



“十二五”普通高等教育本科国家级规划教材

高等学校计算机基础教育规划教材

程序设计基础

——从问题到程序（第2版）

胡 明 王红梅 编著



清华大学出版社



“十二五”普通高等教育本科国家级规划教材

高等学校计算机基础教育规划教材

程序设计基础

——从问题到程序（第2版）

胡 明 王红梅 编著

清华大学出版社
北京

内 容 简 介

本书以 C 语言为工具,以程序设计过程为主线,结合实际问题,基于计算思维,通过“问题→想法→算法→程序”的问题求解过程,带领读者分析问题、构造算法、设计程序,在潜移默化中掌握程序设计语言的基本知识,掌握程序设计的一般过程和基本方法,提高计算思维能力以及应用程序设计语言解决实际问题的能力。

本书适用于程序设计的初学者,主要面向没有任何编程知识和编程经验的读者。本书遵循初学者的认知规律和知识基础,科学安排知识单元之间的拓扑关系,概念清晰,实例丰富,深入浅出,是程序设计初学者的理想教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计基础——从问题到程序/胡明,王红梅编著.--2版.--北京:清华大学出版社,2016

高等学校计算机基础教育规划教材

ISBN 978-7-302-41180-2

I. ①程… II. ①胡… ②王… 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(2015)第 184462 号

责任编辑:袁勤勇

封面设计:傅瑞学

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:20.75

字 数:518 千字

版 次:2011 年 1 月第 1 版 2016 年 1 月第 2 版

印 次:2016 年 1 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:064814-01

第 2 版前言

在 ACM/IEEE-CS 提交的 CC 2005 中,将计算机专业的基本学科能力归纳为计算思维能力、算法设计与分析能力、程序设计与实现能力和系统能力。对于非计算机专业的学生,具有一定的计算思维能力和程序设计能力,才能适应信息社会的知识结构和思维方式。本书遵循初学者的认知规律和知识基础,以 C 语言为工具,以程序设计过程为主线,通过“问题→想法→算法→程序”的问题求解过程,带领读者分析问题、构造算法、应用程序设计语言解决实际问题,在潜移默化中掌握程序设计的基本思想和一般方法,提高计算思维能力和程序设计能力。

本书在保持第 1 版基本体例不变的基础上,进行了如下修改:

① 重新规划知识单元,科学安排知识单元之间的拓扑关系,如图 1 所示。

② 针对 C 语言的结构不良性,根据知识单元的拓扑结构,重新规划了教材的三级目录。对于函数、指针、文件、程序的基本结构等具有一定难度的主题,采用增量的方式,先讲授基本内容,再讲授高级内容;对于数组、函数、指针等具有紧密耦合的知识单元,按照循序渐进的原则,重新安排了教学主线;增加了低级程序设计,将 C 语言的低级特性独立在一章。

③ 按照“问题→想法→算法→程序”的模式进行程序设计,这个过程正是计算思维的运用过程,如图 2 所示。本书所有问题都用伪代码给出了算法描述,并且所有程序均在 Dev C++ 5.8.3、Visual Studio 2010、CodeBlocks 12.11、Visual C++ 6.0 等环境下调试通过。读者可向作者或清华大学出版社索要程序源码。

④ 强调语法和语义之间的关系,站在内存的角度讲语义,通过写语句理解基本语法,通过写程序学会使用语句。在写语句和写程序的过程中,体会数据表示的作用,学会将数据从机外表示转化为机内表示,体会数据处理的作用,学会用算法描述求解问题的操作步骤。

本书由胡明和王红梅共同执笔,2012 级本科生赵守峰、刘兆庚和彭海涛等同学完成了代码调试与校对工作,参加本书编写的还有党源源、肖巍、姚庆安、孙旻等老师。由于作者的知识和写作水平有限,虽几经修改,仍难免有缺点和疏漏。欢迎同行专家和读者批评和指正,使本书在使用中不断改进、日臻完善。

