



普通高等教育“十一五”国家级规划教材

C++程序设计

(第2版)

刘璟 编著



高等教育出版社
HIGHER EDUCATION PRESS

普通高等教育“十一五”国家级规划教材

C++程序设计

C++ Chengxu Sheji

(第2版)

刘 璟 编著



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是普通高等教育“十一五”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程序设计技术，通过大量编程实例（与 C++11 标准及 Visual C++2012 系统兼容）讲解如何使用 C++ 语言解决各种具体问题，特别注重提高读者的编程能力。

本书主要内容包括 C++ 的各种基本数据类型、导出类型及其运算；分支、循环程序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方法；模板与 I/O 流的使用以及异常处理等。在附录中介绍了两种最常用的集成开发环境（IDE）和标准模板库（STL）的使用方法。

本书内容严谨，注重实用，深入浅出，简明生动，适合作为高等学校一般理工科（含计算机）专业程序设计课程的教材，也可供自学 C++ 程序设计的读者使用。

图书在版编目（CIP）数据

C++ 程序设计/刘璟编著. --2 版. --北京：高等教育出版社，2013.2

普通高等教育“十一五”国家级规划教材

ISBN 978-7-04-035456-0

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

中国版本图书馆 CIP 数据核字（2012）第 303586 号

策划编辑 时 阳
插图绘制 邓 超

责任编辑 武林晓
责任校对 窦丽娜

封面设计 于文燕
责任印制 尤 静

版式设计 范晓红

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 100120
印 刷 北京四季青印刷厂
开 本 787mm×960mm 1/16
印 张 29.5
字 数 530 千字
购书热线 010-58581118
咨询电话 400-810-0598

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>
版 次 2004 年 9 月第 1 版
2013 年 2 月第 2 版
印 次 2013 年 2 月第 1 次印刷
定 价 39.70 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 35456-00

前言

IT 产业的高速发展使得程序设计课程成为高等学校计算机及其相关专业（例如电子、通信、自动化、数学、物理、生物工程等）的一门最重要的课程，能够熟练编程的理工科学生往往最受用人单位的欢迎。

目前，国内外高等学校的多数程序设计课程讲授 C++ 语言，可选择的教材（笔者也曾编写过几本）有很多，但令人满意的教材却极少，其根本原因是：教学对象是大学一二年级的初学者，他们几乎没有受过编程思想的训练，而 C++ 语言较 C 语言规模大，不容易理解，在有限的课时内学会编程显然是一件难事。为了攻克这一难关，首先需要读者全力以赴。另一方面，为了付出同样的时间和精力，达到掌握核心思想、学会高水平编程的目的，选用一本好的教材也是非常重要的。

多年 C++ 程序设计教学及应用和编写教材的经验，笔者为本教材的编写设定了几点原则。

（1）基本程序设计部分（第 1~6 章和第 10 章）是最重要的教学内容，必须讲清学好。这部分涵盖了 C 语言的全部有用内容，而且用新的机制取代了 C 语言中已被淘汰的诸如宏定义、I/O 函数、动态存储分配等内容。也就是说，学好第 1~6 章和第 10 章的效果要比学习全部 C 语言好得多。

（2）面向对象程序设计（OOP）是当今程序设计技术的主流，任何程序设计课程都必须讲 OOP，但一定要根据让学生学会 OOP 编程的需要，对内容有所取舍，减少难度。本书分 3 个层次：突出最基本的 OOP 概念、思想和编程方法（主要在第 7 章）；选择最有用的深入内容（第 8、9 章的派生、虚函数、模板），并力求内容简明（这些地方比较难读，但读者花些精力是值得的）；尽量删减与实用编程关系不大的内容，免得读者浪费精力。

（3）注意提高教材的可读性。对于与提高编程能力直接有关的重点内容，特别讲清讲透，精心选用程序实例和图表，保证学生真正弄懂概念，学会动手编程。

（4）为了培养高水平的编程人才，本书为那些未来的编程能手留下进一步学习的空间：各章中注释的内容，在标题后加“*”章节的内容，在章节号前加

“*”的内容，附录 D 和附录 E 的内容等。在把最基本的内容学好之后，本书还将伴随读者继续深入提高。

(5) 为教师的授课提供最大的灵活性。本书支持培养实用编程能力的教学，因此必须坚持讲授和上机实践相结合。另一方面，教师讲授的内容可以根据不同的课时数和学生的程度灵活确定，一个标准的范围是第 1~10 章减去加“*”内容；最小的范围是第 2~7 章和第 10 章减去加“*”内容；加“*”的章节不主张在课堂讲授。学生如果能够（在一两年后）读懂附录 E 的 OOP 程序实例，就达到最高境界了。

