



新编应用型系列技能丛书

省级优质资源共享课程主讲教材

C语言程序设计 及应用教程

谢膺白 ● 主 编

张文胜 王振辉 ● 副主编

- 转型院校程序设计的精品教材
- 技术技能人才培养的共享资源
- 解决工程技术问题的良师益友
- 软件研制开发求精的实用参考

清华大学出版社



新编应用型系列技能丛书

C 语言程序设计及应用教程

主 编 谢膺白

副主编 张文胜 王振辉

参 编 谢稷光 陈 宏 任 华

王艳君 张 敏 刘 辉

清华大学出版社

内 容 简 介

本书是为 600 多所改型新建本科院校、以 C 语言为程序设计入门课程的理工科各专业而编写的教材。

本书以突出实际应用、解决实际问题,提高动手能力,培养工程师、技术技能型高素质劳动者为目标,由浅入深、循序渐进地对 C 语言程序设计中的程序开发环境、数据类型、控制结构、数组、函数、指针、结构体、共用体、文件,进行全面介绍。对各章节的例题精心选取、精心设计,进而精心开发了一批涉及数学物理、工程力学、电工电磁、航天航海、人文历史、经济建设,乃至小学算术、古代算经等方面的例子,使整个教材充满了时代感、新鲜感、生活感和实际感,引人入胜。对激发学生学习兴趣,训练学生计算思维能力,培养学生创新意识,提高学生解决实际问题的动手能力有很大帮助。

为了方便教学和自学,本教材配有电子教案。另外,本书的所有例题与习题全部在 Visual C++6.0 集成开发环境下开发并且调试通过。

本书可作为培养转型工程师以及技术技能型、高素质、应用型人才的本科、高职高专院校各理工类专业的程序设计课程教学用书,同时也可作为从事计算机应用的科学技术人员的参考用书。

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

C 语言程序设计及应用教程/谢膺白主编. —北京:清华大学出版社,2014

(新编应用型系列技能丛书)

ISBN 978-7-302-37214-1

I. ①C… II. ①谢… III. ①C 语言-程序设计-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 152086 号

责任编辑:杜长清

封面设计:刘 超

版式设计:文森时代

责任校对:马军令

责任印制:何 芊

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

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.5 字 数:452 千字

版 次:2014 年 10 月第 1 版 印 次:2014 年 10 月第 1 次印刷

印 数:1~3800

定 价:35.00 元

编审委员会

主 任：陈绥阳

副 主 任：谢膺白 李宝敏 樊学东 贺亚茹 张文胜

委 员：（按姓氏笔画排序）

马军红	王元一	王红刚	王征风	王英强	王 栋
王振铎	王 娜	王艳君	王振辉	王雅静	石永生
刘 辉	任 华	任志宏	汤宏萍	陈 宏	杜晓春
李金良	李校红	李 继	李 梅	杨华勇	苏智华
张小木	张 龙	张首军	张 敏	张 伟	张娓娓
林 青	孟晓丽	范训礼	郑长风	宫 丽	侯亚玲
赵向梅	赵金龙	赵福祥	殷亚玲	唐 明	梁计锋
黄玉蕾	谢稷光	薛慧芳			

策划编辑：苏明芳

序

Preface

2012 年 12 月，清华大学出版社在西安组织了两次电子和计算机类的教材讨论会，着重讨论了应用型本科与高职高专的教材编写问题。本系列丛书即是针对应用型本科的一组教材。

为面向应用型本科的教师和学生提供一组电子和计算机类的教材，不仅是市场细分的要求，而且是应用型本科培养目标与培养模式的要求。在以学历文凭为目的的教育中，应试为其目标，其教学是以知识点为主，而不强调应用。高职高专的专业强调其社会属性，是面向工作岗位的，采用“校企合作、工学结合、顶岗实习”的培养模式，提倡情景教学与面向工作任务的工作过程式教学方法，其教材是技能导向的。应用型本科，一方面是有别于高职类型的普通本科，其专业既有学科属性，表现为知识的基础性、系统性、完整性和时序性，又有社会分工的社会属性；另一方面，“应用型”则表示该专业的人才培养是为地区经济建设与社会发展服务的。兼顾这两方面的要求，是本组教材应体现的基本特征。