作 者

2015 年 7 月

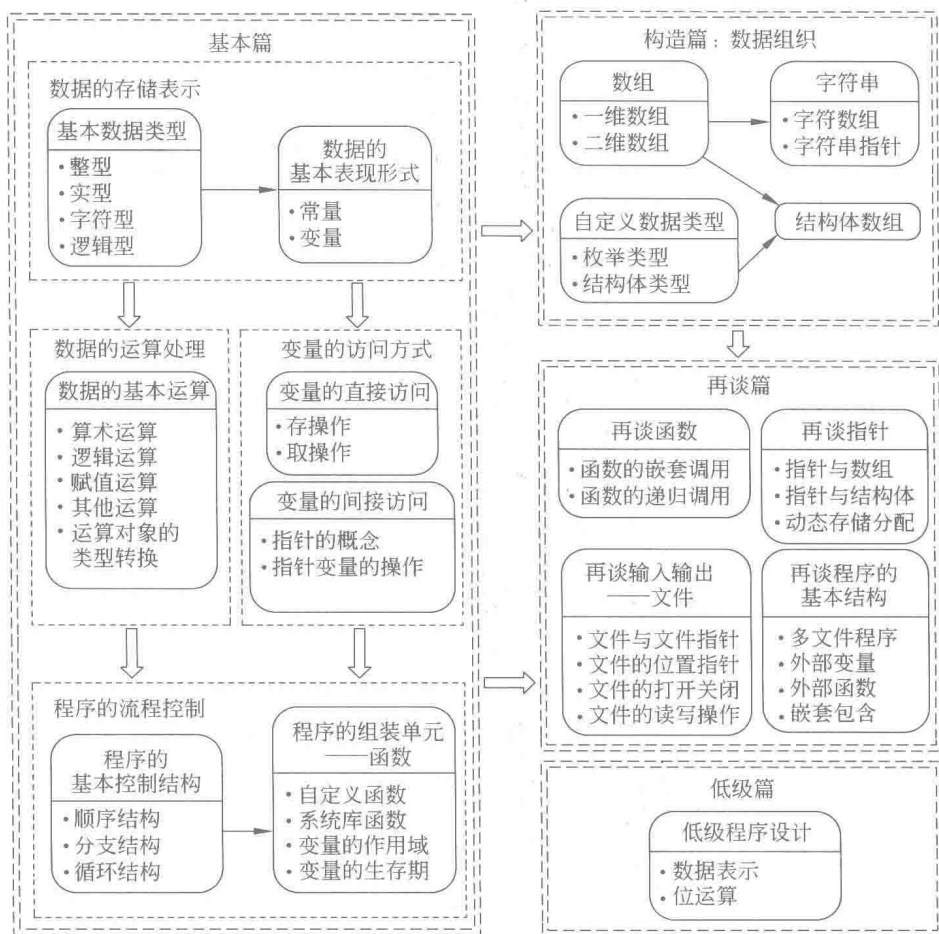


图1 知识单元及其拓扑结构

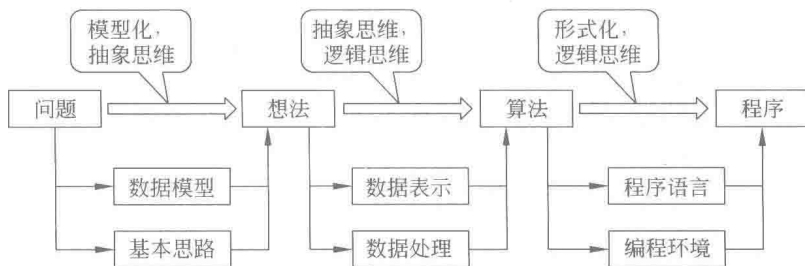


图2 程序设计的一般过程

目 录

第 1 章 绪论	1
1.1 程序、程序设计与程序设计语言	1
1.1.1 程序与程序设计	1
1.1.2 程序设计语言	2
1.2 程序的基本构成	4
1.2.1 程序示例	5
1.2.2 基本字符集	6
1.2.3 词法单位	6
1.2.4 语法单位	8
1.2.5 程序	8
1.3 程序的上机过程	12
1.3.1 编程环境	12
1.3.2 程序编辑	12
1.3.3 程序编译	13
1.3.4 程序连接	13
1.3.5 运行调试	13
1.4 程序风格	14
1.4.1 标识符的命名规则	14
1.4.2 注释	15
1.4.3 缩进	15
1.4.4 行文格式	16
1.5 问题求解与程序设计	17
1.5.1 程序设计的一般过程	17
1.5.2 程序设计的核心——算法	18
1.5.3 程序设计实例——鸡兔同笼问题	21
习题 1	22
第 2 章 数据的存储表示	25
引例 2.1 计算圆的面积	25