(6) 为了体现 C++语言的最新发展，除了保证所有编程实例与 C++11 标准（及 Visual C++2012 系统）兼容之处，本书还介绍了与核心语言有关的 C++11 新扩展。

本书的编写长期得到高等教育出版社的支持。南开大学计算机系王刚教授审阅了全文，提出了若干中肯的改进意见；计算机系刘晓光、周玉龙、王娟等教授也提供了许多宝贵的建议。在此一并向他们致以深挚的谢意。

另外，作者编写的几本程序设计教材得到了众多读者、同行的鼓励 and 爱护，希望他们继续关注本书，对书中的不足和不妥之处提出批评和建议，作者将不胜感激。与本书有关的材料和信息，可访问中国高校计算机课程网（<http://computer.cncourse.com>）。

刘 璟

2012 年 6 月

目 录

第 1 章 绪论	1
1.1 程序设计与程序设计语言	2
1.1.1 计算机与程序设计	2
1.1.2 程序设计语言, 从低级到高级	4
1.1.3 程序设计方法学的发展	7
1.1.4 程序设计的范型	9
1.1.5 程序设计技术的 4 个层次	11
1.2 C++语言概述	13
1.2.1 为什么选择 C++语言	14
1.2.2 C++语言简史	14
1.2.3 C++语言的特点	16
1.3 学习本书的指南	18
1.3.1 学习 C++语言程序设计的难点	18
1.3.2 一本好的 C++程序设计教程	20
1.3.3 本书的内容安排	21
思考题	23
第 2 章 C++语言初步	25
2.1 初识 C++程序	26
2.1.1 C++程序实例	26
2.1.2 输入/输出(I/O)语句	29
2.2 C++语言的基本符号	31
2.2.1 基本符号集	32
2.2.2 基本符号的 ASCII 编码	32
2.3 C++语言的词汇	34
2.3.1 关键字	34
2.3.2 标识符	35
2.3.3 字面常量	36
2.3.4 运算符	39

2.3.5 分隔符.....	39
2.4 C++程序的整体结构.....	39
2.4.1 主函数.....	40
2.4.2 预处理命令.....	41
2.4.3 C++程序的基本框架.....	46
2.5 运行 C++程序.....	50
2.5.1 编辑 C++程序.....	50
2.5.2 编译和链接过程.....	50
2.5.3 运行一个简单的实例.....	52
思考题.....	55
练习题.....	55
第3章 基本数据类型与基本运算	57
3.1 基本数据类型及其派生类型.....	58
3.1.1 数据类型的概念.....	58
3.1.2 基本类型.....	59
3.1.3 基本类型的派生类型.....	63
3.1.4 enum 类型.....	64
3.1.5 整数类型与浮点类型.....	65
3.2 说明语句.....	65
3.2.1 语句.....	65
3.2.2 变量和常量概念.....	66
3.2.3 常量说明.....	67
3.2.4 变量说明.....	67
3.2.5 名字空间.....	70
3.2.6 类型说明.....	71
3.3 基本运算.....	72
3.3.1 运算符和表达式.....	72
3.3.2 运算的分类.....	73
3.3.3 赋值运算.....	74
3.3.4 算术运算.....	74
3.3.5 关系运算.....	76
3.3.6 逻辑运算.....	76
3.3.7 位运算.....	78
3.3.8 其他运算.....	79
3.3.9 运算的优先级.....	82

