

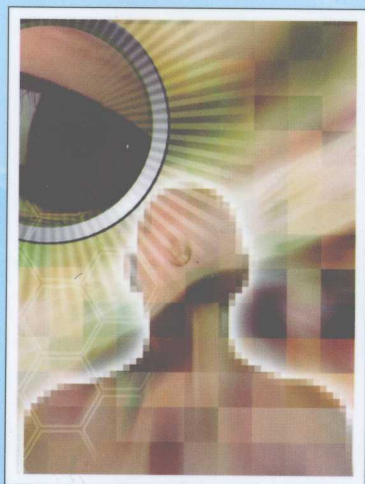


21世纪高等学校系列教材

21 Shiji Gaodeng Xuexiao Xilie Jiaocai

面向对象程序设计 及C++实验指导

朱立华 俞琼 朱建 编著



 人民邮电出版社
POSTS & TELECOM PRESS



21世纪高等学校系列教材

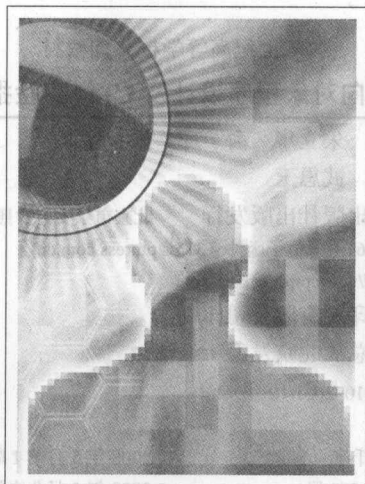
21 Shiji Gaodeng Xuexiao Xilie Jiaocai

面向对象程序设计 及C++实验指导

(1) 阅读主教材第7章全部内容，理解教材中的示例程序。

(2) 阅读本次实验内容，完成实验程序。

朱立华 俞琼 朱建 编著



人民邮电出版社

北京

图书在版编目 (CIP) 数据

面向对象程序设计及 C++ 实验指导 / 朱立华, 俞琼, 朱建编
著. —北京: 人民邮电出版社, 2008.5
(21 世纪高等学校系列教材)
ISBN 978-7-115-17035-4

I. 面… II. ①朱…②俞…③朱… 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 (2008) 第 020988 号

内 容 提 要

本书是《面向对象程序设计及 C++》教材的配套指导用书。全书由 3 部分组成: 第一部分包括主教材中的习题参考答案及解析; 第二部分给出了与主教材每一章内容配套的补充习题, 以弥补主教材因篇幅所限习题量少、题型不全面等不足, 并给出对应的参考答案; 第三部分是实验指导, 安排了 8 个与教材配套的实验, 每个实验包括详细的实验目的、实验要求、实验题目、实验指导并提出了每个实验需要预习的内容, 这些实验对初学者全面掌握面向对象的程序设计及 C++ 语言大有帮助。

本书可作为高等院校面向对象程序设计课程的辅导教材, 也适合单独作为学习 C++ 语言的参考书。

21 世纪高等学校系列教材

面向对象程序设计及 C++ 实验指导

- ◆ 编 著 朱立华 俞 琼 朱 建
责任编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 11
字数: 291 千字
印数: 1—3 500 册
- 2008 年 5 月第 1 版
2008 年 5 月北京第 1 次印刷

ISBN 978-7-115-17035-4/TP

定价: 18.00 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223
反盗版热线: (010) 67171154

前 言

无论学习何种程序设计语言, 仅仅掌握理论知识是不够的, 学习程序设计语言最终的目的是能够编写出结构清晰、功能完整的程序。因此, 程序设计语言课程是一门对实践环节要求很高的课程。初学者想要真正掌握用 C++ 语言进行面向对象的程序设计, 必须抓住两个重要环节: 一是多做习题, 通过各种习题从理论上全面掌握所学的基本知识; 二是多上机实践, 进而掌握面向对象的编程。这两个方面缺一不可, 没有坚实的基本知识作铺垫, 想要设计出高水平的程序是不可能的; 掌握基础知识的最终目的是为了能够编写出更好的程序。为此, 我们编写了这本习题解答及实验指导书。

