北京市高等教育精品教材立项项目

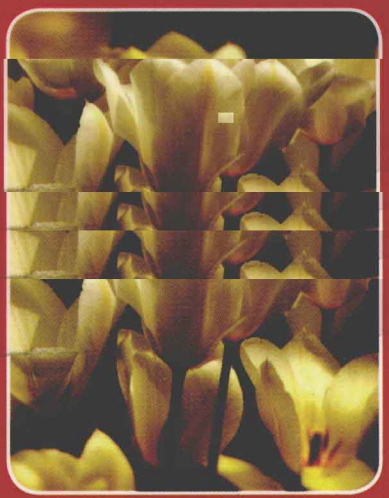
21世纪高等学校计算机规划教材
21st Century University Planned Textbooks of Computer Science

C++高级语言程序设计

Programming with C++

徐惠民 等 编著

- 学习程序设计基本方法　零起点学习C++
- 掌握C++语言基础知识　突出程序应用实践
- 了解面向对象思维方式　直击面向对象核心



精品系列

人民邮电出版社
POSTS & TELECOM PRESS

北京市高等教育精品教材立项项目

21世纪高等学校计算机规划教材
21st Century University Planned Textbooks of Computer Science

C++高级语言程序设计

Programming with C++

徐惠民 等 编著

精品系列

人民邮电出版社
北京

图书在版编目（CIP）数据

C++高级语言程序设计 / 徐惠民等编著. -- 北京 : 人民邮电出版社, 2011.2
21世纪高等学校计算机规划教材
ISBN 978-7-115-24632-5

Ⅰ. ①C… Ⅱ. ①徐… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2011)第007505号

内 容 提 要

本书将C++语言作为大学生学习程序设计的入门语言。全书共9章，第1章、第3章、第5章讲解程序的结构和组织，第2章、第4章、第6章讨论数据类型和数据的组织，第7章、第8章、第9章讲述面向对象的程序设计。

本书的编写目的是为学生打好程序设计的基础，因此，特别注意在介绍基本概念和基本方法的同时，重视良好编程习惯的培养。另外，专业术语配有英文。

本书内容全面，重点突出，深入浅出，同时配有大量习题，适合作为大学各专业的程序设计课程入门教材，也可以作为程序设计基础的培训教材和自学参考书。

21世纪高等学校计算机规划教材

C++高级语言程序设计

◆ 编　　著　徐惠民　等
责任编辑　贾　楠

◆ 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
北京鑫正大印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：17.75　　2011年2月第1版
字数：443千字　　2011年2月北京第1次印刷

ISBN 978-7-115-24632-5

定价：32.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第0021号

前　言

C++语言是一门优秀的程序设计语言，全面兼容 C 语言，保留了 C 语言的简洁、灵活、高效的特点，并且增加了面向对象程序设计的支持，从诞生以来一直受到广大编程人员的喜爱。

本书将 C++语言作为大学生学习程序设计的入门语言。由浅入深，循序渐进，并且把编程的思想融入实例中，力求使读者在掌握 C++语言的同时，能够对现实世界中较简单的问题及其解决方法用计算机语言进行描述。

本书主要有以下几方面的特点。

1. 将 C++语言作为学习程序设计的入门语言，不必有 C 语言的基础，可以在最短的时间内掌握一门面向对象的程序设计语言。

2. 学习语言的最终目的是要编程，编程的精髓是要体会计算机运行顺序的思想，所以书中实例的讲解都以内存的变化为依据，力求使读者在掌握 C++语言的同时，能够掌握编程的思路。

3. 精心选择内容，科学组织内容，便于读者掌握最重要的概念和方法。本书加强了“函数”的讲解，提供了一个完整的较大型的程序设计实例，希望读者能够掌握面向过程程序设计的精髓。面向对象程序设计只安排了 3 章，则是为了突出类、对象、继承和多态等基本内容。

4. 书中对 C++语言的难点提供了深入理解的引导，如“指针”部分、“继承和多态”部分，可以帮助初学者对于这些较难的内容逐步地深入理解。相信有经验的编程人员，阅读本书也会有很大收获。

