



普通高等教育“十五”国家级规划教材  
面向 21 世纪课程教材



High Level Language C++  
Program Design

# 高级语言C++程序设计

( 第 二 版 )

刘 璟 周玉龙



高等教育出版社  
Higher Education Press

普通高等教育“十五”国家级规划教材  
面向 21 世纪课程教材

# 高级语言 C++ 程序设计

(第二版)

刘 璟 周玉龙

高等教育出版社

## 内容简介

本书是普通高等教育“十五”国家级规划教材，书中详细介绍了 C++ 程序设计语言的语法规则和编程方法，同时通过编程实例讲解如何使用 C++ 语言求解与实现具有应用背景的各种具体问题，从而提高读者的编程与动手能力，为进行软件开发及学习后续课程打下良好的基础。

本书主要内容包括：C++ 的各种基本数据类型、派生类型及其运算，分支、循环结构与函数的编程技巧，类与对象、继承与派生的概念以及面向对象程序设计（OOP）方法的学习与应用，模板与输入输出流的使用以及异常处理等。另外在附录中介绍了 Visual C++ 编程环境，列出了最常用的 C++ 标准函数，并对标准模板库（STL）的使用方法与技术做了简要说明。

本书结构清晰、系统性强、叙述简洁、重点突出，适合作为计算机专业和非计算机专业学生学习高级语言程序设计及面向对象技术的教材，也可供自学 C++ 的读者使用。

## 图书在版编目（CIP）数据

高级语言 C++ 程序设计 / 刘景，周玉龙. —2 版.  
北京：高等教育出版社，2004.11（2006 重印）

ISBN 7-04-015496-X

I. 高... II. ①刘... ②周... III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字（2004）第 067265 号

策划编辑 陈红英 责任编辑 彭立辉 封面设计 于文燕 责任绘图 尹莉  
版式设计 王莹 责任校对 杨雪莲 责任印制 陈伟光

出版发行 高等教育出版社  
社 址 北京市西城区德外大街 4 号  
邮政编码 100011  
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司  
印 刷 涿州市星河印刷有限公司

开 本 787×1092 1/16  
印 张 29  
字 数 620 000

购书热线 010-58581118  
免费咨询 800-810-0598  
网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.landaco.com>  
<http://www.landaco.com.cn>  
畅想教育 <http://www.widedu.com>  
版 次 1999 年 6 月第 1 版  
2004 年 11 月第 2 版  
印 次 2006 年 7 月第 4 次印刷  
定 价 35.90 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 15496-00

## 作者简介



**刘 璟** 南开大学信息技术学院计算机科学技术系教授，博士生导师，兼任教育部计算机科学与技术教学指导委员会委员，基础分会副主任，天津市高等学校计算机基础教学指导委员会副主任，中国计算机学会理论计算机科学分会理事，天津市学位委员会学科评议组成员。

长期讲授“高级语言程序设计”、“算法设计与分析”、“程序设计方法学”、“数据结构”等课程。主要研究领域为并行与分布式系统、算法设计与分析、网络存储系统、计算智能与字符图像识别、面向对象程序设计等。曾主持国家 863、科技攻关、自然科学基金、博士点基金项目等十余项研究课题，在国内外发表论文 60 篇，出版教材《计算机算法引论》、《高级语言 C++ 程序设计》、《C++ 语言速成教程》等。



**周玉龙** 南开大学计算机科学与技术系副教授。自毕业留校任教后一直从事计算机软件与应用方面的教学与科研工作，编写并出版有多本与计算机程序设计有关的书籍和教材，主讲多门计算机软件与应用专业的本科生及研究生的课程。主要研究领域为计算机应用以及面向对象程序设计与编程技术、机器学习与机器翻译及智能软件的研究与开发技术、数据库管理与应用技术等。

# 教育部高等学校非计算机专业计算机基础课程教学 指导分委员会推荐教材出版说明

进入 21 世纪之后,我国明显地加快了建设世界教育大国的步伐,现在正向世界教育强国的目标迈进。实现这个历史性任务的最为关键指标是要有国际公认的高等教育质量,而高水平的教材是一流教育质量的重要保证。