2.1	数据的存储	26
2.1.1	二进制	26
2.1.2	内存	27
2.2	基本数据类型	28
2.2.1	整型	29
2.2.2	实型	31
2.2.3	字符型	32
2.2.4	逻辑型	33
2.3	常量	34
2.3.1	字面常量	34
2.3.2	符号常量	36
2.4	变量	38
2.4.1	变量的概念	38
2.4.2	变量的定义和初始化	39
2.4.3	变量的赋值	40
2.5	程序设计实例	43
2.5.1	实例 1——温度转换	43
2.5.2	实例 2——计算本息和	43
	习题 2	44
第 3 章	数据的运算处理	47
3.1	算术运算	47
	引例 3.1 求平均值	47
3.1.1	算术运算	48
3.1.2	运算符的优先级和结合性	49
3.2	逻辑运算	50
	引例 3.2 判断闰年	50
3.2.1	关系运算	51
3.2.2	逻辑运算	51
3.3	赋值运算	52
3.3.1	赋值运算	52
3.3.2	自增/自减运算	53
3.4	其他运算	54
3.4.1	逗号运算	54
3.4.2	取长度运算	54
3.4.3	条件运算	55
3.5	运算对象的类型转换	55
3.5.1	自动转换	56

3.5.2 强制转换	57
3.6 程序设计实例	58
3.6.1 实例 1——通用产品代码 UPC	58
3.6.2 实例 2——疯狂赛车	59
习题 3	60
第 4 章 程序的基本控制结构	63
4.1 顺序结构	63
引例 4.1 整数的逆值	63
4.1.1 简单的顺序结构	64
4.1.2 复合语句	65
4.2 选择结构	65
引例 4.2 水仙花数	66
4.2.1 逻辑值控制的选择结构	67
4.2.2 算术值控制的选择结构	69
4.3 循环结构	71
引例 4.3 欧几里得算法	71
4.3.1 当型循环	72
4.3.2 直到型循环	72
4.3.3 计数型循环	73
4.3.4 循环结构的比较	74
4.3.5 循环结构的嵌套	77
4.4 其他控制语句	78
引例 4.4 素数判定	78
4.4.1 break 语句	79
4.4.2 continue 语句	80
4.5 程序设计实例	81
4.5.1 实例 1——将三个整数由小到大输出	81
4.5.2 实例 2——今年已经过去多久了	81
4.5.3 实例 3——百元买百鸡问题	83
4.5.4 实例 4——哥德巴赫猜想	84
习题 4	85
第 5 章 批量同类型数据的组织——数组	88
5.1 一维数组	88
引例 5.1 舞林大会	88
5.1.1 一维数组的定义和初始化	89
5.1.2 一维数组的操作	92

5.2	二维数组	94
	引例 5.2 哥尼斯堡七桥问题	94
5.2.1	二维数组的定义和初始化	95
5.2.2	二维数组的操作	97
5.3	程序设计实例	99
5.3.1	实例 1——折半查找	99
5.3.2	实例 2——合并有序数组	100
5.3.3	实例 3——对角线元素之和	101
5.3.4	实例 4——幻方问题	103
	习题 5	104
第 6 章	程序的组装单元——函数	107
6.1	用户定义的函数——自定义函数	107
	引例 6.1 欧几里得算法	107
6.1.1	函数定义	108
6.1.2	函数调用	110
6.1.3	函数声明	112
6.2	系统定义的函数——库函数	113
	引例 6.2 素数判定	114
6.2.1	头文件与文件包含	115
6.2.2	标准输入输出函数	116
6.2.3	随机函数	119
6.2.4	其他常用库函数	120
6.3	变量的作用域	121
	引例 6.3 鸡兔同笼问题	121
6.3.1	局部变量	122
6.3.2	全局变量	124
6.4	变量的生存期	125
	引例 6.4 字数统计	125
6.4.1	自动变量	126
6.4.2	静态变量	127
6.5	程序设计实例	128
6.5.1	实例 1——三角形的面积	128
6.5.2	实例 2——猜数游戏	129
	习题 6	131
第 7 章	变量的间接访问——指针	134
7.1	指针	134

