普通高等教育“十二五”规划教材

C++

程序设计习题与实验教程

主　编　祁云嵩　王　芳
副主编　张晓如　华　伟　於跃成

科学出版社

普通高等教育“十二五”规划教材

C++程序设计习题与实验教程

主　编　祁云嵩　王　芳
副主编　张晓如　华　伟　於跃成

科学出版社
北京

内 容 简 介

本书是C++程序设计课程的学习与上机实验教材。在内容和章节安排上与《C++程序设计教程》（科学出版社 2013 年出版）配套。全书共 11 章，内容包括概述、数据类型与表达式、流程控制语句、数组、函数与编译预处理、结构体与简单链表、类和对象、继承与多态性、友元函数与运算符重载、模板与异常处理及输入/输出流。每章都是由知识点概要、典型例题解析、习题及实验内容与指导四部分组成。

本书可作为普通高等学校C++课程的上机实践教材，也可供编程爱好者阅读参考。

图书在版编目(CIP)数据

C++程序设计习题与实验教程 / 祁云嵩，王芳主编. —北京：科学出版社，2013

普通高等教育“十二五”规划教材

ISBN 978-7-03-038386-0

I. ①C… II. ①祁… ②王… III. ①C 语言–程序设计–高等学校–教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 190342 号

责任编辑：相 凌 王迎春 / 责任校对：鲁 素
责任印制：阎 磊 / 封面设计：华路天然工作室

科学出版社出版
北京东黄城根北街 16 号
邮政编码: 100717
http://www.sciencep.com

北京通州皇家印刷厂 印刷
科学出版社发行 各地新华书店经销
*
2013 年 8 月第 一 版 开本：787×1092 1/16
2013 年 8 月第一次印刷 印张：14 3/4
字数：349 000

定价：33.00 元

(如有印装质量问题，我社负责调换)

前　言

C++程序设计是一门实践性很强的课程，仅通过课堂教学和阅读教科书，很难提高程序设计能力。在日常的教学实践中，很多学习者的感觉是课堂内容易懂、难记，实际编程时无从下手，其主要原因还是学习方法不当。本书的主要思想是使学生在应用中学习知识，在练习中巩固知识。

本书共11章，每章内容分为四部分，第一部分为知识点概要，第二部分为典型例题解析，第三部分为习题，第四部分为实验内容与指导。由于编程实践不但可以培养、训练学生对程序设计语言的应用能力，更重要的是使学生在应用中掌握知识。学生要解决应用中的问题，就必须掌握相关知识，这种内在的需求比常规的被动学习效果要好得多。所以，各章节的第四部分的实验是整个C++程序设计语言学习的主线，其内容的安排力求做到由浅入深、由易到难。在每章的实验中，都安排了一个或几个实验程序，对于一些有难度的实验内容，书中还给出了类似的可供参考的例题，以帮助学习者加深理解。建议在完成第四部分实验的基础上再通过第三部分的习题巩固基础知识，细化知识点。

本书由祁云嵩老师负责编写第1章和第7章并统稿，张晓如老师负责编写第2章和第3章，华伟老师负责编写第4章和第8章，於跃成老师负责编写第5章，王芳老师负责编写第6章和第9章，王勇老师负责编写第10章，束鑫老师负责编写第11章。

本书在编写过程中得到了教研室全体教师的帮助，学校教材科全体教师也给予了大力支持，在此一并表示感谢。

由于编者水平有限，书中疏漏之处恳请广大读者批评指正。

作　者

2013年6月

目　录

前言
第 1 章　概述 ····· 1
1.1　知识点概要 ····· 1
1.2　典型例题解析 ····· 3
1.3　习题 ····· 5
1.4　实验内容与指导 ····· 6
第 2 章　数据类型与表达式 ····· 7
2.1　知识点概要 ····· 7
2.2　典型例题解析 ····· 11
2.3　习题 ····· 13
2.4　实验内容与指导 ····· 15
第 3 章　流程控制语句 ····· 17
3.1　知识点概要 ····· 17
3.2　典型例题解析 ····· 19
3.3　习题 ····· 22
3.4　实验内容与指导 ····· 36
第 4 章　数组 ····· 38
4.1　知识点概要 ····· 38
4.2　典型例题解析 ····· 40
4.3　习题 ····· 49
4.4　实验内容与指导 ····· 64
第 5 章　函数与编译预处理 ····· 66
5.1　知识点概要 ····· 66
5.2　典型例题解析 ····· 70
5.3　习题 ····· 75
5.4　实验内容与指导 ····· 88
第 6 章　结构体与简单链表 ····· 89
6.1　知识点概要 ····· 89
6.2　典型例题解析 ····· 94
6.3　习题 ····· 99
6.4　实验内容与指导 ····· 115