在“九五”和“十五”期间,两届计算机基础课程教学指导委员会都把教材建设列为重点工作。非计算机专业计算机基础课程的教育部“面向 21 世纪课程教材”和“普通高等教育‘十五’国家级规划教材”均取得了可喜成果,教材被选用率高,不少还被评为国家、省部级的优秀教材。

本届教学指导分委员会一直着力于研究在新形势下,如何进一步加强高校的计算机基础教学。提出了许多重大的改革举措、新的课程体系框架,计算机基础教学的内容组织和课程设置已反复与各高校教务部门、有关教师研讨,取得许多共识;更令人兴奋的是广大高校表现出极大的热情,一批有创新、改革精神,且有丰富教学经验的教师积极投身到新一轮的计算机基础课程教材编写中。我们对这些教师表示深深的敬意,感谢他们用自己创造性的思维、辛勤的汗水诠释本届教指委的改革思想,把教指委新设计的课程体系和教学内容生动地传达给师生,进行有意义的教学实践。

为了把计算机基础教育的优秀教材及时地推荐给广大从事计算机基础教育的教师和学生,便于他们选用和研究,我们新设计开发了本届教指委组织推荐的“计算机基础课程系列教材”,并将已经出版和即将新出的部分“面向 21 世纪课程教材”、“普通高等教育‘十五’国家级规划教材”与这些新编教材进行了整体规划,系统组织,内容严格把关,形成符合新的教学基本要求的新的教材体系,希望这些教材的出版能起到推动计算机基础教育改革的作用,使我们高校的计算机基础教育质量更上一个台阶。

计算机基础教育改革一直在不断地深化,课程体系和教学内容趋于更加合理和科学。本系列教材与以前出版的教材比较会有较大的变化,这也是我们期待的。

每一本教材都有它的适用范围,面向不同办学层次、学科、地域和人才培养模式的教材必然有差异。本系列教材将会考虑这种差异,以满足各种层次和类型的教学所需。

列入本系列的教材,当在国内同类教材的优秀之列,我们希望作者把它打造成国家级的精品教材,要求做到“三新”,即体系新、内容新、方法新;每一本教材都做成既有文字教材、又有电子教材,既有教科书、又有辅助教材,成为真正意义上的“立体化”。教材的出版仅是“万里长征的第一步”,要成为精品教材,作者还必须根据读者的反映和需求不断修订原作,

真正做到“与时俱进”。

“一切为了教学，一切为了读者”是我们的心愿，书中不足之处，恳望教师和同学们指正。

教育部高等学校非计算机专业计算机基础课程教学指导分委员会

2004年6月

## 第二版前言

经过一年多的努力，本书（第2版）终于和读者见面了。感谢专家和有关领导把这项修订工作列入国家重点教材（“十五”国家级规划教材）的规划之中。

新版在第一版的基础上做了较大的变动，力求在内容上有所突破。为了与程序设计技术的最新发展及 C++ 语言标准（ISO/IEC 14882 Standard for the C++ Programming Language）相一致，书中增加了一些新概念（例如 namespace 等），把异常处理机制作为一章供读者选读，以附录的方式介绍了标准模板库（STL），引导读者了解有关容器类、算法库的概念和用法。为了有利于读者学习一些后续课程（数据结构与算法等）和 STL 的使用，还加强了模板（template）等内容的介绍。同时，在修订过程中充分体现了 C++ 语言的发明人 B. Stroustrup 多次提出的观点，强调诸如名字空间、模板、内联函数、常量等“正确的语言功能”，而对于位操作、联合（union）、宏定义、强制类型转换、C 风格的字符串和多维数组下标的指针运算等，在 C++ 程序中的使用不宜受到鼓励。另外，为了使本书内容深入浅出，易教易学，我们对第一版的内容布局做了一些调整，每一章节的叙述和程序实例都做了增补和修订。