本书共分为三个部分: 第一部分是主教材中的习题参考答案及解析, 能够帮助初学者更好地梳理所学知识。第二部分给出了与主教材每一章内容配套的补充习题。主教材由于受篇幅限制, 习题数量相对较少, 题型也比较有限。因此, 本书在这一部分, 对应每章内容给出了大量的补充习题, 丰富了题型题量, 题目几乎覆盖了教材所有的知识点, 对学生更好地掌握理论知识大有帮助。第三部分是实验指导, 精心设计了 8 个实验, 对每个实验明确了实验目的、实验要求和实验题目, 并对每一个题目提供了必要的实验指导, 同时提出了每个实验需要预习的内容。认真完成这 8 个实验, 对全面掌握面向对象的程序设计及 C++ 语言非常关键。

本书第 1 章、第 2 章和第 4 章的习题解答及补充习题和实验 1、2、5 由朱立华编写, 第 3 章和第 6 章的习题解答及补充习题和实验 3、4、7 由俞琼编写, 第 5 章和第 7 章的习题解答及补充习题和实验 6、8 由朱建编写, 全书由朱立华统稿。

由于编者水平有限, 错误和不当之处在所难免, 在此恳请广大读者批评指正。

作 者

2008 年 2 月

目 录

第一部分 教材习题参考答案与解析···1

第1章 面向对象程序设计及 C++语言概述·····	1
第2章 C++语言对 C 语言的改进及扩展·····	4
第3章 类与对象·····	11
第4章 类的继承与派生·····	22
第5章 多态性·····	33
第6章 模板·····	47
第7章 C++语言的流类库与输入/输出控制·····	53

第二部分 补充习题与解答·····62

第1章 面向对象程序设计及 C++语言概述·····	62
第2章 C++语言对 C 语言的改进及扩展·····	65
第3章 类与对象·····	77

第4章 类的继承与派生·····	103
第5章 多态性·····	121
第6章 模板·····	134
第7章 C++语言的流类库与输入/输出控制·····	141

第三部分 实验指导·····147

实验1 Visual C++6.0 集成开发环境的使用·····	147
实验2 用 C++实现面向过程的程序设计·····	149
实验3 类和对象的定义及使用(一)···	152
实验4 类和对象的定义及使用(二)···	155
实验5 继承与派生实验·····	159
实验6 多态性实验·····	164
实验7 函数模板与类模板的应用·····	166
实验8 C++的 I/O 操作及文件的使用···	168

参考文献·····172

第一部分

教材习题参考答案与解析

第 1 章 面向对象程序设计及 C++ 语言概述

一、单选题

1. 下列各种高级语言中,不是面向对象的程序设计语言的是_____。

- A. C++ B. Java C. C D. Smalltalk

【参考答案】C

【解析】C++、Java、Smalltalk 都是面向对象的程序设计语言,而 C 语言是一种面向过程的程序设计语言。

2. 下列关于类与对象关系的描述中,不正确的是_____。

- A. 类是具有相同属性和行为的一类对象的抽象
B. 对象是类的具体实体
C. 类与对象在内存中均占有内存单元
D. 类是创建对象的模板

【参考答案】C

【解析】类与对象是一组抽象与具体的关系,类实际上是类型,对象实际上是属于类类型的变量,因此只有变量才会占有内存空间,类型不占内存空间。

3. 下列哪一个不是面向对象方法的特征_____。

- A. 开放性 B. 封装性 C. 继承性 D. 多态性

【参考答案】A

【解析】封装性、继承性和多态性是面向对象方法的三大特征。

4. C++源程序文件的扩展名是_____。

- A. .obj B. .cpp C. .exe D. .c

【参考答案】B

【解析】C++源程序的扩展名为 .cpp,而 C 语言源程序的扩展名为 .c,C++和 C 语言源程序经编译后生成的目标文件扩展名为 .obj,链接后的可执行文件扩展名为 .exe。

5. 下列关于对象的描述中,不正确的是_____。

- A. 对象是现实世界中客观存在的某种实体

- B. 对象是类的实例
- C. 对象就是 C 语言中的结构体变量
- D. 对象是属性和行为的封装体

【参考答案】C

【解析】现实世界由一个个对象构成，因此对象是现实世界中客观存在的实体；根据对象与类之间的关系，对象是类的实例，是属性和行为实际的封装体。对象所属的类类型是一种新类型，与结构体类型有区别，因此对象不是结构体变量。