专业的培养方案是规范培养过程以达到培养目标的基本文件，其中的教学计划与课程大纲（标准）则建立了教学的课程体系与内容体系。这一体系的建立是复杂而细致的工作，通常不是个人能力所及的。目前，市场上的教材过于强调教材自身的封闭性，造成教材内容的过度冗余与教材间内容的过度重复，往往也是这一原因造成了一本教材的局限性。参加本组教材编写的院校，在多年的专业实践中，为适应市场的需求，不断进行课程体系与内容体系的建设工作，同时进行内容的梳理，在本组教材的编写过程中采用集体讨论、集体编写的方法，有助于体现相应的建设成果。

本组教材中，有的课程是相关院校的精品课程、重点课程建设项目，在教学方法上有所探究，也有经验与教训。应当看到，精品课程、重点课程建设首先是针对教师的，是为教师提供一个可供示范的样本和教学资料。但教材是面向学生的，本组教材面向应用型本科的学生，这是其明确的定位。在这一定位上，有的教材在写作风格上追求“让学生能读懂”，经验说明，这是比较难做到的。事实上，在一本教材中，既有教师指导学生阅读的部分，又有学生自学的部分，这体现了教学过程中教师的主导作用，也是基于建构主义学习理论的。教师编写并出版的仅仅是文本，只有经过教师的使用和学生的阅读，才成为教材。

电子和计算机行业是科技进步很快的行业，反映到教材上是要兼顾基础性、工具性和现代性。在该组教材中，从基于 Windows 系统的程序架构到基于 Android 系统的应用程序设计，从单片机接口技术与应用实践到嵌入式系统，从语言类教材到基于数据库的 .NET 架构，既反映了基于 PC（Personal Computer，个人计算机）体系结构与互联网结构的应用软件系统设计，又反映了基于非 PC 体系结构的先进计算机系统与移动互联网的应用软件系

统设计；既反映了 Java 中的 J2EE 技术，又反映了微软的 C# 技术，以适合 B/S 结构而满足中小型企业信息化建设的市场需求，读者有较大选择空间。

值得重视的是，在教材的编写中，编者引入了“计算思维”（《计算思维》，卡内基梅隆大学计算机系主任周以真）的观点，即不将计算机科学局限于编写代码，而着重于计算思维的培养。同时，在写作过程中，在涵盖知识点（用知识结构图表述）的基础上，采用基于任务的讲述过程，重视四个层次的实验，即认知性实验、验证性实验、设计性实验和综合性实验，并通过综合实例，以培养学生的应用能力。另外，积极进行立体化教材：含大纲、教学用 PPT、习题、习题答案、模拟试卷、模拟试卷答案、实训指导书等。

在本丛书的成书过程中，编者参考了多本相关书籍，作为附录加以注明，同时又得到清华大学出版社的大力支持，尤其是编辑苏明芳在编辑出版等方面做了大量工作，在此一并感谢。

由于编者学识有限，书中难免挂一漏万，存在不妥之处，敬请读者斧正。

陈安阳

前言

Foreword

为贯彻落实《国家中长期教育改革和发展规划纲要（2010—2020年）》，2014年初，教育部推出了1999年大学扩招后“专升本”的600多所地方本科院校逐步转型做现代职业教育的改革计划。这一计划的实施，将一改高校“学术研究”的知识殿堂这种千篇一律的格局，取而代之的是促进新建地方本科院校向应用技术、职业教育类型的本科高校转变，形成“学术研究型”和“技术技能型”高校并举的人才培养“立交桥”。彻底改变目前不少院校因结构性问题带来的毕业生难以就业，学校办学步履艰难的窘况，为社会提供数量充足、素质较高的经济和社会发展第一线的劳动者。

