

全国高等教育自学考试计算机及应用专业（独立本科段）

# C++ 程序设计 自学考试大纲

全国高等教育自学考试指导委员会制定

## 出版前言

为了适应社会主义现代化建设事业对培养人才的需要,我国在20世纪80年代初建立了高等教育自学考试制度。高等教育自学考试是个人自学、社会助学和国家考试相结合的一种高等教育形式,是我国高等教育体系的重要组成部分。实行高等教育自学考试制度,是落实宪法规定的“鼓励自学成才”的重要措施,是提高中华民族思想道德和科学文化素质的需要,也是培养和选拔人才的一种途径。自学考试应考者通过规定的专业课程考试并经思想品德鉴定达到毕业要求的,可以获得毕业证书,国家承认学历,并按照规定享有与普通高等学校毕业生同等的有关待遇。经过20多年的发展,高等教育自学考试已成为我国高等教育基本制度之一,为国家培养造就了大批专门人才。

高等教育自学考试是标准参照性考试。为科学、合理地制定高等教育自学考试的考试标准,保证教育质量,全国高等教育自学考试指导委员会(以下简称“全国考委”)按照国务院发布的《高等教育自学考试暂行条例》的规定,组织各方面的专家,根据自学考试发展的实际情况,对高等教育自学考试专业设置进行了研究,逐步调整、统一了专业设置标准,并陆续制定了相应的专业考试计划。在此基础上,全国考委各专业委员会按照专业考试计划的要求,从培养和选拔人才的需要出发,组织编写了相应专业的课程自学考试大纲,进一步规定了课程学习和考试的内容与范围,使考试标准更加规范、具体和明确,以利于社会助学和个人自学。

为更好地贯彻十七大和全国考委五届二次会议精神,适应社会经济发展的需要,反映自学考试专业建设和学科内容的发展变化,全国考委电子电工及信息类专业委员会参照普通高等学校相关课程的教学基本要求,结合自学考试计算机及应用专业考试工作的实践,组织修订了《C++程序设计自学考试大纲》,现经教育部批准,颁发施行。

《C++程序设计自学考试大纲》是该课程编写及修订教材和自学辅导材料的依据,也是个人自学,社会助学和国家考试的依据,各地教育部门、考试机构应认真贯彻执行。

全国高等教育自学考试指导委员会

二〇〇七年十月

# 目 录

## 出版前言

一、课程性质与设置目的	236
(一) 本课程的性质和特点	236
(二) 本课程的基本要求	236
(三) 本课程与相关课程的联系与分工	237
二、课程内容与考核目标	237
第1章 认识C++的对象	237
第2章 从结构到类的演变	238
第3章 函数和函数模板	238
第4章 类和对象	239
第5章 特殊函数和成员	241
第6章 继承和派生	241
第7章 类模板与向量	242
第8章 多态性和虚函数	242
第9章 运算符重载及流类库	243
第10章 面向对象设计实例	244
实践环节	244
三、有关大纲的说明与考核实施要求	246
附录 题型举例	249
后记	252

## 一、课程性质与设置目的

为了明确C++程序设计课程的学习内容和考试要求,制定此考试大纲。本考试大纲是C++程序设计(独立本科段)课程的个人自学、社会助学和考试命题的依据,本课程的考试范围以本考试大纲所限定的内容为准。

### (一) 本课程的性质和特点

C++程序设计课程在计算机及其应用专业(本科)的自学考试计划中,是一门重要的专业课。它是为满足计算机应用领域对计算机应用人才的需要而设置的。本课程的任务是通过学习,使考生为以后学习软件工程和Java语言等后继课程及开展课程设计打下必备的基础,并且为以后从事应用软件开发提供合适的工具。

设置本课程的主要目的是使考生掌握使用对象思考、实现面向对象特性的编程方法,提高解决实际问题的基本能力,为进行各种实用程序的开发奠定基础。

### (二) 本课程的基本要求

独立本科段要求读者已经学过C语言,所以本书重点放在C++程序设计方法上,将对象贯穿于每一章,强化对象的概念,以利于概念的建立和应用。STL编程是发展的方向,但STL编程是利用类和模板的、基于对象的面向过程的编程方式,所以如何使用对象进行过程编程,也是学习内容的一部分。本书不介绍STL的理论问题,而是将基本操作分散到几个章节,并以最常用的向量为例,说明其最基本的使用方法。

1. 第1章、第2章主要是了解对象和STL库概念,通过使用类和对象,了解类的基本概念以及面向对象的思想;通过实例对C++语言有别于C语言的各种成分有较好的理解,为学习如何建立类打下基础。

2. 第3章主要是熟悉并掌握函数知识。类大量使用成员函数,将成员函数的基本要求融合在第3章介绍,以便在介绍类时集中于类的结构,所以第3章是本书的重要基础知识,必须熟练掌握。本章还要求将函数知识推广到函数模板。