第 7 章 类和对象 …… 116
7.1 知识点概要 …… 116
7.2 典型例题解析 …… 120
7.3 习题 …… 133
7.4 实验内容与指导 …… 145
第 8 章 继承与多态性 …… 147
8.1 知识点概要 …… 147
8.2 典型例题解析 …… 149
8.3 习题 …… 159
8.4 实验内容与指导 …… 177
第 9 章 友元函数与运算符重载 …… 180
9.1 知识点概要 …… 180
9.2 典型例题解析 …… 186
9.3 习题 …… 191
9.4 实验内容与指导 …… 203
第 10 章 模板与异常处理 …… 205
10.1 知识点概要 …… 205
10.2 典型例题解析 …… 207
10.3 习题 …… 210
10.4 实验内容与指导 …… 214
第 11 章 输入/输出流 …… 215
11.1 知识点概要 …… 215
11.2 典型例题解析 …… 219
11.3 习题 …… 221
11.4 实验内容与指导 …… 228

第1章 概　　述

1.1 知识点概要

1. C++源程序格式

以下为一个简单的C++程序示例：

```
//C++程序设计示例
#include<iostream.h>
void main(void)
{
    cout<<"C++程序设计\n";    /* 简单的屏幕输出 */
}
```

运行该程序，屏幕显示如下：

```
C++程序设计
```

一个简单的C++程序由程序注释、文件包含指令、主函数等部分组成。

1)程序注释

注释仅用来向读者解释程序的内容，系统并不执行注释的内容。“//”后当前行内所有的字符均为注释信息。“/*…*/”用于标识多行注释的开始和结束。程序的注释信息有时很重要，它能帮助读者或程序员理解程序(程序员可能也会忘记设计程序时的某些细节问题)。

2)文件包含指令

以“#”开头的行称为编译预处理指令。上述示例程序中的第二行编译预处理指令的功能是将头文件 iostream.h 包含进来。C++程序如果要进行输入/输出，必须包含文件iostream.h，该系统文件中定义了输入/输出方法。

3)主函数

函数是 C++的程序的基本组成部分。任一 C++程序均由一个或多个函数组成，并且有且只有一个主函数，上述示例程序中的“void main(void)”是主函数的头部。C++程序从主函数的第一条语句开始执行，直至主函数结束。在主函数中可以调用其他函数。C++程序中的每个函数体都必须以“{”开始，以“}”结束。在本例的主函数中，语句“cout<<"C++程序设计\n";”的功能是将双引号中的内容在屏幕上显示出来(“\n”表示换行符，参见第 2 章)。注意，一个完整的 C++功能语句必须以分号结束，语句中的双引号、分号均为西文符号。

2. C++程序上机过程

本书以 Microsoft Visual C++6.0 为编译环境说明 C++程序的上机过程，其步骤如下。

(1)在操作系统环境下启动 VC++集成开发环境，打开图 1-1 所示的界面。

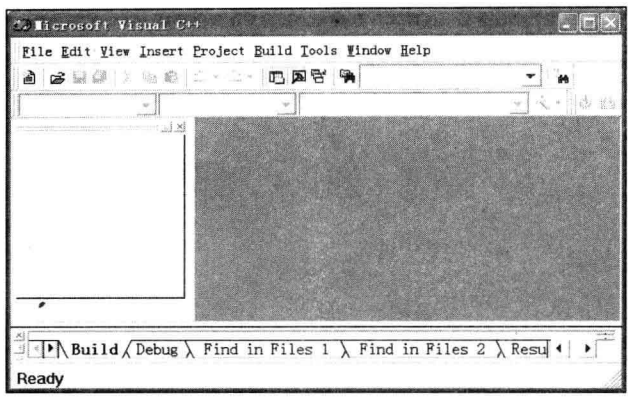

图 1-1 VC++集成开发环境主界面

(2)选择 File(文件)菜单下的 New(新建)命令，出现图 1-2 所示的界面(不可通过“新建”按钮建立新的 C++源程序文件，该按钮的功能是新建一个文本文件)。图 1-2 所示界面中的标签 Projects 可为新程序设定工程项目。对初学者来说，编辑小的源程序不必建立项目，可以直接选择左上角的 Files 标签，打开图 1-3 所示的界面。