然而，适用于改型院校所需要的教材目前却寥寥无几，特别是作为改型的本科院校理工类各专业的程序设计课程用书，多年来一直沿袭着学术型高校甚至是985、211院校的教材。这就要求在技术技能型高校中长期站在教学第一线的教师们，能总结教学经验教训和体会，自力更生，加强合作，尽快独立编写出适合于自己院校的教材。“C语言程序设计”课程是所有高校理工类各专业学生学习程序设计的一门入门课程，该课程的教学效果将直接影响学生后续相关课程的学习、计算机思维的形成、创新意识的培养、动手能力的提高，乃至他们的就业与职业生涯。

本书以培养技术技能型人才为宗旨，本着“讲清理论，突出应用，贴近实际，结合科技，提高能力，解决问题”的原则，突破了以往C语言程序设计教材存在的问题：理论上过于强调程序设计语言本身的系统完整性，举例上过于偏重生僻晦涩数学问题的解决，较好地解决了学生在以往学习C语言程序设计时普遍存在的“课难听懂、书难看明、题难做对、事难解决”的“四难”问题。在由浅入深、循序渐进地对C语言程序设计的理论问题进行简要介绍的基础上，精心选取、精心设计，进而精心开发了一批涉及数学物理、工程力学、电工电磁、航天航海、人文历史、经济建设、社会进步，乃至小学算术、古代算经等方面的例子，使整个教材充满了时代感、新鲜感、生活感和实际感，引人入胜。对激发学生学习兴趣，训练学生计算思维能力，培养学生创新创业意识，提高学生动手解决实际问题的能力有很大帮助。

本教材由谢膺白教授任主编。他曾主持过两个国家级重点科技攻关项目中的子项目：X国两个油藏数值模拟软件的引进测试、消化应用、移植改进，又长期从事程序设计教学与软件研发工作，具有丰富的工程实践背景和教学经验。目前正主持着“软件外包服务人才培养试验研究”、“软件工程专业综合改革试点”两个陕西省省级教育改革项目。同时，谢膺白教授是省级精品资源共享课程“C语言程序设计”的负责人，本书即为该课程的配套主讲教材。

本教材由王振辉、张文胜任副主编，参与编写的人员还有：谢稷光、王艳君、张敏、

任华、陈宏、刘辉。副主编张文胜、王振辉对书稿认真地进行了第一次校对。主编谢膺白教授，对书稿进行了第二次校对和全面修改，增添了大量与服务经济建设和社会发展有关的实例，亲自开发了与工程实践密切相关的全部程序，使本书面目一新，特点突出，方向明确。刘辉在本教材的编写过程中，在组织协调、资料整理、例题筛选、程序调试等方面做了大量工作。

清华大学出版社的苏明芳编辑，清华大学出版社西北办事处的吴刚老师对本教材的编写、出版给予了大力帮助。

为了方便教学和自学，本教材配有电子教案。另外，本书的所有例题与习题全部在 Visual C++6.0 集成开发环境下研发。读者只要将教材中的例题及习题源代码复制到 Visual C++6.0 集成开发环境中，即可进行编译、连接、运行。

高校转型刚刚起步，适应转型的 C 程序设计教材难以寻觅，无从参考，我们成了摸着石头过河的探索者，但由于编者水平有限，加之时间紧急，书中如有错误和不当之处，敬请广大读者批评指正。

编 者

目 录

Contents

第 1 章 C 语言程序开发过程简述.....	1
1.1 程序设计概述.....	1
1.1.1 程序设计与程序设计语言.....	1
1.1.2 用户应用程序开发过程.....	3
1.2 C 程序的结构.....	6
1.3 C 程序开发环境.....	8
1.3.1 Visual C++6.0 集成开发环境简介.....	8
1.3.2 Visual C++平台下开发 C 程序的步骤.....	9
习题 1.....	11
第 2 章 顺序结构暨数据类型和表达式.....	13
2.1 顺序结构暨简单的格式输入/输出语句.....	13
2.1.1 顺序结构.....	13
2.1.2 最简单的格式输入语句.....	14
2.1.3 最简单的格式输出语句.....	14
2.2 C 语言数据类型分类.....	15
2.3 常量.....	16
2.3.1 常量.....	16
2.3.2 变量.....	18
2.4 整型变量.....	20
2.4.1 整型数的补码存储形式.....	20
2.4.2 整数的取值范围.....	20
2.4.3 C 语言中整型变量的取值范围.....	21
2.5 实型变量.....	21
2.5.1 实型数据的存储形式.....	22
2.5.2 实型变量.....	22
2.6 字符变量.....	23
2.6.1 字符的存储形式.....	23
2.6.2 字符变量.....	24
2.7 不同数据类型之间的相互转换.....	24
2.7.1 自动类型转换.....	25