5. 本书编写精心，内容精致，专业术语配有英文全拼注释，附录提供了常用的类库，内容全面且精练，重点、难点突出，可以针对学生分层次地开展教学。

本书的程序开发调试采用的是 Visual C++ 2005 集成开发环境。以前也曾使用 Visual C++ 6.0 的环境，但是，Visual C++ 6.0 环境对于标准 C++语言的支持不是很好（如对于友元函数的使用）。尽管 Visual C++ 2005 环境对于计算机硬件的要求高一些，但对于现在的大多数计算机来说是可以接受的。

本书的作者都是多年从事 C++教学和使用 C++语言进行项目开发的人员，可以说本书凝结了多人的智慧和心血。本书共 9 章，前 6 章涵盖了 C++语言基本程序设计，后 3 章是面向对象程序设计。第 1 章、第 7 章由刘瑞芳编写，第 2 章由许桂平编写，第 3 章和附录由肖波编写，第 4 章、第 5 章由赵衍运编写，第 6 章由徐惠民和杨文川编写，第 8 章、第 9 章由崔毅东编写。全书由徐惠民主编和统稿。

本书的电子教案可以通过人民邮电出版社教学服务与资源网（www.ptpedu.com.cn）获得。书中如有不当之处，欢迎广大读者批评指正。

编者

2010 年 11 月

目 录

第 1 章 C++初步

C++是一种优秀的程序设计语言，它完全兼容 C 语言，并且以其独特的语言机制在计算机科学领域中得到了广泛的应用。

本章将学习：

- C++语言的发展史
- 开发 C++程序的步骤
- 尝试自己的第一个 C++程序

1.1 程序设计语言的发展

1946 年世界上第一台电子计算机 ENIAC 诞生，当时对于计算机的控制使用的是**机器语言**（**Machine Language**）。机器语言是简单的“0”和“1”的组合，便于计算机识别，但对于人来说却晦涩难懂，难于记忆和使用，并且机器语言与 CPU 相关，不同 CPU 的计算机使用不同的机器语言。

20 世纪 50 年代末，出现了晶体管计算机，计算机运算速度从每秒几万次增加到每秒钟几十万次，**汇编语言**（**Assembly Language**）出现并发展起来。汇编语言将机器语言映射为一些可以被人们读懂的助记符，如“ADD”、“SUB”等，方便人们记忆和使用。但汇编语言也是与机器的 CPU 相关的。机器语言和汇编语言都属于低级语言。

随着 20 世纪 60 年代初期集成电路的出现，**高级语言**（**High-Level Language**）开始出现并逐步发展起来。高级语言是计算机编程语言的一大进步，人们不必了解机器的细节，通过更高层次的数据抽象使程序更能体现客观事物的结构和逻辑关系。这使得编程语言和人类的自然语言更加接近，但二者之间还是有很大的区别，因此程序设计语言仍然在不断地发展。程序设计语言的发展过程如图 1-1 所示。

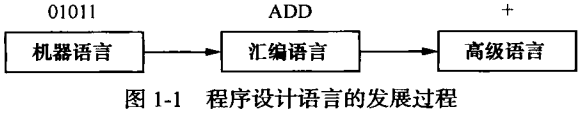


图 1-1 程序设计语言的发展过程

高级语言的发展也经历了不同的阶段。20 世纪 60 年代末出现了面向过程的程序设计语言（Procedural Programming Language），进一步提高了语言的层次。面向过程的程序设计语言通过结构化数据、结构化语句、数据抽象、过程抽象等概念使程序便于体现客观事物的结构和逻辑含义；

缺点是程序中数据和操作分离，程序中定义的变量不能完整反映现实世界的具体事物，软件可重用性较差。目前广泛应用的面向过程的高级语言有 BASIC、Pascal、C 等。

20 世纪 80 年代初出现了面向对象程序设计语言（Object-Oriented Programming Language），这种语言设计的出发点就是为了能更直接地描述客观世界中存在的事物（即对象）以及它们之间的关系。面向对象的编程语言将客观事物看成是具有属性和行为的对象，通过抽象同一类对象的静态特征（数据）和动态特征（操作或行为）形成类。目前，广泛应用的支持面向对象的高级语言有 C++、Java 等。