与第一版的宗旨一样，程序设计课程的重点应是培养学生的编程能力，学习这门课程的目标是培养好的程序员而不是为了应付社会上某些（不适当的）“等级考试”。因此，在讲清 C++ 语言语法规则的同时，注意加强程序实例的选择和配置，尽量选用实用的程序（而不是仅用来说明语法），在每一章讲解语法之前后，都配有完整的 C++ 程序。另外，强化并丰富了各章的练习题，以逐步提高学生读程序、写程序的能力。面向对象程序设计主要用于规模较大的软件开发，单纯学习 C++ 语言的语法规则是不够的，因此本书在第 11 章专门介绍较为完整的 C++ 语言实现的 OOP 程序供选讲或选读（本书还有其他一些章节带有“\*”号，也是供选讲或选读的内容）。

把 C++ 语言作为高级语言程序设计的教学语言对于教师和学生都是有难度的，与 C 语言相比，语言的规模和深度有很大的差别，不过程序设计技术的发展要求我们必须迎接这一挑战。目前，不仅许多计算机专业和多数软件学院的程序设计课程选择了 C++ 语言作为第一门语言，而且越来越多的理工科专业在开设 C++ 语言程序设计课程。实际上，如何使初学者在有限的课时内打下良好的程序设计基础，目前仍是一个探索性的课题，我们愿与老师和同学一起，把这项工作做好。

本书的修订工作主要由刘璟和周玉龙完成，刘璟负责 1~6 章和 11 章的修订并统编全书，周玉龙负责 7~10 章和习题及附录 A、B 的修订并增编了第 12 章和附录 C。另外，使用本书

第一版的老师和同学提出的许多宝贵的意见和建议对本书的修订帮助很大，在此一并致谢。本书在编写过程中，我们虽然全力以赴，但由于水平所限，不妥之处在所难免，恳请同行专家和采用本书的老师和同学批评指正。

编者

2004年4月



# 第一版前言

本教材是为高等学校计算机及相关专业的高级语言程序设计课程编写的。

用 C++ 语言取代 Pascal 语言（或者 C 语言），使教学内容上发生了巨大的变化；把程序设计技术的重点从结构程序设计转向面向对象程序设计，是一个重大变革。作者曾经与一些计算机专家讨论，大家比较一致的看法是：

- 面向对象程序设计（Object Oriented Programming, OOP）技术是目前程序设计与软件开发的主流，是培养计算机专业人员的重点内容，过去只在高年级开一个选修课的方法不能继续下去了。现在国家把发展软件产业作为科教兴国的重大战略任务提出来，培养高级编程人才首当其冲。程序设计课程的变革可以说是当务之急。

- 讲授 OOP 技术最合适的语言是 C++ 语言。一方面它是应用最广的 OOP 语言，另一方面，也有利于学生学习一般的编程技巧。不过，一定不要把它作为 C 语言的扩充来讲授。

然而，在如何进行以 OOP 技术为主的程序设计教学问题上，专家们的意见却不尽统一。问题主要在于，C++ 语言的规模和 C++ 程序的复杂程度，往往使得教师们面对刚刚步入大学的学生时感到困难重重，比起十几年前讲 BASIC 语言不知要难多少倍！

不过，从目前国内外的计算机教学的发展来看，从一开始就讲授 C++ 语言是大势所趋，因为这是现代软件技术的需要。早在 20 世纪 90 年代中期，南开大学、中山大学等高校已经开始采用 C++ 语言作为程序设计主干课的内容，并取得了一些经验。尽管如此，教材的编写仍然是一个艰巨而具有探索性的工作。在本书的编写过程中，作者始终把讲授编程技术作为主线，努力使其不要淹没在大量语法规则的讲解之中；虽然一些 C++ 的实用程序都很长，但仍然尽量多引用高水平的程序实例，避免像许多 C++ 教材中那样把实例简化到仅用来说明语法，成了一本语法说明书。我们的教材就是为了培养高水平编程人员的。另一方面也应指出，掌握现代编程技术是一个长期的过程，同学们应在后继课程（如算法与数据结构、编译技术、数据库技术、软件工程、Java 语言等）中不断地提高和完善。