3. 第4章要求深入理解和掌握面向对象的核心概念——类和对象;掌握建立和使用类的基本方法;学会使用UML描述类。

4. 第5章要求掌握通过为类添加各种特殊成员函数设计类的基本方法,并初步熟悉使用对象解题的思路。

5. 第6章要求能使用继承和派生设计合理的类,深刻理解赋值兼容性规则,以便为理解虚函数的多态性打下基础。

6. 第7章是类模板与向量容器。类模板表达具有相同处理方法的数据对象集,是实现STL库的基础。向量容量vector是使用最广泛的容器之一,它克服了数组的局限性。要求理解并掌握使用向量容器类的基础知识,以便为学习STL库和泛型算法打下基础。

7. 第8章要求重点理解通过虚函数实现多态性的条件。

8. 第9章要求熟悉并掌握运算符重载的方法,掌握文件流的基本使用方法。

9. 第10章要求理解在实际应用中碰到的一些具体问题。在自学过程中,可以将本章知识先行浏览,以便有助于前面章节的理解。



本课程是一门实践性很强的课程,要求考生不仅要掌握编程方法,而且能在计算机上调试和测试一般的程序。要求考生能切实掌握C++程序设计语言作为实际工作中的工具,并为以后学习其他课程打下基础。

### (三) 本课程与相关课程的联系与分工

1. 本课程的先修课程为高级语言程序设计和计算机系统结构,以便对程序设计、计算机结构、软硬件等概念有一个初步的了解。

2. 本课程的后继课程是软件工程和Java语言。C++程序设计(实践)是课程设计的任选课程之一,学好本门课程将有助于课程设计。

## 二、课程内容与考核目标

### 第1章 认识C++的对象

#### (一) 课程内容

- 1.1 初识C++的函数和对象
- 1.2 认识C++语言面向过程编程的特点
- 1.3 程序的编辑、编译和运行的基本概念

#### (二) 学习目的与要求

本章目的是引入C++如何兼容面向过程设计、C++的基本程序结构及其在函数设计上与C语言的异同。

本章要求熟悉C++的基本程序结构。初步理解面向对象程序设计的思想及C++语言中的新思想。要求初步掌握使用函数重载的方法,以便为下一章从结构引入类知识打下基础。

重点:函数原型、重载、动态内存分配、引用、对指针使用const限定符、泛型算法、C++程序的基本结构、数据的简单输入输出格式。

难点:引用、对指针使用const限定符、泛型算法。

#### (三) 考核知识点及考核要求

- 1.1 初识C++的函数和对象
  - 1.1.1 理解C++语言与C语言的关系,达到“领会”层次。
  - 1.1.2 熟练掌握函数原型和命名空间的使用方法,达到“综合应用”层次。
  - 1.1.3 理解对象和预处理等基本概念,达到“领会”层次。
  - 1.1.4 熟练掌握C++程序的基本结构,达到“综合应用”层次。
- 1.2 认识C++语言面向过程编程的特点
  - 1.2.1 理解函数重载,达到“领会”层次。
  - 1.2.2 掌握C++新的基本数据类型,达到“简单应用”层次。
  - 1.2.3 熟练掌握动态分配内存,达到“综合应用”层次。
  - 1.2.4 熟练掌握引用的使用方法,达到“综合应用”层次。
  - 1.2.5 了解对指针使用const限定符的目的和方法,达到“识记”层次。
  - 1.2.6 了解泛型算法应用于普通数组的方法,达到“识记”层次。

- 1.2.7 掌握数据的简单输入输出格式，达到“简单应用”层次。
- 1.3 程序的编辑、编译和运行的基本方法，达到“简单应用”层次。
  - 1.3.1 掌握 Visual C++ 的工程和文件的产生方法。
  - 1.3.2 掌握程序的编辑、编译和运行的基本方法。

## 第2章 从结构到类的演变

### （一）课程内容

- 2.1 结构的演化
- 2.2 从结构演变一个简单的类
- 2.3 面向过程与面向对象
- 2.4 C++ 面向对象程序设计特点
- 2.5 使用类和对象
- 2.6 string 对象数组与泛型算法

### （二）学习目的与要求

本章的目的是通过实例说明结构如何向类变化，从而引入对象和类的知识。要求通过使用C++标准程序库提供的两个典型的类，进一步熟悉类和对象，建立面向对象的基本概念。

重点：引入面向对象程序设计的基本概念，熟悉 string 类的使用方法。

难点：使用对象及泛型算法。

### （三）考核知识点及考核要求

- 2.1 结构的演化，达到“领会”层次。
  - 2.1.1 理解函数与数据共存的意义。
  - 2.1.2 理解使用构造函数初始化结构对象的含义。
- 2.2 从结构演变一个简单的类，达到“领会”层次。
- 2.3 面向过程与面向对象，达到“领会”层次。
- 2.4 C++ 面向对象程序设计特点，达到“领会”层次。
  - 2.4.1 理解对象的含义。
  - 2.4.2 理解抽象和类的含义。
  - 2.4.3 理解封装、继承和多态性的含义。
- 2.5 熟悉并使用类和对象
  - 2.5.1 掌握使用 string 对象的基本方法，达到“简单应用”层次。
  - 2.5.2 掌握 string 类的典型成员函数的使用方法，达到“简单应用”层次。
  - 2.5.3 理解使用 complex 对象的方法，达到“领会”层次。
- 2.6 了解 string 对象数组与泛型算法，达到“识记”层次。