这些高级语言由于时代、思维方式、编程任务以及个人喜好等因素的不同，出现了许多描述方式和具体规则，将这些不同的描述方式称为不同的语言。

例如，要向屏幕打印一个“A”字符，不同的语言其描述方式是不同的。

BASIC 语言的描述：　　PRINT"A"

Pascal 语言的描述：　　writeln('A');

C 语言的描述：　　printf("A");

C++语言的描述：　　cout << "A";

Java 语言的描述：　　System.out.print("A");

对于面向过程的编程语言，使用不同的函数“print”、“writeln”和“printf”来实现输出字符功能；而对于面向对象的编程语言，则将打印功能作为系统对象的一个行为，使用不同的对象“cout”和“System”实现输出字符功能。

1.2　面向过程的程序设计

面向过程的程序设计又称为结构化程序设计，一般强调的是 3 种基本结构，以及模块的单入和单出。3 种基本结构指的是顺序、选择和循环结构；模块的单入指的是该模块被哪些模块所调用，单出指的是该模块调用了哪些模块。

以前程序被看做一系列处理数据的过程。一个过程或函数是指一个接一个执行的指令。数据与过程相分离，编程技巧主要是跟踪哪些函数调用了另一些函数以及哪些数据被改变了。为了避免发生混乱，出现了面向过程的程序设计。

面向过程的程序设计主要思想是：自顶向下，逐步求精。

一个计算机程序可以看成是由一系列任务组成的，任何一项任务如果过于复杂，就将其分解成一系列较小的子任务，直至每一项任务都很小、很容易解决。

例如：计算每门课程的平均成绩，可分为如下 4 个任务。

（1）一共有多少门课?

（2）每门课选课的学生总人数是多少?

（3）每门课所有学生的总分是多少?

（4）用每门课的总分除以每门课选课的学生人数，即可得出每门课程的平均成绩。

其中，第（3）项任务还可以分成以下 3 个子任务。

（1）找出每门课选课的学生档案。

（2）从档案中依次读出该课的成绩。

（3）累加到总成绩。

类似的，子任务（1）还可以分成以下 3 个子任务。

（1）选择一门课。

（2）查找选择该课的学生档案。

（3）从磁盘读出数据。

结构化编程是处理复杂问题的一种非常成功的方法，然而，到 20 世纪 80 年代末，结构化编程的不足逐渐暴露出来。

首先，结构化编程将数据和过程相分离，但客观事物的特性往往与此相背离，数据（如学生成绩）和对于数据的操作（排序、编辑等）是一个不可分割的整体。结构化编程重在过程，而不是数据和过程紧密联系在一起的对象。

其次，结构化编程对代码重用的支持不够，处理相似任务时程序员们不得不做大量重复的工作，而不能使用已有的代码。可重用思想就是创建一些已知属性的组件，然后插入到自己的程序中，这是一种模拟硬件组合的方式，类似于当工程师需要一个晶体管时，他不需要自己发明，只需要到仓库中取一个即可。

面向对象程序设计可以提供一种更加有效的手段，来尝试解决以上问题。

1.3　面向对象程序设计

面向对象的程序设计将数据和处理数据的过程当成一个整体——对象，并且具有以下 3 种特性：封装性、继承性和多态性。

1. 封装性

当一个技术人员要安装一台计算机时，他将各个部件组装起来。需要声卡时，不需要使用原始的集成电路或材料制作一个，只需要去购买一个声卡，插到机箱中即可。技术人员只需关心声卡的功能是否符合要求，不需要知道声卡的内部原理、硬件电路。声卡是自成一体的，这就是封装性（Encapsulation）。无须知道内部如何工作就能够使用的思想称为数据隐藏。

在面向对象程序设计中使用“对象”的概念支持封装。一个对象就是一个将数据和数据操作集合在一起的实体，只需要知道这个对象的外部接口，而不必知道对象的具体实现就可以使用它。

2. 继承性

当一个工程师要制造一辆新车时，有两种选择：或者从头做起，或者对已有的车型进行改进。如果现有的车型已经很好，只需要再添加一个新的变速装置或添加一个新的功能就更加完美了，那么没有必要从头来过，只需要利用现有车型，添加一些附加装置就可以产生一种新的车型。新的车型继承了原有车型的所有属性和行为，又增加了新的属性。这就是继承性（Inheritance）。