在编写本书的过程中，得到教育部高等学校计算机科学技术教学指导委员会和高等教育出版社的专家和编辑的鼓励，天津大学边莫英教授详细审阅了全文，并提出了许多宝贵意见，南开大学周玉龙、于春凡副教授在原稿中发现了许多错误和不妥之处，邵秀丽副教授和研究生熊伟、罗艳以及董沙沙为本书选编了习题，编写了 3 个附录，检查了书中的应用程序，并完成了全部的整理工作，在此向他们表示衷心的感谢。

在编写本书的过程中，作者参阅了国内外许多程序设计语言与数据结构方面的书籍，从

中吸收了新的思想、新的内容，同时又力图有所突破、有所创新，然而能力和水平有限，可能有许多错误和不足之处，敬请阅读本书的老师 and 同学予以指正。

刘 璟

2000 年于南开大学

# 目 录

<b>第一章 绪论</b> ..... 1	<b>2.4 C++程序的基本框架</b> ..... 34
1.1 程序设计与程序设计语言..... 1	2.4.1 主函数..... 34
1.1.1 计算机与程序设计..... 1	2.4.2 预处理命令..... 37
1.1.2 程序设计语言..... 2	2.4.3 C++程序的 SP 框架..... 41
1.1.3 程序设计方法学的发展..... 4	2.4.4 C++程序的 OOP 框架..... 43
1.1.4 程序设计的范型..... 7	<b>2.5 运行 C++程序</b> ..... 44
1.1.5 程序设计技术的 4 个层次..... 8	2.5.1 编辑 C++程序..... 44
1.2 C++语言概述..... 11	2.5.2 编译和链接过程..... 45
1.2.1 为什么选择 C++语言..... 11	2.5.3 运行一个简单的实例..... 46
1.2.2 C++语言简史..... 12	<b>思考题</b> ..... 47
1.2.3 C++语言的特点..... 13	<b>练习题</b> ..... 48
1.3 本书的宗旨及内容安排..... 15	<b>第三章 基本数据类型与基本运算</b> ..... 50
1.3.1 讲授 C++语言的困难..... 16	3.1 包含简单计算的 C++程序..... 50
1.3.2 本书的指导思想..... 18	3.1.1 程序实例——求两数之和..... 50
1.3.3 本书的内容安排..... 19	3.1.2 程序实例——计算圆面积..... 51
<b>思考题</b> ..... 20	3.2 基本类型及其派生类型..... 52
<b>第二章 C++语言初步</b> ..... 21	3.2.1 数据类型的概念..... 52
2.1 初识 C++程序..... 21	3.2.2 基本类型..... 54
2.1.1 程序实例..... 21	3.2.3 基本类型的派生类型..... 55
2.1.2 I/O 语句..... 23	3.2.4 enum 类型..... 56
2.2 C++语言的基本符号..... 26	3.2.5 定点类型与浮点类型..... 57
2.2.1 基本符号分类..... 26	3.3 说明语句..... 58
2.2.2 基本符号的 ASCII 编码..... 26	3.3.1 语句..... 58
2.3 C++语言的词汇..... 28	3.3.2 常量和变量..... 59
2.3.1 关键字..... 28	3.3.3 常量说明..... 59
2.3.2 标识符..... 30	3.3.4 变量说明..... 60
2.3.3 字面常量..... 30	3.3.5 名字空间..... 64
2.3.4 运算符..... 33	3.3.6 类型说明..... 65
2.3.5 分割符..... 34	3.4 基本运算符..... 66