## 第3章 函数和函数模板

### （一）课程内容

- 3.1 函数的参数及其传递方式
- 3.2 深入讨论函数返回值



### 3.3 内联函数

### 3.4 函数重载和默认参数

### 3.5 函数模板

## (二) 学习目的与要求

因为在面向对象程序设计中, 成员函数也是函数, 所以关于函数的讨论, 也适合类的成员函数。本章的目的是将一般成员函数的特征集中在函数中讨论, 以便为设计类打下基础。本章的知识很重要, 要求能够熟练、正确地设计函数原型和参数类型, 并将其推广到函数模板。

本章除了要求掌握函数参数的传递方式和返回值等问题之外, 还要求掌握内联函数、函数重载和函数模板。

重点: 函数参数的传递方式及其返回值、函数重载、默认参数和函数模板。

难点: 函数重载和默认参数。

## (三) 考核知识点及考核要求

### 3.1 函数的参数及其传递方式

3.1.1 熟练掌握传值的使用方法, 达到“综合应用”层次。

3.1.2 熟练掌握传引用方式的使用方法, 达到“综合应用”层次。

3.1.3 掌握使用默认参数的方法, 达到“简单应用”层次。

3.1.4 掌握使用 const 保护数据的方法, 达到“简单应用”层次。

### 3.2 深入讨论函数返回值

3.2.1 返回引用的函数, 达到“简单应用”层次。

3.2.2 返回指针的函数, 达到“简单应用”层次。

3.2.3 返回对象的函数, 达到“简单应用”层次。

3.2.4 函数返回值作为函数的参数, 达到“识记”层次。

### 3.3 内联函数, 达到“简单应用”层次。

### 3.4 函数重载和默认参数, 达到“简单应用”层次。

3.4.1 深刻理解函数重载的概念和方法。

3.4.2 熟练掌握构造函数默认参数的方法。

### 3.5 函数模板, 达到“简单应用”层次。

3.5.1 深刻理解函数模板的作用。

3.5.2 熟练掌握使用函数模板的方法。

## 第4章 类和对象

### (一) 课程内容

#### 4.1 类及其实例化

#### 4.2 构造函数

#### 4.3 析构函数

#### 4.4 调用复制构造函数的综合实例

#### 4.5 成员函数重载及默认参数

#### 4.6 this 指针

4.7 一个类的对象作为另一个类的成员

4.8 类和对象的性质

4.9 面向对象的标记图

4.10 面向对象编程的文件规范

## (二) 学习目的与要求

本章的目的是建立类和对象的概念,要求重点掌握在C++中定义类、建立和使用对象的基本方法。学习时注意与第3章的知识相配合。

重点:类的定义方法、构造函数、复制构造函数、析构函数、成员函数重载及默认参数。

难点:一个类作为另一个类的成员、this指针、UML图和面向对象编程的文件规范。

## (三) 考核知识点及考核要求

4.1 类及其实例化,达到“综合应用”层次。

4.1.1 熟练掌握定义类的方法。

4.1.2 能正确使用类的对象。

4.1.3 深刻理解数据封装的含义并熟练掌握实现数据封装的方法。

4.2 构造函数,达到“综合应用”层次。

4.2.1 深刻理解默认构造函数及其作用。

4.2.2 熟练掌握定义构造函数的不同方法。

4.2.3 深刻理解构造函数和运算符 new 的相互关系及其使用方法。

4.2.4 正确使用构造函数的默认参数。

4.2.5 深刻理解复制构造函数的作用及其定义和使用方法。

4.3 析构函数,达到“综合应用”层次。

4.3.1 深刻理解析构函数的作用及其定义方法。

4.3.2 深刻理解析构函数和运算符 delete 的相互关系及其使用方法。

4.3.3 深刻理解默认析构函数的作用。

4.4 调用复制构造函数的综合实例,达到“领会”层次。

4.5 成员函数重载及默认参数,达到“简单应用”层次。

4.6 this指针,达到“领会”层次。

4.7 一个类的对象作为另一个类的成员,达到“简单应用”层次。

4.8 类和对象的性质,达到“综合应用”层次。

4.8.1 深刻理解类的对象的性质。

4.8.2 深刻理解类的性质。

4.9 面向对象的标记图

4.9.1 认识类和对象的UML标记图,达到“识记”层次。

4.9.2 了解对象的结构与连接,达到“识记”层次。

4.9.3 理解对象、类和消息的意义,达到“领会”层次。

4.10 面向对象编程的文件规范

4.10.1 编译指令,达到“识记”层次。

4.10.2 头文件的编写格式,达到“领会”层次。





## 第5章 特殊函数和成员

### (一) 课程内容

- 5.1 对象成员的初始化
- 5.2 静态成员
- 5.3 友元函数
- 5.4 const 对象
- 5.5 数组和类
- 5.6 指向类成员的指针
- 5.7 求解一元二次方程