二、填空题

1. C++语言既支持_____的程序设计，又支持_____的程序设计。

【参考答案】面向过程、面向对象

【解析】C++语言在 C 语言的基础上扩充，是带类的 C，是同时支持面向过程程序设计和面向对象程序设计的混合型语言。

2. 开发 C++程序应该包括的 4 个步骤依次是：_____、_____、_____和_____。

【参考答案】编辑、编译、链接、运行

【解析】开发 C++程序与开发其他高级语言的程序一样，包括编辑、编译、链接和运行几个步骤。

3. C++源程序和 C 源程序一样，有且仅有 1 个_____函数。

【参考答案】main

【解析】main()函数作为操作系统调用用户程序的入口，必须有且只能有一个。

4. 在 VC++ 6.0 环境中，C++程序无论是单文件还是多文件，都应该在一个_____下管理。

【参考答案】项目

【解析】C++程序可以是单个文件，也可以有多个文件。在 VC++6 环境下，都要在项目中管理程序。

5. VC++ 6.0 集成开发环境中的_____几乎可以帮助找到程序开发中可能产生的所有错误。

【参考答案】调试器

【解析】调试器是 VC++6 系统中最出色的组件之一，它几乎可以帮助找到程序开发中可能产生的所有错误。

三、问答题

1. 简述 C++语言与 C 语言的关系。

【参考答案】C++语言在传统 C 语言的基础上进行改造和扩充，引入了面向对象的概念和方法，增加了对面向对象程序设计的支持。C++语言是同时支持面向过程程序设计和面向对象程序设计的混合型语言。

在支持面向过程的程序设计方面，C++语言首先继承了 C 语言，与 C 语言兼容，C 语言是 C++语言的一个子集。C 语言的词法、语法和其他规则都可以用到 C++语言中。但同时，C++语言又针对 C 语言的某些不足做了改进。例如，用流更方便地实现输入/输出操作；用 const 定义常量取代宏；允许函数重载、函数带有默认形式参数值；增加了引用；提供了更方便的动态内存空间管理方法；提供了异常的检查、处理机制，提高了程序的健壮性等。

C++语言具有 C 语言无法比拟的优越性，因为它同时支持面向对象的程序设计，支持封装性、

继承性和多态性,使得程序更安全、代码可重用性更高、可维护性更强,因而成为目前应用最为广泛的高级程序设计语言。

2. 简述面向对象方法所具有的三个特征。

【参考答案】面向对象的程序设计方法所具有的三个特征是封装性、继承性和多态性。

封装性是面向对象程序设计的第一大特征。封装指将数据和处理这些数据的过程结合成一个有机的整体——类,通过定义类实现封装,而封装的实际单位是属于类的对象。封装机制使对象将非 public 成员以及接口函数实现的内部细节隐藏起来,并能管理自己的内部状态。外部只能从对象所表示的具体概念、对象提供的服务和对象提供的外部接口来认识对象,通过向对象发送消息来激活对象的自身动作,实现一定的功能。封装性使得面向对象程序设计具有高安全性和可靠性。

继承性是面向对象程序设计的第二大特征,是面向对象的程序设计提高代码重用性的重要措施。继承表现了特殊类与一般类之间的上下分层关系,这种机制为程序员提供了一种组织、构造和重用类的手段。继承使一个类(称为基类或父类)的数据成员和成员函数能被另一个类(称为派生类或子类)重用。在派生类中只需增加一些基类中没有的数据成员和成员函数,或是对基类的某些成员进行改造,这样可以避免公共代码的重复开发,减少代码和数据冗余。

多态性是面向对象程序设计的第三大重要特征。面向对象程序设计的多态性指一种行为对应着多种不同的实现。多态性有静态多态性(也称为编译时的多态性)和动态多态性(也称运行时的多态性)两种。静态多态性通过函数重载和运算符重载来实现,动态多态性需要继承、虚函数、基类的指针或引用来实现。多态性使得同一个接口可以实现不同操作,大大方便了使用者。

3. 如何理解面向对象的程序设计体现为类的设计和类的使用这两大过程?

【参考答案】面向对象的程序设计,通过定义类的对象,再向对象发送消息使其产生相应的响应来实现程序的功能,这体现的是对类的使用。但由于对象属于类,对象可以有哪些数据成员来表达其静态特性,可以执行哪些成员函数以实现其动态特性,实际上都取决于类的设计,因此在面向对象的程序设计中,最富挑战性和创造性的工作首先是类的设计。类的设计还体现在通过继承与派生机制定义新的类,进而可以定义新类的对象完成程序的功能。因此,面向对象的程序设计体现为类的设计和类的使用两大过程。

4. 如何在 VC++ 6.0 中获取帮助信息?