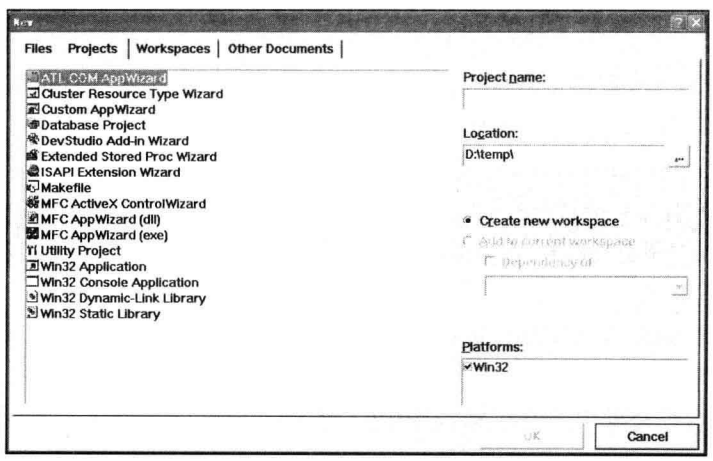

图 1-2 新建 VC++工程项目界面

(3)在图 1-3 所示的界面左侧选定文件类型为 C++Source File，在右侧输入程序文件名并选定文件存放目录，然后单击 OK 按钮，出现如图 1-4 所示的程序编辑界面，即可开始编辑程序。

(4)编辑完源程序后，执行 Build 菜单下的 Compile 命令，对源程序进行编译。系统将在下方的窗口中显示编译信息。如果界面中无此窗口，可按 Alt+2 组合键或执行 View→Output 命令。

如果源程序有语法错误，系统将显示出错误所在的行号并给出提示信息。双击相应的错误提示内容，源程序中的光标将自动移至该错误所在的行，但这仅表示该错误可能由这

一行引起，具体错误内容可根据系统提示信息进行判断，一般从第一个错误开始修改。

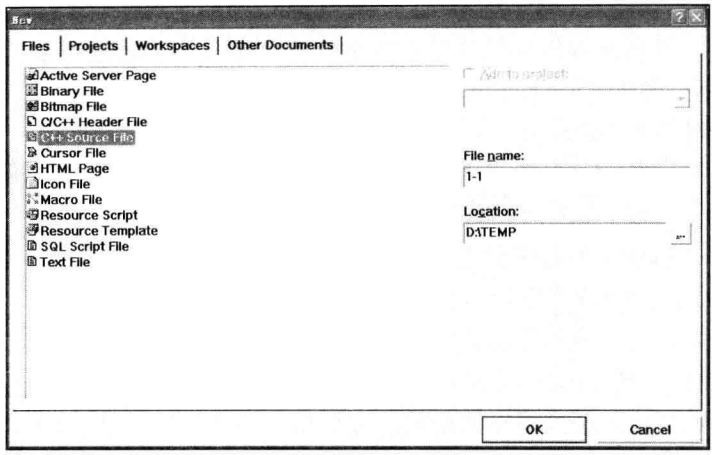

图 1-3　新建 VC++源程序文件界面

图 1-4　VC++源程序编辑界面

如果编译后已无错误提示，则可执行 Build→Build 命令生成相应的可执行文件，随后执行 Build→Execute 命令运行程序。

(5)如果要编辑第二个源程序，则应通过 File 菜单下的 Close Workspace 命令关闭当前工作区(注意：File 菜单下的 Close 命令只能关闭正在编辑的文件，而不能关闭当前工作区)，然后重复以上步骤。否则，即使关闭了原来的文件，新编辑的源程序还将与原来的程序相互影响而难以运行。

1.2　典型例题解析

【例 1.1】C++源程序默认的扩展名为_____。

A. cpp　　B. txt　　C. exe　　D. obj

【答案】A

【解析】系统默认的C++源程序的扩展名为cpp，源程序编译后生成相应的目标文件，其扩展名为obj，目标文件经系统连接后生成的可执行文件扩展名为exe。

【例1.2】下列关于C++程序的书写规则，不正确的是_____。

A. 一行可以写若干条语句　　B. 一条语句可以写成若干行

C. 可以在程序中插入注释信息　　D. C++程序不区分大小写字母

【答案】D

【解析】C++程序是严格区分大小写字母的，例如，主函数名main不能写成Main。

【例1.3】关于C++程序的执行过程，以下说法正确的是_____。

A. 从主函数开始，直到主函数结束

B. 从程序的第一行开始，直到程序的最后一行结束

C. 从主函数开始，直到程序的最后一行结束

D. 从程序的第一个函数开始，直到程序的最后一个函数结束

【答案】A

【解析】不管主函数处于程序的什么位置，C++程序总是从主函数的第一条语句开始执行，主函数执行结束，则整个程序也结束运行。其他所有函数的运行均直接或间接由主函数调用。