### (二) 学习目的与要求

本章的目的是介绍类的一些特殊的函数。要求掌握它们的性质，并通过学习一个多文件编程范例，掌握使用面向对象思想解题的全过程。

重点：静态成员、友元、const 对象、数组和类。

难点：友元、转换函数、指向类成员函数的指针。

### (三) 考核知识点及考核要求

- 5.1 对象成员的初始化，达到“领会”层次。
- 5.2 静态成员，达到“领会”层次。
- 5.3 友元函数，达到“简单应用”层次。
- 5.4 const 对象，达到“领会”层次。
- 5.5 数组和类，达到“领会”层次。
- 5.6 指向类成员函数的指针，达到“领会”层次。
- 5.7 求解一元二次方程，达到“简单应用”层次。
  - 5.7.1 熟练掌握设计方程类及其成员函数的方法。
  - 5.7.2 熟练掌握编程实现的步骤和方法。

## 第6章 继承和派生

### (一) 课程内容

- 6.1 继承和派生的基本概念
- 6.2 单一继承
- 6.3 多重继承
- 6.4 二义性及其支配规则
- 6.5 典型问题分析

### (二) 学习目的与要求

本章的目的是介绍C++语言继承方面的语法特征，要求通过仔细研读实例，掌握派生和包含的设计方法。

重点：派生类的构造函数和析构函数、访问权限和赋值兼容规则。

难点：二义性及其支配规则。

### (三) 考核知识点及考核要求

- 6.1 继承和派生的基本概念, 达到“简单应用”层次。
- 6.2 单一继承, 达到“综合应用”层次。
  - 6.2.1 深刻理解单一继承的一般形式。
  - 6.2.2 熟练掌握派生类的构造函数和析构函数的定义方法。
  - 6.2.3 熟练掌握类的保护成员的定义方法。
  - 6.2.4 深刻理解访问权限和赋值兼容规则。
- 6.3 多重继承, 达到“识记”层次。
- 6.4 二义性及其支配规则, 达到“识记”层次。
  - 6.4.1 知道作用域运算符和成员名限定的含义。
  - 6.4.2 知道派生类支配基类的同名函数的含义。
- 6.5 典型问题分析, 达到“识记”层次。

## 第7章 类模板与向量

### (一) 课程内容

- 7.1 类模板
- 7.2 向量与泛型算法
- 7.3 出圈游戏

### (二) 学习目的与要求

本章的目的是介绍设计C++类模板及使用向量容器的基本方法, 要求掌握类模板的定义及其简单的派生方法、熟悉向量容器的定义和基本操作方法, 了解泛型算法和STL库的概念。

重点: 类模板、定义向量列表和向量容器的基本操作。

难点: 泛型指针、在向量容器指定位置插入和删除对象的方法。

### (三) 考核知识点及考核要求

- 7.1 类模板
  - 7.1.1 类模板的成分、语法及对象, 达到“简单应用”层次。
  - 7.1.2 类模板的派生与继承, 达到“领会”层次。
- 7.2 向量与泛型算法
  - 7.2.1 掌握定义向量列表的方法, 达到“简单应用”层次。
  - 7.2.2 了解泛型指针, 达到“领会”层次。
  - 7.2.3 了解向量的数据类型, 达到“领会”层次。
  - 7.2.4 掌握向量的size和push\_back成员函数的使用方法, 达到“简单应用”层次。
  - 7.2.5 了解向量其他的操作方法, 达到“识记”层次。
- 7.3 出圈游戏, 达到“识记”层次。

## 第8章 多态性和虚函数

### (一) 课程内容

- 8.1 多态性



## 8.2 虚函数

## 8.3 多重继承与虚函数

## 8.4 类成员函数指针与多态性

### (二) 学习目的与要求

本章的目的是引入C++语言的多态性和虚函数的用途,要求理解编译时的多态性和运行时的多态性,掌握多态性和虚函数的有关知识。

重点:虚函数、纯虚函数、抽象类及产生多态性的条件。

难点:虚函数实现多态性的条件、多重继承与虚函数、类成员函数指针与多态性。

### (三) 考核知识点及考核要求

#### 8.1 多态性,达到“领会”层次。

8.1.1 知道静态联编中的赋值兼容性及名字支配规律。

8.1.2 理解动态联编的多态性。

#### 8.2 虚函数

8.2.1 虚函数的定义,达到“综合应用”层次。

8.2.2 虚函数实现多态性的条件,达到“综合应用”层次。

8.2.3 构造函数和析构函数调用虚函数,达到“领会”层次。

8.2.4 纯虚函数与抽象类,达到“综合应用”层次。

#### 8.3 多重继承与虚函数,达到“识记”层次。

#### 8.4 类成员函数指针与多态性,达到“识记”层次。

## 第9章 运算符重载及流类库

### (一) 课程内容

#### 9.1 运算符重载

#### 9.2 流类库

#### 9.3 文件流

### (二) 学习目的与要求

本章的目的是学习运算符重载和文件的概念,要求掌握运算符重载的基础知识、流类库的概念及使用流类库进行文件存取的概念。希望通过文件存取综合实例,掌握进行文件存取的最基本方法。