【参考答案】程序员可以通过按 F1 得到上下文帮助。例如,在编辑文件时按 F1 可以得到有关编辑的帮助,在编译连接错误信息上按 F1 可以得到关于该错误的帮助信息。还可以通过选择菜单 Help|Contents 来启动 MSDN 查阅器,MSDN 查阅器是一个功能强大的程序,可以方便地浏览、查找信息。

5. 简述在 VC++ 6.0 中开发 C++程序的全过程。

【参考答案】开发 C++程序包括编辑、编译、链接和运行几个步骤。在 VC++ 6.0 中开发 C++程序主要也由上述过程组成。

由于 VC++ 6.0 以项目管理程序,所以对于多文件结构的程序,首先要建立一个项目,而单文件结构的程序可以省略此步,在编译时自动建立项目文件。

(1) 编辑源文件。

利用 VC++ 6.0 的编辑器可以进行源文件的编辑工作,生成扩展名为.cpp 的源程序文件。

(2) 编译链接源文件。

对编辑完成的文件需要进行编译和链接,在 VC++ 6.0 下这两步可以合在一起完成,方法是选择主菜单 Build 下的 Build 二级菜单,或选择工具栏上对应的工具按钮,如果编译链接正确,则

在下面的输出窗口中显示“0 errors 0 warnings”否则，将显示错误或告警信息，需要回到编辑状态重新修改，直到编译链接正确。

(3) 运行可执行文件。

在生成了.exe 可执行文件步骤完成后，就可以运行程序了。

运行程序的方法很多，在集成环境下，可以选择主菜单 Build 下的 Execute 二级菜单，或单击 Build 工具栏中的“!”按钮，则运行结果输出显示在 DOS 窗口的屏幕上。如果运行结果不正确，则可以启用调试器，找出问题所在，再重新修改源程序，编译链接后再运行。

第 2 章 C++ 语言对 C 语言的改进及扩展

一、单选题

1. 下列语句中错误的是_____。

- A. `int n=5; int y[n];`
- B. `const int n=5; int y[n];`
- C. `int n=5; int *py=new int[n];`
- D. `const int n=5; int *py=new int[n];`

【参考答案】A

【解析】定义一维数组时，一定要用常量指明数组变量所含分量个数，A 答案中的 n 是一个已赋值的变量而不是常量，故不能用来表示 y 数组所含分量个数。答案 B 中的 n 是一个用 const 定义的符号常量，故正确。在利用指针申请动态一维数组空间时，元素个数既可以是常量也可以是变量，因此答案 C 和 D 都是正确的语句。

2. 以下设置默认值的函数原型声明中错误的是_____。

- A. `int add(int X=3,int y=4,int Z=5);`
- B. `int add(int x,int y=4,int z);`
- C. `int add(int x,int y=4,int z=5);`
- D. `int add(int x,int y,int z=5);`

【参考答案】B

【解析】函数的形式参数提供默认值必须按从右到左依次的要求，显然答案 B 是错误的。答案 A 中所有的形式参数都有默认值，正确；答案 C 中右边两个形式参数都有默认值，也是正确的，答案 D 中只有最右边一个形式参数有默认值，左边两个形式参数都没有默认值，符合从右到左依次提供默认值的要求，也是正确的。

3. 下列不正确的重载函数是_____。

- A. `int print(int X);` 和 `void print(float X);`
- B. `int disp(int X);` 和 `char *disp(int Y);`
- C. `int show(int X, char * s);` 和 `int show(char * s, int X);`
- D. `int view(int X, int y);` 和 `int view(int X);`

【参考答案】B

【解析】根据函数重载的要求，同名函数在形式参数的个数、类型、顺序的一个或多个方面有

区别才是正确的重载，函数返回值类型不作为重载的条件。此题中，答案 A 在形式参数的类型上不同；答案 C 在形式参数的顺序上不同；答案 D 在形式参数的个数上不同，它们都符合函数重载的条件。答案 B 仅在返回值类型上不同，因此不是正确重载。