3.4.1 运算符和表达式的概念	66	4.6.2 结构变量的引用和赋值	107
3.4.2 运算类型与运算符	67	4.6.3 结构数组	107
3.4.3 赋值运算	67	4.7 C++程序实例	108
3.4.4 算术运算	68	4.7.1 统计学生成绩	108
3.4.5 关系运算	69	4.7.2 输出三角函数表	112
3.4.6 逻辑运算	70	4.7.3 画一个四叶玫瑰线图形	114
3.4.7 位运算	71	4.7.4 Eratosthenes 筛法求素数	116
3.4.8 其他运算	72	思考题	119
3.4.9 运算的优先级	77	练习题	119
3.4.10 运算与运算符小结	78	<b>第五章 函数、函数与运算符的重载</b>	127
3.4.11 关于 C++语法及其 实现系统的注释	80	5.1 三次方程求根程序的设计	127
思考题	80	5.2 函数的说明与使用	130
练习题	81	5.2.1 函数的说明	130
<b>第四章 基本控制结构与导出数据类型</b>	83	5.2.2 函数的调用	132
4.1 控制语句、复合语句和空语句	83	5.2.3 函数的返回	132
4.1.1 简单的计算器程序	83	5.2.4 函数的参数	133
4.1.2 控制语句	87	5.2.5 值调用与引用调用	134
4.1.3 复合语句和空语句	87	5.2.6 内联函数	137
4.2 分支语句	88	5.3 函数的嵌套与递归	138
4.2.1 两类分支语句	88	5.3.1 函数嵌套	138
4.2.2 温度值变换程序	91	5.3.2 函数的递归	140
4.3 循环语句	94	5.4 函数与运算符的重载	142
4.3.1 3种循环语句	94	5.4.1 函数重载	142
4.3.2 求素数	96	5.4.2 可重载运算符	144
4.3.3 计算常数 e 的值	98	5.4.3 运算符重载函数的定义	145
4.4 转向语句	99	5.5 函数与 C++程序结构	146
4.5 导出数据类型和数组	101	5.5.1 库函数的使用	146
4.5.1 导出类型的概念	101	5.5.2 SP 框架结构	147
4.5.2 一维数组	102	5.5.3 函数间的数据传递	148
4.5.3 多维数组	104	5.5.4 变量与函数的作用域	150
4.5.4 数组与字符串	105	5.6 程序实例	153
4.6 结构类型	105	5.6.1 “三色冰淇淋”程序	153
4.6.1 结构类型与结构变量说明	106	5.6.2 Hanoi 塔问题	155
		思考题	157

练习题	158	进行重载	235
<b>第六章 指针、引用与动态内存分配</b>	166	7.8.2 利用运算符重载实现 集合类型	238
6.1 选择排序算法	166	7.9 结构与联合	243
6.2 指针类型	170	*7.10 拷贝构造函数	243
6.2.1 指针变量说明	170	*7.10.1 概述	243
6.2.2 指针变量的操作	171	*7.10.2 显式拷贝构造函数的使用	244
6.2.3 指针与数组	174	思考题	245
6.2.4 字符串指针	175	练习题	246
6.2.5 指针与函数	178	<b>第八章 继承与派生</b>	251
6.3 指针与动态内存分配	181	8.1 公司雇员档案的管理	251
6.3.1 动态分配运算符	181	8.2 派生类说明及其构造和析构函数	255
6.3.2 用指针进行内存动态分配	182	8.2.1 派生类说明	255
6.4 引用类型	183	*8.2.2 有关成员存取权限 问题的进一步讨论	257
6.5 程序实例	186	8.2.3 派生类的构造函数和 析构函数	259
6.5.1 按人名字典序排列电话簿	186	8.3 其他特征的继承关系	263
6.5.2 构建人员档案链表	189	8.3.1 友元关系以及静态 成员的继承	263
思考题	192	8.3.2 与基类对象和派生类对象 相关的赋值兼容性问题	263
练习题	194	8.4 派生关系中的二义性处理	265
<b>第七章 类与对象</b>	201	*8.5 虚基类	268
7.1 设计一个栈类	202	*8.5.1 虚基类一般应用示例	269
7.2 类和对象的说明	205	*8.5.2 具有显式有参构造函数 的虚基类的初始化问题	270
7.3 对象的初始化、构造与析构函数	209	8.6 多态性与虚函数	272
7.3.1 基本概念及定义	209	8.6.1 函数重载与静态联编	273
7.3.2 构造与析构函数使用示例	211	8.6.2 函数超载、虚函数及 动态联编	273
7.4 类的定义及其使用	213	8.6.3 纯虚函数与抽象基类	277
7.4.1 创建一个集合类型	213	8.7 虚函数使用实例	278
7.4.2 利用 stack 类型解迷宫问题	218	8.7.1 计算函数的定积分	278
7.5 类的静态成员及常量成员	224		
7.5.1 类的静态成员	224		
7.5.2 类的常量成员	227		
7.6 友元	229		
7.7 类之间的关系	232		
7.8 自定义类中的运算符重载	235		
7.8.1 以两种方式对运算符			