重点:运算符重载、格式控制、文件存取。

难点:运算符重载。

### (三) 考核知识点及考核要求

#### 9.1 运算符重载

9.1.1 重载对象的赋值运算符,达到“简单应用”层次。

9.1.2 运算符重载的实质,达到“简单应用”层次。

9.1.3 <<、>>和++运算符重载实例,达到“简单应用”层次。

9.1.4 类运算符和友元运算符的区别,达到“识记”层次。

9.1.5 下标运算符“[]”的重载,达到“识记”层次。

#### 9.2 流类库

- 9.2.1 流类库的基本类等级,达到“识记”层次。
- 9.2.2 默认输入输出格式控制,达到“识记”层次。
- 9.2.3 使用 `ios_base` 类,达到“领会”层次。

### 9.3 文件流

- 9.3.1 文件流的概念,达到“简单应用”层次。
- 9.3.2 几个典型流成员函数,达到“识记”层次。
- 9.3.3 文件存取综合实例,达到“领会”层次。

## 第10章 面向对象设计实例

### (一) 课程内容

- 10.1 过程抽象和数据抽象
- 10.2 发现对象并建立对象层
- 10.3 定义数据成员和成员函数
- 10.4 如何发现基类和派生类结构
- 10.5 接口继承与实现继承
- 10.6 设计实例

### (二) 学习目的与要求

本章的目的是要求考生认识到面向对象语言和面向对象编程不能简单等同。正像不能说采用结构化程序设计语言编写的程序一定符合结构化编程的特点一样,并不能说采用面向对象语言的编程就一定具有充分的面向对象特性。要求通过本章讨论的几个专题和实例,加深对知识的理解并锻炼解决实际问题的能力。

重点:过程抽象和数据抽象、接口继承与实现继承。

难点:接口继承与实现继承。

### (三) 考核知识点及考核要求

- 10.1 过程抽象和数据抽象,达到“识记”层次。
- 10.2 发现对象并建立对象层,达到“识记”层次。
- 10.3 定义数据成员和成员函数,达到“识记”层次。
- 10.4 如何发现基类和派生类结构,达到“识记”层次。
- 10.5 接口继承与实现继承,达到“领会”层次。
- 10.6 设计实例,达到“领会”层次。

## 实践环节

### (一) 类型

课程实验

### (二) 目的与要求

本课程的实验采用 Microsoft Visual C++ 6.0。通过上机实践,加深对课程的理解,更好地掌握C++面向对象程序设计的基本方法,增强实际动手能力。

### (三) 内容

本课程给出8个可供选择的实验,它们涵盖了课程的不同内容。这些实验能加深对课文



的理解,建议尽可能多做实验。要完成实验,必须加强自学,一般自学时数为上机实验的两倍;实验对理解课文能起到事半功倍的效果。

建议必须涵盖如下3个重要实验。

1. 多文件编程实验的内容:熟悉建立工程文件、头文件和多个源文件的程序设计规范,编译运行程序。

2. 公有继承的赋值兼容性规则和虚函数多态性实验的内容:理解公有继承的赋值兼容性规则和虚函数的多态性的关系,产生多态性的条件。

3. 向量的实验内容:理解向量的基本使用方法,能用来解决实际问题。

#### (四) 与课程考试的关系

本课程实验必须在课程笔试前完成,以促进学习者掌握课程内容。实验考核应在课程笔试后择时进行,并须在主考院校或主考院校委托单位进行。应考者须提供实验报告和提交实验考核的结果。实验环节为2学分,以下为实验大纲的具体内容。

##### 实验一

(1) 实验名称: Microsoft Visual C++ 6.0 集成环境的使用。

(2) 实验目的与要求: 了解并初步掌握 Microsoft Visual C++ 6.0 集成环境。

(3) 实验内容: 要求建立工程文件 `cpp1`, 源文件 `cpp1.cpp`; 编写一个程序, 动态分配能存入 10 个 `int` 值的内存并通过键盘为这块内存赋值, 然后输出这些值的和以及最小值; 编译并运行这个程序。

注意: 也可以直接选用第1章的程序进行练习。

##### 实验二

(1) 实验名称: 使用函数模板。

(2) 实验目的与要求: 了解函数模板的概念, 练习编制函数模板。

(3) 实验内容: 用模板函数实现三个数值中按最小值到最大值的排序程序。

##### 实验三

(1) 实验名称: 使用多文件编程。

(2) 实验目的与要求: 熟悉面向对象编程思想和多文件编程规范。

(3) 实验内容

① 建立工程文件 `Point`、头文件 `Point.h` 和源文件 `Point.cpp`。

② 设计一个 `Point` 类, 这个类描述屏幕上一个点的位置, 其中成员函数 `Distance` 用于计算两个点的距离。这个类在 `Point.h` 中声明, 在 `Point.cpp` 文件中实现。