4. 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);`

【参考答案】D

【解析】根据利用 new 申请动态空间的用法，用 new 可以申请一个单位的动态空间，也可以申请一组连续的动态空间实现动态一维数组。答案 A 表示在申请一个动态空间的同时向这一空间中赋入值 10，此句相当于 `int *p=new int; *p=10;` 语句；答案 B 表示申请连续 10 个 int 型动态空间构成含有 10 个分量的动态一维数组；答案 C 表示申请了一个 int 型动态空间，但未向该动态空间中写入确定内容。答案 D 的错误在于试图在申请动态一维数组的同时为每一个元素赋以相同的初值，这种用法 C++ 不支持，因此是错误的。

5. 假设已有定义 `const char *name="chen";`，下面语句中错误的是_____。

A. `name[3]='a';`

B. `name="lin";`

C. `name=new char[5];`

D. `cout<<name[3];`

【参考答案】A

【解析】根据 const 与指针的结合内容，以上定义的名指针为指向常量的指针，此时 name 所指向的内容是常量，但是 name 指针本身是变量，因此可以被重新赋值，也可以利用它重新申请空间，故答案 B 和 C 都是正确的语句；答案 D 是输出内容，不涉及指针或其指向内容的改变，因此肯定正确；答案 A 试图修改指针所指向的内容，故错误。

二、填空题

1. 用 C++ 风格的输入/输出流进行输入/输出处理时，必须包含的头文件是_____。

【参考答案】`iostream.h` 或者名空间 `std` 中的文件 `iostream`

【解析】使用 C++ 的流控制方法进行输入/输出操作时，需要重点掌握以下要点“必须用 `#include <iostream>` 及 `using namespace std;` 或用 `#include <iostream.h>` 作文件包含”。

2. 在 C++ 语言的异常处理机制中，_____ 语句用于抛出异常；_____ 语句块用于检测异常；_____ 语句块用于捕捉和处理异常。

【参考答案】`throw`、`try`、`catch`

3. C++ 语言提供了专门用于表达逻辑型的数据类型_____，这种类型有两个常量分别用关键字是_____ 和 _____ 表示，分别代表_____ 和 _____。

【参考答案】`bool`、`true`、`false`、逻辑真、逻辑假

4. 在动态内存空间管理方面，C++ 语言用运算符_____ 取代了 C 语言中的 `malloc` 进行动态内存空间的申请，用运算符_____ 取代了 C 语言中的 `free` 释放动态内存空间。

【参考答案】`new`、`delete`

5. 为提高安全性，C++ 语言可以用_____ 来代替 C 语言中用宏定义符号常量，可以用_____ 来代替 C 语言中不安全的带参数的宏。

【参考答案】`const` 定义符号常量、内联函数

三、问答题

1. C++中的同名全局变量通过何种方式可以在同名局部量所在的函数内进行访问?

【参考答案】C++语言中,在同名局部量所在的函数内,通过在同名变量前加上域解析符“::”对被隐藏的同名全局变量进行访问。

2. 如果在程序中,我们只用 new 进行动态内存空间分配,而忘记用 delete 来释放,会产生什么样的后果?使用 new 和 delete 动态申请和释放内存空间,有什么好处?

【参考答案】C++语言中,在用 new 或 new[] 申请了动态内存空间后,一旦这部分动态内存空间使用结束,一定要及时使用 delete 或 delete[] 释放空间,如果忘记用 delete 释放,则会产生内存垃圾,影响程序的正常运行,还有可能导致死机。

使用 new 和 delete 动态申请和释放内存空间,其好处首先表现在使用形式比 malloc、calloc、free 要简洁。更重要的是,使用 new 和 delete 可以正确地使用 C++语言中一些重要技术,例如,面向对象程序设计中的构造函数与析构函数等。

3. 函数被说明为内联函数后,对哪些开销有影响?

【参考答案】函数被说明为内联函数后,对空间和时间开销均有影响。空间方面:调用内联函数时,编译器对其进行扩展,即用内联函数体的代码替换掉内联函数调用语句,故代码区所需空间会更大一些;但是调用内联函数避免了调用一般函数时将程序当前的一些状态信息存到栈中所需要的空间开销。时间方面,节省了时间开销。因为调用内联函数时不需要转到函数的代码处去执行函数体语句,再返回到调用点等,这些传递过程所需要的时间开销节省了。总的来说,内联函数是一种用空间换时间的措施,故内联函数代码不能过长,否则将失去其优势。

四、读程序写结果

1. 当从键盘上输入 23.56 10 90<回车>时,写出下面程序的运行结果。

```
#include "iostream.h"
void main( )
{
    int a,b,c;
    char ch;
    cin>>a>>ch>>b>>c;
    cout<<a<<endl<<ch<<endl<<b<<endl<<c;
}
```