3.3.10 混合运算与数据的类型转换	84
*3.3.11 关于 C++语言与实现系统的注释	85
思考题	86
练习题	87
第 4 章 程序的基本控制结构及导出数据类型	89
4.1 C++程序的基本控制结构	90
4.1.1 控制语句	90
4.1.2 复合语句和空语句	90
4.2 分支语句	92
4.2.1 两类分支语句	92
4.2.2 温度变换程序	95
4.2.3 简单的计算器程序	98
4.3 循环语句	100
4.3.1 3 种循环语句	100
4.3.2 求素数	103
4.3.3 计算常数 e 的值	104
4.4 无条件控制语句	106
4.5 数组类型	110
4.5.1 一维数组	111
4.5.2 多维数组	113
4.5.3 数组与字符串	114
4.6 结构类型	114
4.6.1 结构类型与结构变量说明	115
4.6.2 结构变量的引用和赋值	116
4.6.3 结构数组	116
4.7 C++程序实例	117
4.7.1 统计学生成绩	117
4.7.2 输出三角函数表	121
4.7.3 画一个四叶玫瑰线图形	122
4.7.4 Eratosthenes 筛法求素数	126
思考题	128
练习题	129
第 5 章 函数	137
5.1 三次方程求根程序的设计	138

5.2 函数的说明与使用	139
5.2.1 函数的说明	139
5.2.2 函数的调用	142
5.2.3 函数的返回	143
5.2.4 函数的参数	144
5.2.5 值调用与引用调用	145
5.2.6 内联函数	149
*5.2.7 关于主函数参数的说明	150
5.3 函数的嵌套与递归	152
5.3.1 函数的嵌套	152
5.3.2 函数的递归	154
5.4 函数与运算符的重载	157
5.4.1 函数重载	157
5.4.2 运算符重载	158
5.5 函数与 C++ 程序结构	160
5.5.1 库函数的使用	160
5.5.2 SP 框架结构与自顶向下程序设计	161
5.5.3 函数间的数据传递	164
5.5.4 变量与函数的作用域	166
5.6 程序实例	169
5.6.1 “三色冰激凌” 程序	169
5.6.2 Hanoi 塔问题	171
思考题	173
练习题	174
第 6 章 指针和引用类型及动态内存分配	183
6.1 指针类型	184
6.1.1 指针类型变量的说明	184
6.1.2 指针运算符	184
6.1.3 指针变量的运算	185
6.1.4 指针变量的应用	189
6.1.5 关于指针的进一步话题	191
6.1.6 指针与动态内存分配	198
6.2 引用类型	200
6.3 程序实例	203

6.3.1 按人名字典序排列电话簿·····	203
6.3.2 构建人员档案链表·····	205
思考题·····	209
练习题·····	210
第 7 章 类与对象·····	219
7.1 类和对象的说明·····	220
7.1.1 类的说明·····	220
7.1.2 对象的说明·····	221
7.1.3 this 指针·····	224
7.1.4 作为类的结构·····	224
7.2 构造函数与析构函数·····	225
7.2.1 对象初始化·····	225
7.2.2 构造函数与析构函数简介·····	226
7.2.3 构造与析构函数的简单实例·····	228
7.3 类与对象的其他特征·····	229
7.3.1 类就是一个（用户定义的）类型·····	229
7.3.2 类的静态成员·····	233
7.3.3 类的常量成员·····	234
7.4 友元与类之间的关系·····	236
7.4.1 友元函数和友元类·····	236
7.4.2 类与类之间的关系·····	239
7.5 运算符重载用于用户定义类型·····	241
7.5.1 设计一个普通的 Set 类型·····	241
7.5.2 利用运算符重载改进 Set 类型·····	246
*7.6 程序实例：利用 Stack 类解迷宫问题·····	250
7.6.1 Stack 类也是一个用户定义类型·····	250
7.6.2 利用 Stack 类解迷宫问题的思路·····	251
7.6.3 采用回溯技术设计的迷宫程序·····	252
思考题·····	256
练习题·····	257
第 8 章 继承与派生·····	263
8.1 派生与继承·····	264
8.1.1 派生类说明·····	264
8.1.2 派生类的构造函数和析构函数·····	265
8.1.3 一个实例：公司雇员档案管理·····	268