③ 在文件 `Find.cpp` 中编写主程序, 输入三角形的3个顶点坐标计算3条边的长度。

④ 编译运行程序。

⑤ 使用不同的方法编写成员函数 `Distance`, 以加深对课文的理解。

##### 实验四

(1) 实验名称: 公有继承的赋值兼容性规则。

(2) 实验目的与要求: 公有继承的赋值兼容性规则、构造函数、复制构造函数和析构函数的设计方法。

(3) 实验内容

① 设计一个 `Point` 类, 这个类描述屏幕上一个点的位置, 用它派生一个线段类 `Line`。

② 为它们设计适当的成员函数，以便演示公有继承的赋值兼容性规则，以及构造函数、复制构造函数和析构函数的调用顺序。

③ 重新使用模板设计这个实验。

#### 实验五

(1) 实验名称：使用向量。

(2) 实验目的与要求：练习使用向量解决实际问题。

(3) 实验内容：调试教材第 7.3 节的出圈游戏。

#### 实验六

(1) 实验名称：运算符重载。

(2) 实验目的与要求：熟悉运算符重载。

(3) 实验内容：重载 “--” 运算符。

#### 实验七

(1) 实验名称：文件读写。

(2) 实验目的与要求：熟悉使用文件的基本方法。

(3) 实验内容

① 为第 9.3.3 节的例子中的 student 类增加 “名字” 属性。

② 调试运行这个新的程序，重点掌握这种基本的文件存取方法。

#### 实验八

(1) 实验名称：虚函数的多态性。

(2) 实验目的与要求：熟悉继承和包含以及虚函数的多态性。

(3) 实验内容

① 调试教材第 10.6.4 节的程序。

② 理解继承和包含的区别及其各自的应用场合。

③ 通过单步跟踪，进一步理解虚函数、友元函数和重载的概念。

## 三、有关大纲的说明与考核实施要求

### (一) 自学考试大纲的目的和作用

课程自学考试大纲是根据专业自学考试计划的要求，结合自学考试的特点而确定的，其目的是对个人自学、社会助学和课程考试命题进行指导和规定。

课程自学考试大纲明确了课程学习的内容以及深广度，规定了课程自学考试的范围和标准。因此，它是编写自学考试教材和辅导书的依据，是社会助学组织进行自学辅导的依据，是自学者学习教材、掌握课程内容知识范围和程度的依据，也是进行自学考试命题的依据。

### (二) 课程自学考试大纲与教材的关系

课程自学考试大纲是进行学习和考核的依据，教材是课程知识基本内容和范围的载体，教材内容是大纲所规定的课程知识的扩展与发挥。

大纲与教材所体现的课程内容应基本一致；大纲里面的课程内容和考核知识点，教材里一般也要有。反过来教材中有的内容，大纲里就不一定体现。



### （三）关于自学教材与主要参考书

指定教材：《C++ 程序设计》，全国高等教育自学考试指导委员会组编，刘振安主编，机械工业出版社 2008 年出版。

### （四）关于自学要求和自学方法的指导

1. 本大纲的课程基本要求是依据专业考试计划和专业培养目标而确定的。课程基本要求还明确了课程的基本内容以及对基本内容的掌握程度。基本要求中的知识点构成了课程内容的主题部分。因此，课程基本内容掌握程度、课程考核知识点是高等教育自学考试考核的主要内容。

2. 在自学要求中，对各部分内容掌握程度的要求由低到高分识记、领会、简单应用、综合应用 4 个层次，其相应的表达用语依次是：了解、知道；理解、清楚；掌握、会用；深入理解、熟练掌握。为有效地指导个人自学和社会助学，本大纲指明了课程内容的重点和难点，自学时必须严格按照大纲的要求进行学习、总结、练习和自我考查。

3. 本课程是一门知识和技能并重的专业基础课，共 5 个学分，其中 2 学分为实验内容的学分。

4. 本课程是按照使用对象编程、建立并使用类和对象、深入使用类和对象、总结提高 4 个模块组织的。重点放在程序设计方法上，将对象贯穿于每一章，强化对象的概念，以利于概念的建立和应用。需要说明的是，STL 编程是发展的方向，但 STL 编程是利用类和模板的、基于对象的面向过程的编程方式，所以如何使用对象进行过程编程，也是学习的重要组成部分。建议按 4 个组成部分划分自学阶段。

第 1 部分是第 1 章～第 3 章。主要是引入对象、STL 库概念以及函数知识，重在练习如何使用类和对象。通过使用类，了解类的基本概念，从而为学习建立类打下基础。本书将函数和许多公共内容浓缩集中在第 3 章。第 3 章不仅是本书的重要基础知识，而且在以后的学习中不再讲述（学习成员函数时，可以自己回顾本章知识），所以必须熟练掌握。学习这部分也用到 C 语言的知识，虽然有似曾相识的感觉，但也要切忌学习进度过快，应该通过对比的方法加深理解，为后面各章的学习打下牢固的基础。