在面向对象程序设计中，使用“类”的概念支持继承。我们可以实现一个原有车型的类，然后对这个类进行扩展，从而产生一个新的类。新类是从已有的类派生出来的，不需要重新实现原有类的功能，只需要实现新添加的功能即可，从而实现了代码重用。

3. 多态性

对于不同的车型，当司机踩下油门时，车的反应是不一样的，可能这些不同的车型使用了不同的变速器。但作为司机不必知道这些差别，只需要知道驾驶的是哪一款车，跟这款车型配套的

机器自然运转即可。

面向对象程序设计支持这种思想，它使用相同的接口（踩油门），但不同的类（车型）运行状态（车的反应）不一样，这就是多态性（Polymorphism）。

关于C++语言的封装性、继承性和多态性更具体的描述，将在后续章节中给出。

1.4 C++的诞生

C++语言是从C语言发展演变而来的，因此我们首先回顾一下C语言的发展历程。

C语言最初是贝尔实验室的Dennis M.Ritchie在B语言的基础上开发出来的，1972年他在一台DEC PDP-11计算机上实现了最初的C语言。之后C语言作为UNIX操作系统的开发语言被人们广泛使用。C语言是与硬件无关的高级语言，并且由于C语言设计严谨，使得使用C语言编写的程序能够被移植到大多数计算机上。到20世纪70年代末期，C语言已经发展得比较成熟，Brian W.Kernighan和Dennis M.Ritchie于1978年合作出版了《The C Programming Language》一书，全面介绍了最初的C语言，是最早的、最经典的介绍C语言的书，该书只有两百多页，又被称为C语言的“圣经”。

C语言在各种计算机上的推广导致当时很快出现了多个不兼容的C语言版本，因此需要制定一种C语言的标准。美国国家标准化委员会在1989年通过了ANSI C标准，后来又被ISO接受为国际标准。Brian W.Kernighan和Dennis M.Ritchie根据ANSI C标准编著了《The C Programming Language（第2版）》，全面介绍了标准C的内容。

C语言是一种面向过程的编程语言，它具有如下优点。

- 语言简洁、紧凑、灵活。
- 具有丰富的运算符和数据类型。
- 可直接访问内存地址，能进行位操作。
- 程序运行效率高。
- 可移植性好。

但随着时代的发展，它的缺点也逐步暴露出来。

- 类型检查机制弱，导致许多错误不能在编译时被发现。
- 几乎不支持代码重用。
- 对于大规模程序，很难控制程序的复杂性。

随着程序规模的逐步扩大，C语言的局限性也越来越明显。为了满足管理大规模程序复杂性方面需要，1980年贝尔实验室的Bjarne Stroustrup开始对C语言进行改进和扩充。他根除了C语言中存在的问题，并使其支持面向对象的程序设计，将“类”的概念引入到C语言中，因此形成了最初的“带类的C”。1983年经过进一步改进，这种“带类的C”语言正式取名为C++。1985年Bjarne Stroustrup出版的《The C++ Programming Language》一书是最早介绍C++语言的经典著作，Bjarne Stroustrup也被誉为C++之父。Bjarne Stroustrup对《The C++ Programming Language》一书进行了3次改版，1991年出版第2版，1997年出版第3版，2000年出版特别版。

ANSI/ISO C++的标准化工作从1989年开始，第1版标准是在1998年通过的。第2版标准在2003年发布，即ISO/IEC 1482:2003。现在还在进行新标准的修订。

C++语言全面兼容 C 语言，它保持了 C 语言的简洁、高效等特点，比 C 语言更安全，并全面支持面向对象的程序设计，这极大地促进了 C++语言的发展。C 和 C++语言的关系如图 1-2 所示。Bjarne Stroustrup 曾这样描述 C++语言："C++是一种通用的程序设计语言，其设计就是为了使认真的程序员能觉得编写程序变得更愉快"。