8.1.4 有关成员存取权限问题的记注	271
8.2 其他特征的继承关系	273
8.2.1 友元关系以及静态成员的继承	273
8.2.2 基类对象和派生类对象间的赋值	273
8.3 派生关系中的二义性问题	275
8.4 多态性与虚函数	278
8.4.1 函数重载与静态联编	279
8.4.2 函数超载、虚函数及动态联编	279
8.4.3 纯虚函数与抽象基类	283
8.5 两个程序实例	284
8.5.1 计算函数的定积分	284
8.5.2 利用图元类画图的程序	288
思考题	293
练习题	293
第 9 章 模板	299
9.1 函数模板	300
9.1.1 函数模板的概念及说明	300
9.1.2 函数模板应用举例	302
9.2 类模板	305
9.2.1 类模板的概念与说明	305
9.2.2 一个队列类模板	307
9.2.3 带有类型参数和普通参数的类模板	310
*9.3 关于类模板的几点说明	312
9.3.1 静态成员及友元	312
9.3.2 类模板的类型参数的错误检测	312
9.3.3 标准模板库 STL	314
9.4 程序实例：链表类模板的设计	315
思考题	318
练习题	319
第 10 章 输入输出 (I/O) 流	321
10.1 文件、流及 C++ 的流类库	322
10.1.1 流类库的优点	322
10.1.2 文件与流的概念	323
10.1.3 C++ 的流类系统	324

10.2 对标准设备的 I/O 操作.....	325
10.2.1 I/O 函数.....	325
10.2.2 对用户定义类型进行运算符重载.....	328
10.3 I/O 的格式控制.....	330
10.3.1 类 ios 中用于格式控制的成员函数.....	330
10.3.2 格式控制符.....	336
*10.3.3 用户定义格式控制符.....	338
10.4 磁盘文件的输入输出.....	340
10.4.1 文件的打开与关闭.....	341
10.4.2 使用插入与提取运算符对磁盘文件进行读写.....	343
10.4.3 使用类成员函数对文件流进行操作.....	344
10.5 text 文件与 binary 文件.....	350
10.5.1 按用户设置的文件形式进行读写.....	350
10.5.2 对数据文件进行随机访问.....	353
10.6 其他输入输出控制函数.....	357
10.6.1 I/O 操作状态控制.....	357
*10.6.2 其他 I/O 控制.....	359
思考题.....	360
练习题.....	361
*第 11 章 异常处理.....	365
11.1 异常处理的基本思想以及 C++ 实现.....	366
11.1.1 异常和异常处理.....	366
11.1.2 C++ 的 try-throw-catch 异常处理机制.....	367
11.1.3 try-throw-catch 异常处理机制的必要性.....	370
*11.2 异常处理的进一步讨论.....	372
思考题.....	372
练习题.....	373
附录 A Visual C++ 2008 IDE 概述.....	375
附录 B Visual C++ 6.0 编程环境简介.....	391
附录 C 标准模板库.....	405
附录 D 标准函数.....	417
附录 E 一个 OOP 程序设计实例——楼宇电梯系统仿真程序的 OOP 设计过程.....	423
参考文献.....	457

第1章

绪论

本章要点

- ★ 了解程序设计在整个计算机科学与技术中的核心位置,计算机在各行各业的广泛应用都是通过程序设计实现的。
- ★ 了解程序就是人告诉计算机如何工作的命令序列,程序设计语言就是人与计算机之间交流的媒介。
- ★ 了解程序设计语言的发展历史,低级语言和高级语言,命令型程序设计和面向对象程序设计,程序设计方法学和计算机算法。
- ★ C++语言是目前最重要的程序设计语言,它与 BASIC、Pascal、C、Java、C#等语言的关系。
- ★ 了解 C++语言的产生、特点和发展历史,学习 C++语言的难点,如何利用本教材学习程序设计的概念、方法和技巧。

关键词

低级语言 (low level language), 高级语言 (high level language), 机器语言 (machine language), 汇编语言 (assembly language)

结构程序设计 (Structured Programming), 面向对象程序设计 (Object-Oriented Programming)

程序设计工具与环境 (programming tool and environment), 计算机算法设计与分析 (computer algorithm design and analysis)

我们把本章作为全书的开篇,让读者在学习本书的详细内容之前,首先对其相关领域有一个从大到小,由远及近的“鸟瞰”:

计算机——程序设计——程序设计语言——C++语言——C++语言教材——本书。认真阅读这一章,进入计算机学者的境界,有利于面对不容易理解和掌握的 C++程序设计原理和方法,较快地入门,深入地思考。当你能够自由地编程,并领悟其中之奥妙时,再回忆绪论的内容,一定会有更为深入的体会。