引例 7.1 答疑教室	134
7.1.1 指针的概念	135
7.1.2 指针变量的定义和初始化	137
7.1.3 指针变量的操作	139
7.1.4 指针所指变量的操作	141
7.2 指针作为函数的参数	141
引例 7.2 鸡兔同笼问题	141
7.2.1 值传递方式——函数的输入	143
7.2.2 指针传递方式——函数的输出	144
7.2.3 指针传递方式——函数的输入输出	145
7.3 数组作为函数的参数	146
引例 7.3 顺序查找	146
7.3.1 一维数组作为函数的参数	147
7.3.2 二维数组作为函数的参数	149
7.4 程序设计实例	150
7.4.1 实例 1——三个整数由小到大输出	150
7.4.2 实例 2——哥德巴赫猜想	151
7.4.3 实例 3——求一元二次方程的根	152
7.4.4 实例 4——简单选择排序	154
习题 7	155
第 8 章 字符数据的组织——字符串	158
引例 8.1 恺撒加密	158
8.1 字符串变量的定义和初始化	159
8.1.1 字符数组	160
8.1.2 字符串指针	162
8.2 字符串的操作	162
8.2.1 输入输出操作	162
8.2.2 赋值操作	166
8.2.3 字符串的比较	167
8.2.4 常用字符串库函数	168
8.3 程序设计实例	168
8.3.1 实例 1——字数统计	168
8.3.2 实例 2——字符串匹配	169
习题 8	171
第 9 章 自定义数据类型	174
9.1 可枚举数据的组织——枚举类型	174

引例 9.1 行走机器人	174
9.1.1 枚举类型的定义	176
9.1.2 枚举变量的定义与初始化	176
9.1.3 枚举变量的操作	178
9.2 不同类型数据的组织——结构体类型	179
引例 9.2 统计入学成绩	179
9.2.1 结构体类型的定义	180
9.2.2 结构体变量的定义和初始化	181
9.2.3 结构体变量的操作	183
9.3 批量不同类型数据的组织——结构体数组	185
引例 9.3 统计入学成绩	186
9.3.1 结构体数组的定义和初始化	187
9.3.2 结构体数组的操作	189
9.4 为自定义数据类型定义别名	190
9.5 程序设计实例	191
9.5.1 实例 1——荷兰国旗问题	191
9.5.2 实例 2——最近对问题	193
习题 9	195
第 10 章 再谈函数	198
10.1 函数的嵌套调用	198
引例 10.1 字符串的循环左移	198
10.1.1 函数——封装的小程序	200
10.1.2 函数的嵌套调用	201
10.2 函数的递归调用	202
引例 10.2 求 $n!$	202
10.2.1 递归的定义	203
10.2.2 函数的递归调用	204
10.3 程序设计实例	206
10.3.1 实例 1——公共子序列	206
10.3.2 实例 2——弦截法求方程的根	208
10.3.3 实例 3——汉诺塔问题	210
10.3.4 实例 4——折半查找	211
习题 10	213
第 11 章 再谈指针	216
11.1 指针与数组	216
引例 11.1 判断回文串	216

11.1.1	用指针访问一维数组	217
11.1.2	用指针访问二维数组	219
11.1.3	指针数组	221
11.2	指针与结构体	222
引例 11.2	统计入学成绩	222
11.2.1	指向结构体的指针	223
11.2.2	结构体指针作为函数参数	224
11.3	动态存储分配	225
引例 11.3	发纸牌	226
11.3.1	申请和释放存储空间	227
11.3.2	动态数组	230
11.3.3	链表	232
11.4	程序设计实例	236
11.4.1	实例 1——求最大字符串	236
11.4.2	实例 2——进制转换	237
习题 11		239
第 12 章	再谈输入输出	241
引例 12.1	统计入学成绩	241
12.1	文件与文件指针	244
12.1.1	文本文件和二进制文件	244
12.1.2	文件缓冲区	245
12.1.3	文件指针	246
12.2	文件的当前位置指针	246
12.2.1	什么是文件的当前位置指针	246
12.2.2	跟踪文件的当前位置指针	247
12.2.3	定位文件的当前位置指针	248
12.3	文件的打开与关闭	249
12.3.1	文件的打开	249
12.3.2	文件的关闭	251
12.4	文件的读写操作	251
12.4.1	字符方式文件读写	251
12.4.2	字符串方式文件读写	252
12.4.3	格式化方式文件读写	254
12.4.4	二进制方式文件读写	255
12.5	程序设计实例	256
12.5.1	实例 1——文件复制	256
12.5.2	实例 2——注册与登录	258