【例1.4】在C++程序中，要使用库函数，必须用编译预处理指令将相应的头文件包含进来。如要使用标准的数学函数，相应的编译预处理指令为_____。

【答案】#include<math.h>

【解析】C++的数学头文件math.h中包含了常用的数学函数。

【例1.5】完整的C++程序中，有且只有一个_____函数。

【答案】main

【解析】任何一个完整的可执行C++程序必定有且只有一个主函数。

【例1.6】编程实现在屏幕上显示几行文字信息。

【程序】

```
#include<iostream.h>
void main(void)
{   cout<<"我设计的第一个C++程序！"<<'\n';
    cout<<"从中我了解了C++程序的基本组成。"<<endl;
}
```

【解析】该程序经编译、连接后运行时在屏幕上显示以下两行文字：

```
我设计的第一个C++程序！
从中我了解了C++程序的基本组成。
```

程序的第一行表示程序编译时要将系统提供的文件iostream.h包含进来，因为在该文件中定义了标准的输入/输出方法。在C++程序中如果要用到标准设备(键盘、显示器)

的输入/输出，均应通过该指令将 iostream.h 文件包含进来。

任何一个可执行的 C++程序均应包含一个且只能包含一个主函数，C++程序总是从主函数开始执行。程序的第二行是主函数的头部，下面用一对大括号括起来的部分是主函数的内容，只要修改这部分内容就可得到不同的运行结果。