【参考答案】23

56

10

【解析】根据 C++输入流读入数据的方法,当读入 23.56 时,因为第 1 个接受变量 a 定义为 int 型,因此当遇到.时自动认为第 1 个整数 23 输入结束,接下来的.自然读入到字符型变量 ch 中,接下来的 56 读入到变量 b 中,而 10 读入到变量 c 中,90 是多余的输入被忽略。

2. 写出下面程序的运行结果。

```
#include <iostream>
using namespace std;
void main()
{
    int arr[4]={1,2,3,4};
    int *a=arr;
```

```

int *p=a; //p 是一个指针引用
p++;
*p=100;
cout<<*a<<" "<<*p<<endl;
for (int i=0;i<4;i++)
    cout<<arr[i]<<" ";
cout<<endl;
int b=10;
p=&b;
cout<<*a<<" "<<*p<<endl;
for ( i=0;i<4;i++)
    cout<<arr[i]<<" ";
cout<<endl;
}

```

【参考答案】100 100

```

1 100 3 4
10 10
1 100 3 4

```

【解析】此题考查利用指针间接访问一维数组元素、普通变量的值，以及引用是其所代表的变量别名的含义。此题中，a 一开始是一个指向数组空间的指针，p 是一个引用，是 a 的别名，因此，p++ 实际上就是 a++，对 *p 的赋值就是对 *a 的，也就是 arr[1] 元素的赋值，因此，*p、*a 和 arr[1] 的输出值相等。p=&b 实际上就是 a=&b，因此，*p 和 *a 的输出值都是 b 的值。

3. 写出下面程序的运行结果。

```

#include <iostream>
using namespace std;
int i=0;
void main()
{ int i=5;
  { int i;
    i=7;
    cout<<"::i="<<::i<<endl;
    cout<<"i="<<i<<endl;
    ::i=1;
    cout<<"::i="<<::i<<endl;
  }
  cout<<"i="<<i<<endl;
  cout<<"::i="<<::i<<endl;
  i+=::i;
  ::i=100;
  cout<<"i="<<i<<endl;
  cout<<"::i="<<::i<<endl;
}

```

【参考答案】::i=0

```

i=7
::i=1
i=5
::i=1
i=6
::i=100

```

【解析】此题考查同名变量的作用域问题。题中 3 个 *i* 分别是全局变量 *i*、*main* 函数开始处定义的局部变量 *i*、*main* 函数的复合语句开始位置定义的局部变量 *i*，它只在该复合语句内有效，在 *main* 函数的其余位置直接用 *i* 操作的是第 2 个 *i*，而在 *main* 函数中对全局变量 *i* 的操作形式都是 `::i`，搞清楚了每一处的 *i* 是哪一个 *i*，该题的运行结果就非常清楚了。

4. 写出下面程序的运行结果。

```
#include<iostream>
using namespace std;
void f(double x=50.6, int y=10, char z='A');
void main( )
{   double a=216.34;
    int b=2;
    char c='E';
    f( );
    f(a);
    f(a,b);
    f(a,b,c);
}
void f(double x, int y, char z)
{   cout<<"x="<<x<<" y="<<y<<" z="<<z<<endl;
}
```

【参考答案】`x=50.6 y=10 z=A`

`x=216.34 y=10 z=A`

`x=216.34 y=2 z=A`

`x=216.34 y=2 z=E`

【解析】此题考查对带有默认参数值的函数的调用方式。函数 *f* 的 3 个形式参数均有默认参数值，因此对 *f* 的调用可以有 4 种形式，即不提供实际参数、只提供第 1 个实际参数、提供前两个形式参数、3 个实际参数都提供。因此第 1 行的输出结果表明在未提供实际参数时，3 个形式参数均使用默认参数值；下面 3 行的结果表明实际参数与形式参数对应的方式是按从左到右依次，如果未获得对应的实际参数，则形式参数使用默认参数值。

5. 写出下面程序的运行结果。

```
#include <iostream.h>
int & s(const int &a,int &b)
{   b+=a;
    return b;
}
void main( )
{   int x=500,y=1000,z=0;;
    cout<<x<<' \t'<<y<<' \t'<<z<<' \n';
    s(x,y);
    cout<<x<<' \t'<<y<<' \t'<<z<<' \n';
    z=s(x,y);
    cout<<x<<' \t'<<y<<' \t'<<z<<' \n';
    s(x,y)=200;
    cout<<x<<' \t'<<y<<' \t'<<z<<' \n';
}
```