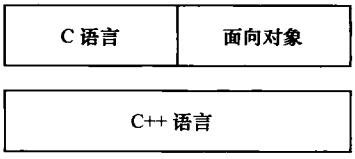


图 1-2　C 和 C++语言的关系

1.5　程序开发过程

大多数现代的编译程序都提供了一个集成开发环境（Integration Development Environment，IDE），本书选用的集成开发环境是 Microsoft Visual C++ 2005（简称 VC2005），所有的程序都在该集成环境下编辑、编译、运行。为了帮助读者更好地理解程序开发的过程，首先描述几个基本概念。

1. 源程序

使用源语言编写的、有待翻译的程序称为源程序。VC2005 集成环境支持 C++程序和 C 程序的编译和调试。源程序文件扩展名为.c 时，称为 C 源程序；在本书中，一律使用 C++语法编写的源程序，扩展名为.cpp，称为 C++源程序。

2. 目标程序

源程序经过翻译加工后所生成的程序称为目标程序。一般来说，目标程序使用机器语言表示（也称为目标代码），扩展名为.obj。

3. 可执行程序

目标程序和所用的其他资源进行链接，生成的可以直接运行的程序就是可执行程序。目标程序不可以直接运行，只有可执行程序可以直接运行。在 VC2005 环境下，可执行程序的扩展名为.exe。

4. 翻译程序

翻译程序是指用来将源程序翻译为目标程序的工具。对翻译程序来说，源程序是它的输入，目标程序是它的输出。翻译程序分成 3 类：汇编程序、编译程序、解释程序。

（1）汇编程序（Assembler）：将汇编语言编写的源程序翻译成机器语言形式的目标程序。

（2）编译程序（Compiler）：将使用高级语言编写的源程序翻译成机器语言形式的目标程序。在 VC2005 环境下，编译程序是安装目录下的一个名为"CL.exe"的文件，在集成开发环境中，一般使用"compile"命令进行编译。

（3）解释程序（Interpreter）：将使用高级语言编写的源程序翻译成机器指令。它与编译程序的区别在于解释程序是边翻译边执行，即输入一句、翻译一句、执行一句，直到将整个源程序翻译并执行完毕。解释程序不产生完整的目标程序，因此对于源程序中循环执行的语句需要反复解释执行，效率较低。Basic 语言就是典型的使用解释程序的编程语言。

5. 链接程序

链接程序（Linker）是用来对汇编程序或编译程序生成的目标程序与所需的其他资源进行链接，生成可执行文件的程序。对链接程序来说，目标程序是它的输入，可执行程序是它的输出。在 VC2005 环境下，链接程序是安装目录下的一个名为"LINK.exe"的文件，在集成开发环境中，

一般使用“build”命令进行链接。

VC2005 集成开发环境是使用编译方式进行程序开发的。在该环境下，开发 C++程序的步骤如图 1-3 所示。

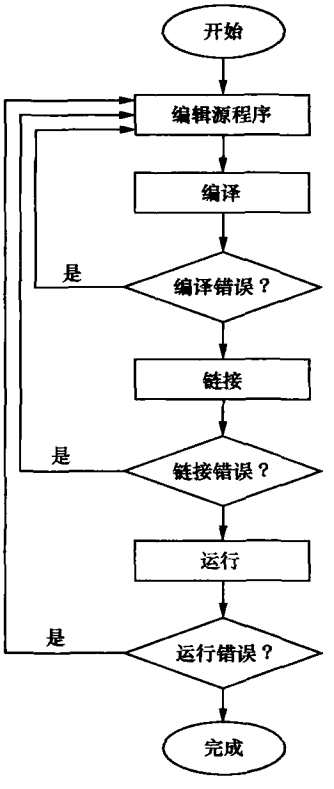


图 1-3　开发 C++程序的步骤

1.6　最简单的 C++程序

下面从一个最简单的程序入手，来学习和分析 C++程序的构成。

例 1-1　用 C++程序在屏幕上输出“Hello World!”字样。

解：

```
/************************************************
程序文件：ch1_1.cpp
程序功能：在屏幕上输出 Hello World!
作　　者：×××
创建时间：××年××月××日
输　　入：无
输　　出：字符串 Hello World!
************************************************/
#include <iostream>
using namespace std;
```

```
void main()
{
  cout<<" Hello World!"<<endl;
}
```