8.7.2 利用图元类画图的程序·····	283	10.4.2 使用插入与提取运算符 对磁盘文件进行读写·····	343
*8.7.3 在 VC6.0 下实现利用 图元类画图的程序·····	288	10.4.3 使用类成员函数对文件流 (类对象) 进行操作·····	345
思考题·····	294	10.5 text 文件与 binary 文件·····	350
练习题·····	294	10.5.1 按用户设置的文件形式 进行读写·····	351
<b>第九章 模板</b> ·····	299	10.5.2 对数据文件进行随机访问·····	354
9.1 函数模板·····	299	*10.6 字符串流·····	358
9.1.1 函数模板的概念及说明·····	299	10.7 其他输入输出控制函数·····	361
9.1.2 函数模板应用举例·····	302	10.7.1 I/O 操作状态控制·····	361
9.2 类模板·····	304	*10.7.2 其他 I/O 控制·····	363
9.2.1 一个队列类模板·····	304	思考题·····	364
9.2.2 类模板说明·····	307	练习题·····	365
9.2.3 使用类型参数和普通 参数的类模板·····	309	<b>*第十一章 用 C++ 语言设计面向 对象程序</b> ·····	370
9.3 关于类模板的若干问题说明·····	311	*11.1 一个 Palindrome 的识别程序设计·····	370
9.3.1 静态成员及友元·····	311	*11.2 楼宇电梯系统仿真程序的设计·····	373
9.3.2 特例版本·····	312	*11.2.1 电梯仿真程序的 初步框架·····	375
9.3.3 按不同方法派生类模板·····	314	*11.2.2 电梯运行系统 building·····	377
9.4 程序实例——链表类模板的设计·····	318	*11.2.3 电梯仿真程序的总体框架·····	379
思考题·····	322	*11.2.4 人员类 person 与人员 集合类 persSet·····	383
练习题·····	322	*11.2.5 楼层类 floor 与楼层 集合类 floorSet·····	391
<b>第十章 输入输出流</b> ·····	325	*11.2.6 电梯类 elevator 与电梯 集合类 elevSet·····	395
10.1 文件、流及 C++ 的流类库·····	325	*11.2.7 小结·····	402
10.1.1 流类库的优点·····	325	思考题·····	403
10.1.2 文件与流的概念·····	326	练习题·····	403
10.1.3 C++ 的流类库·····	327	<b>*第十二章 异常处理</b> ·····	405
*10.2 插入与提取运算符的重载·····	329	*12.1 异常处理的基本思想及实现机制·····	405
10.3 I/O 的格式控制·····	332	*12.2 多级多层次捕获与处理·····	407
10.3.1 用于格式控制的 ios 类 成员函数·····	332		
10.3.2 格式控制符·····	337		
*10.3.3 用户定义格式控制符·····	339		
10.4 磁盘文件的输入输出·····	340		
10.4.1 文件的打开与关闭·····	341		

---

*12.3 系统自动进行“堆栈展开”过程·····	410	附录 A Visual C++ 6.0 编程环境简介·····	415
思考题·····	412	附录 B 标准函数·····	428
练习题·····	413	附录 C 标准模板库·····	432
附录·····	415	参考文献·····	446

# 第一章 绪 论

## 1.1 程序设计与程序设计语言

半个多世纪以来，计算机技术无论作为科学学科还是作为现代产业，都已从一棵幼苗成长为枝繁叶茂的参天大树。回顾其发展历程，计算机科学与工程的发展一直是围绕着程序设计这个中心课题进行的。

### 1.1.1 计算机与程序设计

计算机是人类文明历史上最伟大的发明之一，它对人类生活产生了巨大的影响。人们把计算机与人类其他伟大发明，如飞机、电灯、汽车、电视机等相比较，从中发现它的作用、影响和前景。