习题 12	260
第 13 章 再谈程序的基本结构	262
13.1 多文件程序	262
13.1.1 多文件程序的构成	262
13.1.2 将源程序文件分解为多个程序文件模块	263
13.2 外部变量和外部函数	264
13.2.1 外部变量	264
13.2.2 外部函数	265
13.3 嵌套包含	267
13.3.1 条件编译	267
13.3.2 保护头文件	270
13.4 程序设计实例——石头、剪子、布游戏	271
习题 13	276
第 14 章 低级程序设计	277
引例 14.1 XOR 加密	277
14.1 数据表示	278
14.1.1 二进制数与八进制数和十六进制数之间的转换	278
14.1.2 位域	279
14.2 位运算	280
14.2.1 位逻辑运算	281
14.2.2 移位运算	282
14.2.3 补位原则	283
14.2.4 位运算的应用	283
14.3 程序设计实例	284
14.3.1 实例 1——快速欧几里得算法	284
14.3.2 实例 2——过滤特殊字符	285
习题 14	287
第 15 章 基本的算法设计技术	288
15.1 蛮力法	288
15.1.1 设计思想	288
15.1.2 程序设计实例——起泡排序	289
15.2 穷举法	291
15.2.1 设计思想	291
15.2.2 程序设计实例——假币问题	292
15.3 递推法	295



15.3.1	设计思想.....	295
15.3.2	程序设计实例——捕鱼知多少	296
15.4	分治法	297
15.4.1	设计思想.....	297
15.4.2	程序设计实例——数字旋转方阵	299
15.5	动态规划法	302
15.5.1	设计思想.....	302
15.5.2	程序设计实例——0/1 背包问题	303
15.6	贪心法	305
15.6.1	设计思想.....	305
15.6.2	程序设计实例——埃及分数	306
习题 15		308
附录 A	标准 ASCII 码	310
附录 B	运算符的优先级和结合性	311
附录 C	常用库函数	312
参考文献		317

绪 论

对于程序设计的初学者，用一种具体的程序设计语言作为工具来学习程序设计是比较通用的方法。所有程序设计语言的最终目的都是控制计算机按照人们的意愿去工作，因此，各种各样的程序设计语言具有共同的知识基础：数据表示、运算处理、流程控制、数据传递等。无论选择哪种程序设计语言，无论编写什么程序，都会运用这些知识基础。只要理解了程序设计的基本概念，掌握了程序设计语言的基本规则，就能够触类旁通、举一反三。本书选择 C 程序设计语言作为工具来讲授程序设计的基本思想和一般方法，由于篇幅所限，本书仅讨论结构化程序设计。

需要强调的是，对于程序设计的初学者，学习程序设计的目的不仅是学习某种具体的程序设计语言，更重要的是学习程序设计的基本思想和一般方法；学习程序设计的目标不仅是掌握程序设计语言的语法规则，更重要的是在程序设计实践中逐步培养运用程序设计语言解决实际问题的能力。

1.1 程序、程序设计与程序设计语言

计算机是一个大容量、可以高速运转、但是没有思维的机器，计算机看起来聪明是因为它能够精确、快速地执行基本的算术运算和逻辑运算。用计算机求解问题，必须使用计算机能够识别的语言，告诉计算机需要做哪些事，按什么步骤去做，计算机才会执行这些指令为人们解决特定的实际问题。

1.1.1 程序与程序设计

程序是能够实现特定功能的指令序列的集合，这些指令序列描述了计算机求解某一问题的步骤。简言之，程序是告诉计算机“做什么”以及“如何做”的指令集合。

所谓**指令**，就是计算机可以识别的命令，一台计算机硬件系统能够识别的所有指令的集合，称为计算机的指令系统。一条指令只能完成一个最基本的动作，如一次加法或一次移位，但一系列指令的组合却能够完成复杂的功能，这正是计算机的奇妙和强大功能所在。

程序设计是给出解决特定问题的程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，给出用这种语言编写的程序。专业的程序设计人员称为程序员。

需要强调的是，程序设计不仅仅是编写程序，而是从要解决的问题开始，进行问题分析，设计好解决方案后，才能够编写出正确的程序，最后还要在计算机上运行程序才能获得问题的解。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。

👉程序作为一种逻辑实体，具有抽象性。计算机系统的运行是看不见的，由于很难获得感性认识，所以理性认识就很难建立起来。抽象和符号化是程序设计初学者必须跨越的一关。

1.1.2 程序设计语言

计算机不能理解和执行人类的自然语言，人要和计算机进行交流就必须使用计算机能够识别的语言，因此，需要一种能够准确表达问题的求解步骤，同时还能够被计算机识别的表达方法。程序设计语言是为了方便描述计算过程而人为设计的符号语言，是人与计算机之间进行信息交流的语言工具。