【参考答案】`500 1000 0`

`500 1500 0`

`500 2000 2000`

500 200 2000

【解析】此题考查引用作为形式参数和返回值的用法。函数 s 的返回值类型为引用，因此其调用形式有 3 种，即直接作为一个函数调用语句、将调用结果作为右值、将调用结果作为左值（这是引用返回特有的用法）。另外， s 函数的第 1 个形式参数是常引用，不可被改变；第 2 个形式参数是引用参数，在函数中被改变了，由于引用参数是实际参数在函数中的别名，故与第 2 个形式参数对应的实际参数变量将随着每一次调用得到改变。从输出结果上可以看出，第 1 个实际参数 x 在几次调用结束后都保持原值 500，第 2 个实际参数 y 在每次调用后都增加 500，但最后一次调用是将函数返回值作为左值赋值为 200，因此 y 值为 200， z 的结果容易理解。

五、编程题

1. 将下面的 C 语言风格的程序改写成 C++ 语言风格的源程序。

```
#include <stdio.h>
add(int a,int b);
void main( )
{
    int x,y,sum;
    printf("Please input x and y:\n");
    scanf("%d%d",&x,&y);           /*输入变量 x 和 y 的值*/
    sum=add(x,y);                 /*调用求和函数，结果存于 sum 中*/
    printf("%d+%d=%d\n",x,y,sum) ; /*显示计算结果*/
}
add(int a,int b)                 /*求和函数*/
{    return a+b; }
```

【参考答案】/*biancheng2_1.cpp 程序改写题*/

```
#include <iostream>             //包含 std 名字空间中的 iostream 文件
using namespace std;
int add(int a,int b);          //函数应该有明确的返回值类型
void main( )
{
    int x,y,sum;
    cout<<"Please input x and y:\n"; //用输出流输出提示信息
    cin>>x>>y ;                 //用输入流输入变量 x 和 y 的值
    sum=add(x,y);              //调用求和函数，结果存于 sum 中
    cout<<x<<"+"<<y<<"="<<sum<<endl; //用输出流显示计算结果
}
int add(int a, int b)          //定义求和函数
{    return a+b; }
```

2. 用 C++ 语言的编程风格编程实现一个输入半径，输出其面积和周长的程序。

【参考答案】/*biancheng2_2.cpp :输入半径求圆的周长和面积*/

```
#include <iostream>
using namespace std;
const double PI=3.1415926;
void main()
{
    double radius,circumference,area;
    cout<<"Please input radius"<<endl;
    cin>>radius;
```



```

    circumference=2*PI*radius;
    area=PI*radius*radius;
    cout<<"circumference="<<circumference<<endl;
    cout<<"area="<<area<<endl;
}

```

3. 用 `new` 运算符为一个包含 20 个整数的数组分配内存, 输入若干个值到数组中, 分别统计其中正数和负数的个数, 输出结果, 再用 `delete` 运算符释放动态内存空间。

【参考答案】/*biancheng2_3.cpp: 动态数组空间的管理和操作*/

```

#include <iostream>
using namespace std;
void main()
{
    int num, positive=0, negative=0;
    int *p=new int[20];
    cout<<"Please input the number you want use of array:\n";
    cin>>num;
    if (num>20)
    {
        cout<<"number too large,exit.";
        return;
    }
    for (int i=0;i<num;i++)
        cin>>p[i];
    for (i=0;i<num;i++)
        if (p[i]>0) positive++;
        else if (p[i]<0) negative++;
    cout<<"There are "<<num<<" figures,\n";
    cout<<positive<<" of them are positive numbers,\n";
    cout<<negative<<" of them are negatives.\n";
    delete []p;
}

```

4. 编写程序, 从键盘上输入一个学生的姓名 (建议用 `string` 类型)、年龄 (合理的年龄在 16~25), 五级制 C++ 语言考试分数 (合理范围: 0~5), 调用函数 `float checkAgeScore(int age, float score)`, 该函数主要完成两件事: 通过检查两个形式参数的范围是否合理, 抛出不同的异常信息; 如果无异常, 则返回对应的百分制成绩。主函数中定义 `try-catch` 块检测、捕获并处理异常。最后输出的是该同学的姓名、年龄、百分制成绩。