在设计 C++程序时还应注意以下几点。

(1) 主函数的内容必须包含在一对大括号内。

(2) 主函数中各语句必须以分号结束。

(3) 双引号中的内容是要在屏幕上显示的内容。程序中的 endl、\n 均表示换行信息。如果不输出换行信息，则第二行输出将接在第一行后面。

(4) 编程时应注意分号和引号均应为西文字符。C++程序中除了程序注释信息以外，只有双引号内才可以出现中文信息。

1.3 习 题

一、选择题

1. 关于 C++程序设计语言，下列叙述正确的是_____。

A. 花括号“{”和“}”只能作为函数体的定界符

B. 界于“/*”和“*/”之间的注释部分可以多于一行

C. C++程序的每一行都应以分号结束

D. C++程序中只能有一行库文件包含命令

2. 下列说法不正确的是_____。

A. C++程序总是从主函数开始执行

B. C++的主函数必须出现在所有函数之前

C. C++的主函数必须以 main 命名

D. C++中除了主函数以外，还可以有其他函数

3. 下列可用于标识 C++源程序注释的符号为_____。

A. #　　B. //　　C. %　　D. ;

4. 一个完整的 C++源程序中，_____。

A. 必须有一个主函数　　B. 可以有多个主函数

C. 必须有主函数和其他函数　　D. 可以没有主函数

二、填空题

1. 一个 C++程序必须有且只能有一个_____函数。

2. 在 C++程序中，要使用库函数，必须用编译预处理指令将相应的头文件包含进来；如要进行标准输入/输出，则该编译预处理指令为_____。

3. C++源程序编辑好后，还必须经过____①____和____②____才能得到可执行文件。

4. C++源程序中，函数体应置于_____之内。

5. 一个完整的 C++功能语句应以_____结束。

6. C++源程序缺省扩展名为____①____，经编译后生成的目标文件扩展名为____②____，

连接后生成的可执行文件扩展名为____③____。

三、编程题

编写一个简单的C++程序，实现在屏幕上显示自己的姓名和学号。

1.4 实验内容与指导

【实验目的】

1. 熟练使用VC++的编程环境。
2. 初步了解C++程序的编译、连接和运行过程。
3. 掌握和理解C++程序的结构。
4. 熟悉C++程序数据的输入/输出。

【实验内容】

编程在屏幕上按如下格式显示唐诗：

春眠不觉晓，
　处处闻啼鸟；
　　夜来风雨声，
　　　花落知多少。

【实验指导】

总结实验中在编辑、编译、连接、运行等环节所出现的问题及解决方法。

第 2 章　数据类型与表达式

2.1　知识点概要

2.1.1　标识符

1. 字符集

C++的字符集由 ASCII 码中的可见字符构成。

2. 关键字

关键字是由 C++中预先约定用于固定用途的字符组合。

3. 标识符

C++中的关键字和用户自定义标识符均由 C++符号集中的符号组成。

用户自定义标识符只能由字母、数字和下划线 3 种字符组成，第 1 个字符必须为字母或下划线，不能是数字。

4. 分隔符

分隔符用来分隔各语法单位，常用的分隔符有空格、制表符、换行符、注释符、运算符和标点符号。另外，还用一些标点符号作为语法约束，如表 2-1 所示。

表 2-1　C++语言中常用标点符号及其作用

标点符号	作用	标点符号	作用
,	作为数据分隔或运算符	;	作为语句结束符
:	作为语句标号	{ }	作为复合语句的标记符或自定义类型的成员范围
'	作为字符常量标记符	()	作为运算符运算次序标记符，或用于函数参数及调用
"	作为字符串常量标记符	…	作为可变参数

2.1.2　数据类型与表达式

1. 基本数据类型

常量与变量都具有类型，整型数据可用关键字 short、long、unsigned 和 signed 来修饰，如长整型(long int)、短整型(short int)、无符号长整型(unsigned long int)等。不同数据分配不同大小的空间。

对于像 unsigned long int 这样长的数据类型名，C++语言提供了一个关键字 typedef，可用它定义一个简单的名字，用来简化程序，例如：

```
typedef unsigned long int ulint;
```

这样，在程序中所有出现 ulint 的地方均表示 unsigned long int。注意，这里并没有定义新的数据类型，只是给原类型名起了一个别名。

2. 字面常量

从字面形式即可识别的常量称为字面常量，如整型常量、实型常量、字符型常量、字符串常量等。

(1)整型常量。整型常量可用十进制、八进制和十六进制 3 种不同的方式表示。其中，八进制常量以数字 0 开头，十六进制常量以数字 0 和英文字母 X(或 x)开头。

(2)实型常量。实型常量可用十进制小数形式和指数形式两种不同方式表示。其中，指数形式表示为

数符+尾数+e(或 E)+阶码+阶数

其中，尾数和阶数均不可少。

(3)字符型常量。在内存中用 ASCII 码存储，与整型数据可以通用。

对于转义字符，若“\”后接数字，则只能是八进制或十六进制数，且取值范围为十进制数 0~255。

(4)字符串常量。在存储时编译系统会自动加一个结束标志“\0”，它不属于字符串的长度部分，但占一个字节。

3. 符号常量

用一个符号名代表一个常量，称为符号常量，即以标识符形式出现的常量。

(1)用宏定义。格式如下：

```
#define 宏名 常量值
```

其中，常量值可以是前面介绍的各种类型。

(2)用 const 定义。格式如下：

```
const 数据类型  常量名=常量值;
```

或

```
const 数据类型  常量名(常量值);
```

其中，数据类型可以是除空类型外的任何一种数据类型，“=”称为赋值号，用于完成赋值工作。

用宏定义表示常量时数据没有类型，在内存中并不存在以符号常量命名的存储单元。用宏定义常量为编译预处理，最后不可以用分号结束。

用 const 定义的量又称为常变量，它具有变量的特征、类型，在内存中存在着以它命名的存储单元，可以用 sizeof 运算符测出其长度。与一般变量唯一的不同是其值不能改变。用 const 定义常变量的形式最后以分号结束。

4. 变量

变量定义的一般格式如下：

```
数据类型  变量名 1，变量名 2，…，变量名 n;
```

变量赋初始值的一般格式如下：

```
数据类型　变量名=表达式;
```

或

```
数据类型　变量名(表达式);
```

指针变量定义的一般格式如下：

```
数据类型　*变量名 1，*变量名 2，…，*变量名 n;
```

引用变量定义的一般格式：

```
数据类型 &引用名=变量名;
```

(1)变量必须先定义后使用。变量定义后可存放相应的数据，即为变量赋值。

(2)指针变量中存放的是地址，可通过该地址取到相应内存中的数据。

(3)引用变量为某个已有变量起的别名，实际上与已有变量是同一个变量，故其值与原已有变量的值相同。

5. 数据的输入/输出

数据输入的一般格式如下：

```
cin>>变量名 1>>变量名 2>>…>>变量名 n;
```

数据输出的一般格式如下：

```
cout<<表达式 1<<表达式 2<<…<<表达式 n;
```

输入的数据可用空格分隔，也可用回车符分隔。输入/输出的整型数据在通常情况下默认为十进制数，也可输入/输出其他进制整数。例如，若需要输入/输出八进制整数，可用 oct 表示；若需要输入/输出十六进制整数，可用 hex 表示；十进制整数可用 dec 表示。

