函数名存在, 称为 C。

- A. 显式类型转换 B. 结构体 C. 函数重载 D. 函数原型

答案: C

3. C++ 语言中提供内存申请运算符 B, 它能可靠地控制内存的分配。

- A. delete B. new C. pos D. auto

答案: B

二、填空题

1. C++ 语言程序中常用的注释为: /* ... */, 另外一种注释为: //。

答案: //

2. C++ 语言程序中提供两种形式的常量说明: #define 语句和 const 说明符。

答案: const

三、判断题

1. C++ 中必须在枚举类型名前使用限定词 enum。

答案: 错误

2. 语句 for (int i=0; i<10; i++) 在 C++ 中是可以的。✓

答案: 正确

3. 函数说明语句: int sum (int n) 在 C++ 中是错误的。✗

答案: 错误

4. C++ 函数中参数能够指定缺省值。✓

答案: 正确

5. 不要使用过多的 inline 函数, 否则 EXE 文件会过份增大。✓

答案: 正确

四、读程序并写出运行结果

1. #include <iostream.h>

```
int sum;
```

```
void main ( )
```

```
{
```

```
    double sum;
```

```
    sum = 10.2;
```

```
    ::sum = 12;
```

```
    cout << "local sum =" << sum << endl;
```

```
    cout << "global sum =" << ::sum << endl;
```

```
}
```

答案: local sum = 10.2

global sum = 12

2. #include <iostream.h>

```
void inc (int &i)
```

```
{i=i+1;}
```

```
void inc (double &x)
```

```
{x=x+1;}
```

```
void inc (char &c)
```

```
{c=c+1;}
```

```
void main ( )
```

```
{
```

```
    char C = 'A';
```

```

int I=10;
double X=10.2;
cout << "C=" << C << endl << "I=" << I << endl << "X=" << X << endl;
inc (C);
inc (I);
inc (X);
cout << "After using the overloaded inc functions" << endl;
cout << "C=" << C << endl;
cout << "I=" << I << endl;
cout << "X=" << X << endl;

```

答案: C=A

I=10

X=10.2

After using the overloaded inc functions

C=B

I=11

X=11.2

3. #include <iostream.h>

#include <math.h>

inline double sqr (double x)

{return x * x;}

double distance (double x1, double y1, double x2=0, double y2=0)

{return sqrt (sqr (x1 - x2) + sqr (y1 - y2));}

void main ()

double x1, x2, y1, y2;

cout << "Enter x coordinate for point 1:";

cin >> x1;

cout << "Enter y coordinate for point 1:";

cin >> y1;

cout << "Enter x coordinate for point 2:";

cin >> x2;

cout << "Enter y coordinate for point 2:";

cin >> y2;

cout << "distance between point =" << distance (x1, y1, x2, y2) << endl;

cout << "distance between point 1 and (0, 0) =" << distance (x1, y1, 0, 0) << endl;

```
cout << "distance between point 2 and (0, 0) ="
    << distance (x2, y2) << endl;
}
```

答案: Enter x coordinate for point 1: 1

Enter y coordinate for point 1: 1

Enter x coordinate for point 2: -1

Enter y coordinate for point 2: 1

distance between point = 2

distance between point 1 and (0, 0) = 1.1421

distance between point 2 and (0, 0) = 1.1421

分析: 程序中有 2 个头文件, 其中使用 "math. h" 文件的原因是由于程序中使用了平方根数学函数 sqrt。inline double sqr (double x) 说明了一个内联函数 sqr, 该函数返回参数 x 的平方值。程序中说明了有 4 个 double 类型参数的函数 distance, 参数 x1 和 y1 分别代表第 1 个点的 x 和 y 坐标, 参数 x2 和 y2 分别代表第 2 个点的 x 和 y 坐标, 函数返回两个点之间的距离。

2.3 在较大范围内如何将 C++ 增强 C

【本节要点】

本节介绍了 C++ 的面向对象的特点, 主要了解:

- (1) 类的构造和数据封装;
- (2) 结构 (struct) 是一个特殊类;
- (3) 构造函数和析构函数;
- (4) 私有、保护和公共部分;
- (5) 对象和信息;
- (6) 友元 (friends);
- (7) 类中运算符和函数名重载;
- (8) 导出类;
- (9) 虚拟函数;
- (10) 流库。

【习题及分析】

一、选择题

1. 下列说法中错误的是 D。

A. 类是一种数据结构 ✓

B. 类是建立对象的关键 ✓

C. 类是数据抽象和数据隐藏的工具 ✓

D. C++ 语言中结构不是一种类

答案: D

2. C++ 语言中 class 和 struct 的区别仅在于 struct 缺省 D。

A. 私有部分

B. 函数成员

C. 数据成员

D. 存取权限

答案: D

3. C++ 语言中提供 C，它与类的成员函数具有相同的特权。

A. 构造函数 B. 析构函数 C. 友元函数 D. 虚拟函数

答案：C

二、填空题

1. C++ 语言中类作为一种数据类型与普通的数据类型的主要区别在于：类可以拥有 数据成员 和 成员函数。

答案：成员数据、成员函数

三、判断题

1. C++ 中不允许运算符重载。✗

答案：错误

2. 可以继承是 C++ 中类的重要特征。✓

答案：正确

3. 虚拟函数支持多态性，其声明规则是“一旦为虚，总是为虚”。✓

答案：正确

4. 类 cin 和 cout 是流库中的类。✓

答案：正确

四、问答题

1. 构造函数和析构函数的功能是什么？

答案：C++ 中构造函数和析构函数的主要功能是自动建立类实例和消除类实例。构造函数是一种特殊的成员函数，用来在内存中建立具体的对象。构造函数必须初始化成员变量，申请必要的内存空间，并且将内存转化成对象。与构造函数相反的是析构函数，它是一种当对象被取消时才被调用的成员函数，用来释放一个对象相关联的存储区。