1.1 程序设计与程序设计语言

电子计算机的诞生是发生在 20 世纪中叶的一件大事，在其后的六十几年间，计算机对于人类已经远不仅仅是一种计算工具，与其相关的计算机技术已从一棵幼苗成长为枝繁叶茂的参天大树，目前，计算机科学与技术已经是最重要的科学学科，信息技术（IT）产业已成为最大的现代产业。

1.1.1 计算机与程序设计

国产高性能计算机“天河一号”的系统峰值性能（计算速度）达到每秒 4 700 万亿次、持续性能为每秒 2 507 万亿次，这个数字在当时的世界 500 强中位列第一。与 1946 年 2 月 15 日在美国诞生的世界第一台电子计算机 ENIAC（Electronic Numerical Integrator And Computer）的计算速度相比，已经提高万亿倍，前者一秒钟的计算量大致相当于后者计算一百万年的计算量。然而，无论是早期的计算机，还是“天河一号”，都离不开程序设计。

注释：

2010 年 10 月 29 日诞生的“天河一号”（位于国家高性能计算天津中心）在当时的世界 500 强中超过美国，位列第一。不过，“天河一号”的领先地位仅保持了 8 个月。2011 年 6 月，日本的超级计算机“京”超过了“天河一号”，到了 2012 年 6 月，美国 IBM 研制的“红杉”又以每秒 163 24 万亿次的速度夺回了领先地位。然而，设计和组装更快的计算机固然重要，但为其编写足够多的程序“喂”饱它也是向编程者提出的挑战。

计算机也许是人类文明历史中最伟大的发明之一。人们总是能够把计算机与近代人类其他伟大发明，如飞机、电灯、汽车、电视机、手机通信等相比较，从中可以感到它的威力、影响力和壮美前景。

计算机与其他发明相比有两个明显的区别。

（1）人类的发明大都是对自己器官的延长或替代，别的发明都可归结为人的运动和感觉器官的延长或替代；而计算机则是人类大脑功能的延长或某种替代，所以它被称为“电脑”。

（2）人类的发明可以分别应用在各个不同的局部领域，计算机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，计算机的广泛应用还在发展之中，正在运行的 PC 的数量已经达到几十亿台（2004 年的统计是 7.72 亿台），其他专用计算机就更多了，手机、照

相机、空调、电梯、电视（机顶盒）等，都装有计算机。据说，一辆汽车的功能相当于包含了超过 30 台计算机。另一方面，几乎每个人每天都在享受着计算机的服务，一次上网就可能有几十甚至几百台服务器及交换机、路由器为你服务。

计算机正在不断延长或代替大脑的活动，正在越来越被广泛而深入地应用于各个领域，谁也不能指出一个“到顶”不再发展的时间，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。例如，仅为手机编程就需要数以万计的程序员。

没有程序和程序设计，计算机可以说是一堆废物，也可以换一个说法：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，它的诞生又导致人们投入数十倍、数百倍的精力和智慧用于程序设计和软件开发，从而又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，特别是发展到今天的网络应用时代，不仅网络系统离不开程序设计，就是新型计算机本身（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”，所以也可以说整个信息技术产业（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。

计算机专家和专业人员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。

其实，把计算机比做电视机或电冰箱并不恰当，计算机更像是人所驯养的一匹马，它能够按照人的指令去完成各种任务。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步骤。计算机好像是听从人命令的仆人，严格地按照程序规定的步骤完成任务。当然，计算机程序不是简单的几条或几十条命令放在一起，现代的计算机每毫秒（1/1 000s）可以执行成千上万条指令。因此，计算机程序规模很大，内容十分复杂。为计算机编程是一种非常复杂，具有挑战性的工作。也可以说，自计算机问世以来，人们都是在研究设计各种各样的程序，使计算机能够完成各种各样的任务。