计算机与其他发明的主要差别有如下两点：

(1) 人类的发明都是对自己的器官的延长或替代，而计算机则是人类大脑功能的延伸或某种替代，所以被称为“电脑”。

(2) 人类的发明可以应用在各个不同的局部领域，计算机的与众不同之处在于它可以应用在几乎所有的人类活动领域。

目前，计算机的这两个特征还在发展之中，它可以在怎样的程度上延伸或代替大脑的活动，可以在何种程度上被广泛而深入地应用于各个领域，目前还不能确定，不过现在可以指出的是，使计算机具有如此影响力的根本原因是：计算机不是一个一次性的直接服务产品，它为人类服务是有条件的，这个条件就是程序设计。

没有程序和程序设计，计算机将不能做任何事情，换一个说法：程序（软件）是计算机的必要组成部分。计算机首先要求人们不断地在程序设计上付出大量的创造性劳动，然后才能享受到它的服务。

计算机本身是人类智慧的产物，它的诞生又导致了人们投入十倍、百倍的精力和智慧用于程序设计和软件开发，从而引发出无穷无尽的新的发明创造。

有关计算机科学与技术的大部分研究工作都是围绕程序设计进行的，新型计算机本身（主要指计算机硬件核心的芯片）的设计也归结为使用高级硬件描述语言的“程序设计”，所以也可以说整个计算机产业（硬件和软件的研究、设计和生产）就是在进行程序的设计与开发。因此，计算机专家和专业人员的培养和训练的最重要任务就是让学生掌握程序设计及其



相关理论的研究和开发能力。

所谓程序，就是要计算机完成某一任务所规定的一系列动作或步骤。当然，计算机程序不是简单地将几条或几十条命令放在一起，现代的计算机每秒钟可以执行成千上万条指令（目前，最快的计算机的峰值速度可达每秒钟处理 35 万亿条指令，现在人们正在研制千万亿次的 P 级高性能计算机）。因此，计算机程序规模很大，内容也十分复杂。为计算机编程是一种非常复杂且具有挑战性的工作。也可以说，自计算机问世的半个世纪以来，人们都是在研究、设计各种各样的程序，使计算机完成各种各样的任务。

1946 年美国的 Mauchly 和 Eckert 研制的第一台电脑 ENIAC 上应用的程序是用来计算弹道函数的。微软公司开发的 Windows 系统是一种用来管理计算机资源的图形界面操作系统，它也是一个大规模的程序。这个程序已为开发商创造了数百亿美元的财富。1997 年 5 月份，另一个计算机程序的开发也引起了新闻界的注意，这就是击败了国际象棋世界冠军卡斯帕罗夫的计算机程序“Deeper Blue”（“超蓝”）。这是一个能够下棋的程序，它以 4:2 击败了当代最具实力的国际象棋大师。

因此，程序设计是一种工作量大，极其困难、复杂而又富有魅力和创造性的工作。这样一种职业，每年都吸引着数以十万计的优秀人才投入其中，他们的智慧、想像力和创造性，推动了计算机产业和计算学科的迅速发展。

### 1.1.2 程序设计语言

程序设计的任务就是用计算机可以识别的语言即程序设计语言编写程序，然后交给它去执行。

#### 1. 计算机指令系统

严格地说，一台“纯粹”的计算机（或称“裸机”）并没有特别高的性能，必须通过“程序”完成各种任务。

（1）计算机本身只会完成几十种（或一百多种）不同的简单“动作”，例如把内存中某地址的数据送到某寄存器，把某地址的两个数相加送到某寄存器，判断某个值是否为 0 等。

（2）计算机设计者把计算机可以完成的动作编辑成一个指令表，每种动作赋予一个二进制编码，并为机器的每种动作设计一种通用的格式：由指令码和内存地址组成的指令。一条指令就是一个固定长度的由指令码和地址码组成的二进制位串，这就是计算机惟一可以读懂的语言，一般称作机器语言。

（3）程序员把要计算机完成的任务分解为一系列其指令表（或指令系统）包括的“动作”，以指令序列的形式写出来，这就是机器语言程序设计。

#### 2. 低级编程语言

1949 年，普林斯顿大学的冯·诺依曼（Von Neumann）研制的 EDVAC 计算机，首次把