【参考答案】/*biancheng2_4.cpp: 异常处理*/

```

#include <iostream>
#include <string>
using namespace std;
float checkAgeScore(int age, float score)
{
    if (age<16||age>25) throw age;
    if (score<0||score>5) throw score;
    return score*20;
}
void main()
{
    string name;
    int age;
    float score5, score100;
    cout<<"Please input name, age and score of 5 grade:\n";
    cin>>name>>age>>score5;
    try

```

```

    {   score100=checkAgeScore(age,score5);
        cout<<name<<" "<<age<<" "<<score100<<endl;
    }
catch (int)
{   cout<<"Out of natural age\n";
}
catch (float)
{   cout<<"Out of natural score\n";
}
}

```

第3章 类与对象

一、单选题

1. 下列类定义方式正确的是_____。

- | | | | |
|--|---|---|--|
| A. struct st{
char s[20];
int top;
}; | B. class st{
char s[20];
int top;
}; | C. class st{
char s[20];
int top;
};A; | D. class st{
char s[20];
int top;
};A |
|--|---|---|--|

【参考答案】C

【解析】选项 A、B 在结构体和类定义结束时应以“;”结束，选项 D 在定义变量时错误。

2. 下列正确的类定义方式是_____。

- | | | | |
|--|--|---|--|
| A. class Retangle
{private:
int X=15,Y=20;
}; | B. class Location
{ int X;
int Y,Z;
public:
void dod(int=0,int=0,int=0);
}; | C. class Sample
{ int X, Y;
public:
Sample(int m,int n)
{m=X; n=Y;}
}; | D. class example
{ int figure;
char name;
};
int i=figure; |
|--|--|---|--|

【参考答案】B

【解析】A 项中仅有私有数据，而无对外的接口，无法完成对私有数据的操作；D 项中对私有数据 figure 的引用是非法的；C 项中构造函数对私有数据进行赋值的方向错误。

3. 如果 class 类中的所有成员在定义时都没有使用关键字 public、private、protected，则所有成员缺省的访问属性为_____。

- | | | | |
|-----------|------------|-----------|--------------|
| A. public | B. private | C. static | D. protected |
|-----------|------------|-----------|--------------|

【参考答案】B

【解析】类定义中缺省访问属性为 private。

4. 下列有关类和对象的说法不正确的是_____。
- A. 对象是类的一个实例 B. 任何一个对象只能属于一个具体的类
- C. 一个类只能有一个对象 D. 类与对象的关系类似于数据类型与变量的关系

【参考答案】C

【解析】类是相同属性的抽象，可以定义若干个实体对象。

5. 对于任意一个类，析构函数的个数为_____。
- A. 0 B. 1 C. 2 D. 3

【参考答案】B

6. 通常类的拷贝构造函数的参数是_____。
- A. 某个对象名 B. 某个对象的成员名
- C. 某个对象的引用名 D. 某个对象的指针名

【参考答案】C

7. 下面是关于一个类的友元的说法，其中不正确的是_____。
- A. 友元函数可以访问该类的私有数据成员
- B. 友元的声明必须放在类的内部
- C. 友元函数可以是另一个类的成员函数
- D. 若 X 类是 Y 类的友元，Y 类就是 X 类的友元

【参考答案】D

【解析】友元关系没有交互性，只有在 X, Y 双方都声明对方为自己的友元时才可以互认友元。

8. 下面是关于静态成员的说法，其中不正确的是_____。
- A. 静态成员有类作用域，但与普通非静态成员有所不同
- B. 静态函数没有 this 指针，同一个类的不同对象拥有相同的静态变量
- C. 静态数据成员的初始化必须在类外进行
- D. 静态成员函数可以直接访问非静态数据成员

【参考答案】D

【解析】静态成员函数一般不访问非静态数据成员，如果需要，只能通过对象名来访问。

二、问答题

1. 类声明的一般格式是什么？

【参考答案】

```
class 类名
{
    [private:]
        私有数据成员和成员函数
    public:
        公有数据成员和成员函数
};
```

2. 构造函数和析构函数的主要作用是什么？它们各有什么特性？

【参考答案】构造函数可以创建对象并初始化对象，其主要特性如下：

(1) 构造函数的函数名必须与类名相同，以类名为函数名的函数一定是类的构造函数。