当年美国的 ENIAC 是用来计算火炮的弹道函数的，它需要数学家和程序员为其编写计算程序。

微软公司开发的 Windows 系列是一种用来管理计算机资源的图形界面操

作系统，它也是一种大规模的程序。现在，Windows 系列软件运行在数以亿计的计算机上，已为比尔·盖茨创造了数百亿美元的财富。

目前在全国以至全世界，几乎所有的银行、医院、能源电力企业、通信、交通、商业、国防、安全等部门都是在计算机软件系统的管理控制之下。

用电脑代替人脑已经可以做许多事情，其“智能”性越来越高。早在 1997 年，一个著名的计算机程序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（深蓝）”。近几年，国内开发的中国象棋弈棋软件“浪潮天梭”和国际象棋弈棋软件“紫光之星”都达到了可以击败国际特级大师的水平。2011 年 IBM 公司又推出了取名“Watson（沃森——纪念 IBM 的创始人托马斯·沃森）”的新的挑战性的智能系统，它可以参加《Jeopardy（危险边缘）》（一种在美国十分流行的电视智力竞赛），与水平最高的人（世界冠军）进行比赛，获得优胜。人们一方面用计算机代替自己动脑做事，同时又在动脑设计更新更强的软件。

注释：

“沃森”的计算能力是“深蓝”的 1 000 倍，每秒可完成 80 万亿次运算；存储空间达到 15TB，储存了大量的图书、新闻、文献资料。“沃森”的功能比“深蓝”更为复杂，其 DeepQA 软件可以在 3s 内听懂并理解问题（涉及文化、艺术、体育、历史、新闻等领域），搜索相关信息（类似谷歌、百度，但不联网），确定正确答案，并能根据比赛形势确定抢答策略。在 2011 年 2 月，“沃森”挑战两位“危险边缘”大赛高手 K.Jennings（曾获 74 次优胜）和 B.Rutter（曾获 \$325 万奖金），经过 3 天比赛，“沃森”以较大优势获胜（3 人成绩为 W-77147、J-24000、R-21600）。

如此看来，程序设计是一件工作量永无止境、极其困难复杂而又充满魅力和创造乐趣的工作。该职业每年都吸引着数以十万计的优秀人才投入其中，他们利用自己的全部智慧、想象力和创造性进行工作，换来了信息技术产业和计算学科日新月异的发展。

1.1.2 程序设计语言，从低级到高级

程序设计的任务就是用计算机懂得的语言即程序设计语言编写程序，然后交给它去执行。

1. 计算机指令系统与低级编程语言

严格地说，一台“纯粹”的计算机（或称“裸机”）并没有特别高超的本领，必须由人通过“程序”指挥它完成各种任务。

(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”，例如把内存某地址的数取到某寄存器；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代码，并为机器的每种动作设计一种通用的格式：由指令码和地址码组成的指令。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，这就是计算机唯一可以读懂的语言，一般称为机器语言。

例如，做一次加法 $TOTAL=PRICE+TAX$ ，程序可由以下 5 条指令组成：

```
0001010101101100    // (156C) 取 6C 内容送寄存器 5
0001011001101101    // (166D) 取 6D 内容送寄存器 6
0101000001010110    // (5056) 把两值相加，结果送寄存器 0
0011000001101110    // (306E) 把寄存器 0 中的结果送地址 6E
1100000000000000    // (C000) 停机
```

括号中为指令的十六进制数表示。虽然机器可以“看”懂这种“语言”，但对于程序员来说却很不方便。完成一个简单的计算公式也要编写几十条指令，编程工作枯燥烦琐，程序冗长难读，调试、修改、移植和维护都是难题。因此，早期的计算机应用并不广泛，编程的专业性也极强，人们逐渐感到用机器语言编程是计算机应用向前发展的瓶颈。

20 世纪 50 年代，人们想办法使得编程语言兼顾对话的双方——计算机与人。于是，一种“汇编系统”(assembler)程序问世了，这种程序的功能是把一种“汇编语言”翻译为“机器语言”。例如，上面的加法程序用汇编语言可以写为：

```
LD R5,PRICE
LD R6,TAX
ADDI R0,R5,R6
ST R0,TOTAL
HLT
```

汇编语言用符号代替二进制表示操作码和地址码，这样大大减少了编程工作的困难。因此，汇编语言编程很快取代了机器语言编程。到了 20 世纪 60 年代，机器语言编程已经比较罕见了。

后来以 FORTRAN 和 ALGOL 60 为代表的高级语言逐渐流行，汇编语言的应用从应用程序的开发转向了系统程序，如操作系统(Operating System, OS)、编译系统(Compiler)的开发。到了 20 世纪 70 年代，新一代的高级语言 Pascal 和 C 语言问世，又逐渐在系统程序的开发领域取代了汇编语言。

20 世纪 80 年代，计算机一些新的应用形式如单板机、单片机等的简单编