作为计算机的关键技术，程序设计语言一直经历着改进和变化，其根本的推动力是对抽象机制的更高要求，以及对程序设计思想的更好支持。换言之，就是把机器能够识别的语言提升到能够很好地模仿人类思考问题的形式，很好地表示人类的思维。

1. 第一代程序设计语言（first generation language, 1GL）——机器语言

在第一台电子计算机诞生后便产生了机器语言。机器语言是内置在计算机电路中的指令，由 0 和 1 组成，如图 1.1 所示。这种计算机能直接识别和执行的二进制指令称为机器指令。机器语言是面向机器的语言，用机器语言编写程序相当繁琐，程序生产率很低，质量难以保证并且程序不能通用。

```
01010000101001010001010010101110101
00001010010101010010100101000010100
10100010100101011101010000101001010
10100101001010000101001010001010010
101110101000010...
```

想象一下，如果不小心弄错了一个二进制位，该如何找出来？

图 1.1 机器语言程序示例

2. 第二代程序设计语言（second generation language, 2GL）——汇编语言

20 世纪 50 年代初出现了汇编语言，它使用助记符表示每条机器指令，例如 MOV 表示传送数据，ADD 表示加，SHL 表示将数据左移，还可以使用十进制数或十六进制数，如图 1.2 所示。显然，汇编程序与硬件密切相关，因此程序也不能通用。

由于程序最终在计算机上执行时采用的都是机器指令，因此，需要用一种翻译程序把汇编程序翻译成等价的机器指令，如图 1.3 所示。



图 1.2 汇编语言程序示例

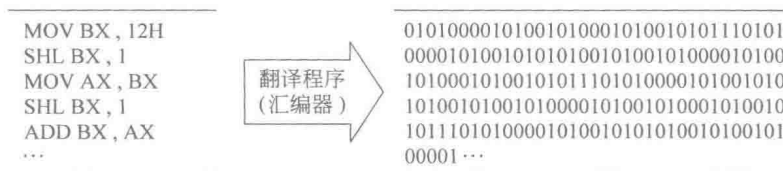


图 1.3 翻译程序将汇编程序转换成对应的机器指令

3. 第三代程序设计语言（third generation language, 3GL）——高级语言

高级程序设计语言（简称高级语言，相应地，机器语言和汇编语言称为低级语言，低级意味着程序员要从机器的层面上考虑问题）的指令形式类似于自然语言和数学语言，不仅容易学习，方便编程，也提高了程序的可读性。20 世纪 50 年代中期出现了第一个高级语言 FORTRAN，后来又相继出现了 COBOL、ALGOL、BASIC 等高级语言。目前，高级语言已形成一个庞大的家族，包括结构化程序设计语言、面向对象程序设计语言、可视化程序设计语言、网络程序设计语言等。

1968 年，荷兰计算机科学家 Edsgar W.Dijkstra 发表了论文《GOTO 语句的害处》，指出调试和修改程序的难度与程序中包含 GOTO 语句的数量成正比，从此，各种结构化程序设计理念逐渐确立起来。Pascal 语言是采用结构化程序设计规则制定的，BASIC 语言被升级为具有结构化的版本，此外，还出现了灵活且功能强大的 C 语言。

面向对象程序设计最早是在 20 世纪 70 年代提出的，其出发点和基本原则是尽可能地模拟现实世界中人类的思维进程，使程序设计的方法和过程尽可能地接近人类解决现实问题的方法和过程。随着面向对象程序设计方法和工具的成熟，从 20 世纪 90 年代开始，面向对象程序设计逐渐成为最流行的程序设计技术，Java、C++、C#等都是面向对象程序设计语言。

可视化程序设计是在面向对象程序设计基础上发展起来的，可视化程序设计语言把设计图形用户界面的复杂性封装起来，编程人员只需使用系统提供的工具在屏幕上画出各种图形对象，并设置这些图形对象的属性，系统就会自动产生界面代码，从而大大提高程序设计的效率。Visual Basic、Visual C++等都是可视化程序设计语言。

1989 年，Tim Berners-Lee 发明了 WWW，推动了计算机网络的发展，程序设计语言又呈现出网络化的发展趋势。网络程序设计是在网络环境下进行程序设计，包括服务器端程序设计和客户端程序设计，常用的服务器端程序设计语言有 JSP、PHP、ASP 和 .NET 等，常用的客户端程序设计语言有 JavaScript 和 VBScript 等。