运行结果：

```
Hello World!
```

C++程序由注释、编译预处理、程序主体组成。

1．注释

注释是程序员为程序语句所作的说明，用来提高程序的可读性。C++程序在编译过程中忽略注释。在 C++中使用两类符号来标明注释。

- “//”用来注释一行说明，“//”之后的文字直到换行都为注释。一般用来对于程序中难懂的语句进行说明。
- “/*”和“*/”用来注释一段说明，“/*”和“*/”之间的部分，不管多长都为注释，一般用在程序或函数的开头，说明程序、函数或文件的名称、用途、编写时间以及输入/输出等。注意：“/*”和“*/”必须成对出现。

2．编译预处理

所有以“#”开头的代码，称为编译预处理。例如，程序清单的第一行代码为：

```
#include <iostream>
```

每次启动编译器时，先运行预处理器，预处理器找到所有以“#”开头的代码行并进行处理。include 是一条预处理指令，意思是“后面跟的是一个文件名，请找到该文件并将其加入”。

本例中包含的文件是 iostream，该文件是系统定义的一个“头文件”，它设置了 C++的 I/O 相关环境，并定义了输入/输出流对象 cout 和 cin 等。本例中应用了 cout 对象将字符输出到屏幕，因此需要将 iostream 文件使用#include 预处理指令引入。

1998 年批准的标准 C++使用 namespace（命名空间）标准。iostream 是一个标准函数库，cout 是标准库函数提供的一个对象，标准库函数在 namespace 说明书中被指定为“std”命名空间。因此如果想要在代码中使用标准库函数，就必须加入代码“using namespace std;”，这句代码的意思是使用标准命名空间 std 中的函数。

3．程序主体

正式的程序从代码“void main()”开始，它包含一个名为 main()的函数，也称为主函数。每个 C++程序有且仅有一个 main()函数。函数是指能实现一个或多个功能的代码块。函数通常都是由其他函数调用的，而 main()函数比较特殊，程序在开始运行时会自动调用 main()函数。

main()前面的 void 表示函数返回值的类型，标识出程序执行结束后将向操作系统返回什么，是一个整数还是一个实数。这里使用 void 表示不返回任何信息。

主程序 main()通常还有另外一种写法：

```
int main()
{
    cout<<" Hello World!"<<endl;
    return 0;
}
```

int main()表示函数要返回一个整数，与此对应的是语句“return 0;”，代表它返回值为 0，一般来说，返回值为 0 代表程序运行正确。

所有的函数都以左大括号{开始，右大括号}结束，位于大括号{}之间的部分称为函数体。

函数体中的第二句代码“cout<<" Hello World!"<<endl;”将一个字符串“Hello World!”显示到屏幕上。其中，cout 是标准输出流对象，“<<”是插入操作符，可以连续多次使用，“endl”代表换行符，因此代码“cout<<" Hello World!"<<endl;”的意思是使用 cout 将“<<”后面的内容显示在屏幕上。

函数体内的每一句代码后面都有一个分号“;”表示一个 C++语句的结束。

1.7 内存的抽象表示和使用

在学习编程时，了解内存的工作模型很重要，因为自计算机诞生以来，其工作原理一直沿用冯·诺依曼提出的“存储程序”的原理。

我们都知道，一个程序要执行时一定会先复制到内存，然后由 CPU 逐句（一条指令一条指令）地读取过来再执行。我们把内存抽象表示为如图 1-4 所示的形式。

每个存储单元可以存放一个字节（8 bit）数据，每个内存单元有一个唯一的地址。一般来讲，地址是顺序编址的，绝大部分计算机按字节顺序编址。CPU 按地址读取内存中的指令和数据，有时也把计算结果按地址存放到某个内存单元，称为 CPU 访问内存（进行取/存操作，读/写内存中的信息）。

如果一台计算机安装有 256 MB 内存，它有 256 × 1024 × 1024 个内存单元，如果用 7 位十六进制数表示它的地址值，那么地址范围是 0x0000000～0xFFFFFFF。C++用 0x 表示十六进制数。