第 2 部分是第 4 章～第 6 章，这部分的重点是培养建立类和使用对象的方法。学习时要抓住各章的特点。第 4 章要求掌握建立类的基本方法，并注意理解使用一个类的对象作为类的数据成员的意义；第 5 章要求在第 4 章所学知识的基础上，通过特殊成员函数扩充类的功能，特别要注意掌握静态成员和友元函数；第 6 章将通过派生进一步深化，要求真正理解赋值兼容性规则。这一部分还通过实例要求考生初步熟悉面向对象方法解题的思路。注意在学习时要有第 3 章知识的配合。

第 3 部分是第 7 章～第 9 章，主要是围绕类进一步展开并扩展知识。第 7 章是想通过参数类型化设计功能更强大的类，类模板表达具有相同处理方法的数据对象集，而向量容器就是一个例子。类模板也是实现 STL 库的基础。向量容器 `vector` 克服了数组的局限性，所以获得了广泛应用。这一章看似很难，但如果仔细对照例子，尤其是辅助上机实践，应该是比较容易理解的。注意考试大纲对向量容器成员函数知识点的要求，不要面面俱到。第 8 章是多态性和虚函数，要求在设计中进一步展现面向对象程序设计的特点。第 9 章是运算符重载及流类库，这将给类的设计提供更大的舞台。

第 4 部分只有第 10 章。这一章是进一步抽象和提高，主要讨论在实际应用中需要考虑

的几个专题，并结合实例进一步说明面向对象程序设计的基本方法。

第4部分的理论和实践性均很强，所以只要求了解，并不是考试的重点。全书的重点是第3章~第9章。特别要注意的是，第3章是考试必不可少的内容，成员函数的变化均出于此，必须给予足够的重视。

5. 学习编程语言最有效手段之一就是边学边练，这能起到事半功倍的效果，应该给予足够重视。对于一些疑难问题，还可以编写针对性的程序探讨有关知识点，不仅可以达到解惑的目的，还能学习并巩固相关知识点。

#### （五）对社会助学的要求

为了学习本课程，设置社会助学是必要的。为了保证教学质量，各助学点应做到：

1. 熟知考试大纲对课程的总的要求以及各章的知识点。
2. 掌握各知识点要求达到的层次，深刻理解对各知识点的考核要求。

3. 助学辅导时应以指定教材为基础，考试大纲为依据，不得删节的内容；辅导时应把教材上的定义和例子讲深讲透，并注意引导学习者刻苦自学，钻研教材，以便掌握所要求的内容。

4. 辅导时要注意基础，突出重点，要帮助考生对课程内容建立一个整体的概念，对考生提出的问题，应以启发引导为主。

5. 注意对考生能力的培养，特别是自学能力的培养。要引导考生逐步学会独立学习，在自学过程中善于提出问题、分析问题和解决问题，最好有指导教师讲课和批改作业。

6. 辅导时应对学习方法进行指导。建议注意如下问题：

（1）应特别重视面向对象的程序设计和传统的面向过程的程序设计之间的本质区别，指导考生完成程序设计思想上的根本转变，这种助学方法应贯穿于整个教学过程中。

（2）注意指导考生区别对象和其他数据类型在“建立”，“赋值”，“撤消”等过程中的不同之处，强调“数据封装”、“私有”及“局部”等概念。

（3）“继承”是面向对象设计中最重要的一环，除讲授其观点与方法之外，还要注意编写应用程序以加深认识与理解。

（4）“虚函数”和“多态性”是程序设计的重点和难点，讲述时注意与类的“数据封装”和“公有继承的赋值兼容性规则”相结合。

（5）助学单位必须具备上机实验的条件。上机对于学习语言类课程特别重要，能起到事半功倍的效果。上机时，应注意培养学生规范的编程习惯，特别是多文件编程方法。

（6）注意安排上机实验，结合每一部分内容进行编程练习，以便加深理解并巩固所学知识。

（7）要使考生了解试题的难易程度与能力层次的高低并不完全是一回事，在各个能力层次中都存在不同难度的试题。

#### （六）对考核内容与考核目标的说明

1. 本课程要求考生学习和掌握的知识点都是考核的内容。由于各知识点在课程中的地位、作用，以及知识自身特点不同，自学考试将各个知识点分别按四个认知层次（或叫能力）确定其考核要求。

2. 四个能力层次从低到高依次是：识记、领会、简单应用、综合应用。具体解释如下：

识记：要求考生能够识别和记忆本课程中规定的有关知识点的主要内容（如定义、定





理、定律、表达式。公式、原则、重要结论、方法、步骤及特性、特点等),并能根据考核的不同要求,作出正确的表述、选择和判断。

**领会:**要求考生能够领悟和理解本课程中规定的有关知识点的内涵与外延,熟悉其内容要点和它们之间的区别与联系,并能根据考核的不同要求,作出正确的解释、说明和论述。

**简单应用:**要求考生能够运用本课程中规定的少量知识点,分析和解决一般应用问题,如简单的计算、绘图和分析、论证等。

**综合应用:**要求考生能够运用本课程中规定的多个知识点,分析和解决较复杂的应用问题,如计算、绘图、简单设计、编程和分析、论证等。

简单地说,识记,能够指出是什么;领会,要能回答为什么;简单应用和综合应用,要能回答做什么和怎么做。

### (七) 关于考试命题的若干规定

1. 本课程的考试分为笔试和实验考核两部分。笔试采用闭卷方式,时间为 150 分钟。评分采用 100 分制,60 分为及格。笔试需带笔(钢笔或圆珠笔、铅笔)、橡皮、尺子等,答卷规定用蓝色或黑色的水笔,或者黑色圆珠笔。主考院校可以提供必要的草稿纸。实验考核由各地主考院校自行安排进行。凡通过笔试和实验考核的考生方能获得本课程的单科合格证书。