2.1.3　运算符与表达式

1. 运算符

运算符除了具有相应的含义外，还要考虑其优先级、结合性，以及操作数的类型等。各运算符的具体情况见表 2-2。

表 2-2　C++的运算符

优先级	运算符	含义	目数	结合性
1	::	作用域运算符	2	从左向右
	()	改变运算优先级		
	[]	下标运算符		
	->	指向结构体成员运算符		
	.	结构体成员运算符		
	&	引用运算符	1	从右向左
	++、--	后置自增、自减运算符		

续表

优先级	运算符	含义	目数	结合性
2	!	逻辑非运算符	1	从右向左
	++、--	前置自增、自减运算符		
	-	取负运算符		
	+	取正运算符		
	(类型)	类型转换运算符		
	*	指针运算符		
	&	地址运算符		
	sizeof	数据类型长度运算符		
	new	分配存储单元		
	delete	释放存储单元		
3	*	乘法运算符	2	从左向右
	/	除法运算符		
	%	求模运算符		
4	+	加法运算符	2	从左向右
	-	减法运算符		
5	<< 、>>	左移位、右移位运算符	2	从左向右
6	<、 <= 、> 、>=	小于、小于等于、大于、大于等于运算符	2	从左向右
7	==	相等运算符	2	从左向右
	!=	不相等运算符		
8	&	按位与运算符	2	从左向右
9	^	按位异或运算符	2	从左向右
10	|	按位或运算符	2	从左向右
11	&&	逻辑与运算符	2	从左向右
12	||	逻辑或运算符	2	从左向右
13	?:	条件运算符	3	从右向左
14	=、+=、 -=、 *=、 /=、 %=	赋值运算符	2	从右向左
15	,	逗号运算符	2	从左向右

自增、自减运算要注意前置和后置之分。前置是指操作数先自增/自减后，其值参与当前表达式的运算，后置是指先取操作数参与当前表达式的运算，然后再使操作数自增(后减)。

2. 类型转换

(1)在双目运算中，如果两个操作数的类型不一致，则自动进行类型转换。

(2)赋值运算时，若左右两边操作数的类型不一致，则将右边操作数转换成左边变量的类型。

除了自动类型转换外，有时需要进行强制类型转换。例如，由于两个整数相除的结果为整数，此时若希望结果为实型，则可将其中之一先强制转换为实型。强制类型转换的一般格式如下：

(数据类型名) 表达式

或

数据类型名 (表达式)

3. 表达式

用变量、常量、运算符、函数调用、圆括号等按一定规则连接起来的式子称为表达式。单个的常量、变量、函数调用等都是表达式。通常有算术表达式、赋值表达式、关系表达式、逻辑表达式等。在进行表达式运算时需要考虑结果的类型。

2.2　典型例题解析

【例 2.1】执行以下程序时，若依次输入：

123　　44　　a　bc

则输出结果是什么？

```
#include <iostream.h>
void main( )
{    inti,j;                          //A
     char k,s,t;                      //B
     cin>>i>>j;                       //C
     cin>>k>>s>>t;                    //D
     cout<<hex<<i<<'\t'<<j<<endl;     //E
     cout<<k<<s<<t<<'\n';             //F
}
```

【答案】7b　　2c

abc

【解析】程序中，A、B 行分别定义整型变量和字符型变量；C 行输入语句执行时将 123 存入变量 i，将 44 存入变量 j；D 行输入语句执行时分别将字符数据 a、b、c 存入变量 k、s、t，这里字符之间既可用空格或回车符分隔，也可以不用分隔符；E 行输出时，i 的输出值为十六进制数 7b，j 的输出值为十六进制数 2c；F 行输出 3 个字母字符 abc。

【例 2.2】写出下列程序的输出结果。

```
#include <iostream.h>
void main( )
{
     int a=2,x,y;
     x=a++ + ++a;                     //A
     cout<<a<<'\t'<<x<<'\n';
     y=5+(a++);                       //B
```