操作系统一般会把内存划分区域来使用，以便于管理，如代码区、数据区等。被编译成机器码的程序在执行时会被复制到内存的代码区。程序中的变量和常量会被存放到数据区。在后续编程序和调试时，我们经常会看到某段内存区域在某时刻的“快照”。

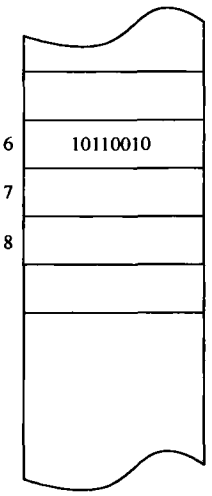


图 1-4 内存的抽象表示

数据区分为如下几种情况。

栈区，也叫堆栈区，用于存放程序函数中的局部变量。栈区中的变量也叫自动变量，用到某个函数时，该函数中定义的变量就保存在栈区，退出函数时，相应的变量会自动释放。栈区的操作还有一个特点——“先入后出”，即先进栈的变量后退出。

全局变量和静态变量区是存放长期数据的区域。当一个变量被定义为全局变量或者静态变量时，从程序开始执行到结束，它都会在内存中占有固定的字节。

常量区一般是存放字符串常量的地方。

堆区，在程序执行过程中申请的内存空间属于堆区，这些申请的空间也应该在程序中释放。

可以通过一个程序来观察上述各种区域。

例 1-2 显示各种不同的数据的地址。

解：

程序如下。

```
//例 1-2 显示不同数据的地址
#include <iostream>
```

```
using namespace std;
int a=10;                                          //全局变量
void main()
{
    int b=20,c=30;                                 //局部变量
    char *ch="Beijing";
    static int e=50;                               //静态变量
    int *p =new int(60);                           //申请堆区空间
    float *f=new float(0);
    int *q =new int(60);
    cout<<"全局变量 a 的地址："<<&a<<endl;
    cout<<"局部变量 b 的地址："<<&b<<endl;
    cout<<"局部变量 c 的地址："<<&c<<endl;
    cout<<"常量区的地址："<<(void*)ch<<endl;
    cout<<"静态变量 e 的地址："<<&e<<endl;
    cout<<"堆区变量 p 的地址："<<p<<endl;
    cout<<"堆区变量 f 的地址："<<f<<endl;
    cout<<"堆区变量 q 的地址："<<q<<endl;
}
```

程序运行结果：

```
全局变量 a 的地址：00474DC0
局部变量 b 的地址：0012FF7C
局部变量 c 的地址：0012FF78
常量区的地址：0046C0D8
静态变量 e 的地址：00474DC4
堆区变量 p 的地址：00481FF0
堆区变量 f 的地址：00481FC0
堆区变量 q 的地址：00481F90
```

全局变量和静态变量位于同一个区域，先定义的放在低地址，后定义的放在高地址。局部变量则相反：先定义的放在高地址，后定义的放在低地址。

这一段的内容，现在大家不一定能理解得很好，也不需要关心这个程序是如何编出来的。关键是要逐渐熟悉程序执行过程中，数据如何在内存中发生变化。

本章小结

程序设计语言的发展过程依次经历了机器语言、汇编语言和高级语言。C++是一种与硬件无关的高级语言，它在 C 语言的基础上发展起来，全面兼容 C 语言。学习 C++语言时，不一定非要先学习 C 语言。学好 C++也就学会了 C 语言。

一个 C++程序需要经过编辑、编译和链接，才能产生可执行文件。

C++程序由函数构成，每个程序有且仅有一个名为 main()的函数，程序总是从 main()函数开始运行。

习题和思考题

1.1 在 VC 集成开发环境中，要产生一个可执行的 EXE 文件的步骤是什么？

1.2 C 语言与 C++语言的关系是什么？

1.3 结构化程序设计与面向对象程序设计有什么异同点？

1.4 面向对象程序设计的基本特征是什么？

1.5 为了编辑和运行 C++程序，在 VC 环境下已经建立了一个工程 Proj01，也建立了一个 C++文件 file01.cpp。现在有一个 C++程序 input.cpp，希望调入到这个工程中编译和运行，应该如何操作？

1.6 C++是否可以输出中文字符串？仿照例 1-1 编写程序，实现在屏幕上显示“北京欢迎你”。