2. 本大纲各章所规定的基本要求、知识点及知识点下的知识细目,都属于考核的内容。考试命题覆盖到章,并适当考虑课程重点、章节重点,加大重点内容的覆盖度。

3. 命题不应有超出大纲中考核知识点范围的题目,考核目标不得高于大纲中所规定的最高能力层次要求。命题应着重考核自学者对基本概念、基本知识和基本理论是否了解和掌握,对基本方法是否会用或熟练。不应出与基本要求不符的偏题或怪题。

4. 本课程在试卷中对不同能力层次要求的分数比例大致为:识记占 20%,领会占 30%,简单应用占 30%,综合应用占 20%。

5. 要合理安排试题的难易程度,试题的难度可分为:易、较易、较难和难四个等级。每份试卷中不同难度试题的分数比例一般为 2:3:3:2。

必须注意试题的难易程度与能力层次有一定的联系,但二者不是同等的概念。在各个能力层次中对于不同的考生都存在着不同的难度。考生不要混淆。

6. 本课程考试命题的主要题型有:单项选择、填空、改错、分析程序、完成程序 5 种。题型举例见附录。

## 附录 题型举例

### 一、填空题

在公有派生时,基类的保护成员在派生类中是\_\_\_\_\_,基类的公有成员在派生类中是\_\_\_\_\_。

### 二、单项选择题

在类中用来声明虚成员函数的关键字是( )。

A. virtual

B. public

C. protected

D. private

### 三、改错题

改正下面程序中的错误。

```
template < class T >
max ( T x, y)
{
    return (x > y)? (x): (y);
}
```

### 四、分析程序题

1. 分析下面的程序，指出虚函数并说明其成立的理由。

```
class A {
public:
    virtual int fun(int);
    int gun(int);
};
class B:public A{
public:
    int fun(int);
    virtual int gun(int);
};
```

2. 找出下面程序中的错误并说明错误的理由。

```
class base {
protected:
    int p;
};
void fun() {
    base b;
    int x = b.p;
}
```

3. 分析下面的程序，给出程序的输出结果。

```
#include <iostream>
#include <vector>
using namespace std;
void main() {
    char st[11] = "abcdefghij";
    vector< char > a( st, st + 10);
    cout << a.capacity() << " , " << a.size() << endl;
    a.pop_back();
    a.push_back('W'); a.push_back('P');
    for(j=0; j < a.size(); j++)
        cout << a[j] << " ";
```



```
cout << endl << a.capacity() << " " << a.size() << " ";
a.clear();
cout << a.empty() << endl;
```

## 五、完成程序题

1. 完成下面求两个数据中的最大值的函数模板程序。

```
template <class T>
____ max(____)
{ return(m1 > m2)? m1:m2; }
```

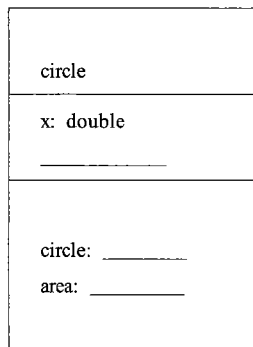
2. 一个类的头文件如下所示:

```
#include <iostream>
```

```
____
class test {
    private:
        int num;
    public:
        test(int n):
        void show();
};
____ show() { cout << num << endl; }
```

完成头文件的定义并编写一个主程序，在主程序中产生对象 a，且 a 的成员 num 等于 5，并使用 show () 函数输出这个对象的值。

3. 一个圆 circle 具有圆心坐标 x 和半径 radius 两个属性，有一个构造函数和能够给出圆面积的成员函数 area，请完成下面这个圆类的类图。



## 后 记

本大纲是根据全国高等教育自学考试指导委员会电子电工与信息类专业委员会制定的“高等教育自学考试计算机及应用专业（独立本科段）考试计划”及有关文件精神制定。

本大纲提出初稿后，由专业委员会在合肥市组织召开审稿会进行审稿，根据审稿会意见进行了修改。最后由电子电工与信息类专业委员会定稿。

本大纲由刘振安教授（中国科学技术大学）负责编写和修改，参加审稿并提出修改意见的有陈怀义教授（国防科学技术大学，主审）、陈铁年教授（上海交通大学，参审）、苏仕华副教授（中国科学技术大学，参审）。

对参加本大纲编写和审稿的教授表示感谢。

全国高等教育自学考试指导委员会

电子电工与信息类专业委员会

2007年10月