2.7.2	强制类型转换	25
2.8	运算符和表达式	26
2.8.1	算术运算符和算术表达式	27
2.8.2	赋值运算符和赋值表达式	28
2.8.3	关系运算符和关系表达式	30
2.8.4	逻辑运算符和逻辑表达式	31
2.8.5	条件运算符和条件表达式	33
2.8.6	逗号运算符和逗号表达式	34
2.9	位运算	34
2.9.1	位运算符和位运算表达式	34
2.9.2	位运算详解	35
2.10	完整的输入/输出函数	39
2.10.1	字符的输入/输出函数	39
2.10.2	字符串的输入/输出函数	40
2.10.3	格式转换说明符	40
2.10.4	格式修饰符	43
2.11	顺序结构暨运算符表达式综合应用举例	45
习题 2		48
第 3 章	选择结构和循环结构	52
3.1	选择结构	52
3.1.1	选择结构概述	52
3.1.2	if 语句和单双路分支	53
3.1.3	多路分支结构	57
3.2	循环控制结构	61
3.2.1	循环结构概述	61
3.2.2	用 goto 语句实现当循环	61
3.2.3	while 当循环语句	62
3.2.4	do...while 当循环语句	63
3.2.5	for 循环语句	64
3.2.6	循环语句的嵌套	66
3.2.7	break 语句和 continue 语句	66
3.2.8	几种循环的比较	69
3.3	控制结构应用举例	69
习题 3		76
第 4 章	数组与字符串	82
4.1	一维数组	82
4.1.1	一维数组的声明、定义和初始化	82
4.1.2	一维数组元素的引用	84

4.1.3	一维数组的应用	84
4.2	二维数组	87
4.2.1	二维数组的声明、定义与初始化	87
4.2.2	二维数组及元素的引用	89
4.2.3	二维数组的应用	89
4.3	字符数组与字符串	92
4.3.1	字符数组	93
4.3.2	字符数组的引用	95
4.3.3	字符串的操作	96
4.3.4	字符串处理标准库函数	99
4.4	数组综合应用举例	102
习题 4		110
第 5 章	函数与带参数宏	115
5.1	函数的声明、定义与调用	115
5.1.1	函数的声明与定义	115
5.1.2	函数调用	116
5.1.3	函数参数传递	118
5.1.4	主函数 main() 中的参数	121
5.2	函数的嵌套调用与递归调用	122
5.2.1	函数的嵌套调用	122
5.2.2	函数的递归调用	123
5.3	内部函数和外部函数	124
5.3.1	内部函数	125
5.3.2	外部函数	125
5.4	带参数宏	126
5.5	库函数及预处理命令 #include	129
5.6	函数综合应用举例	130
习题 5		138
第 6 章	变量的作用域与存储类型	142
6.1	变量的作用域	142
6.1.1	局部作用域	142
6.1.2	全局作用域	143
6.2	局部变量与全局变量	143
6.2.1	局部变量	143
6.2.2	全局变量	145
6.3	全局变量作用域的扩展	146
6.4	变量的生命周期与存储类型	147
6.4.1	自动变量	148

6.4.2	外部变量	150
6.4.3	静态变量	151
6.4.4	寄存器变量	152
6.5	综合应用举例	152
6.5.1	局部变量综合应用举例	152
6.5.2	外部变量的应用举例	154
习题 6		156
第 7 章	指针	161
7.1	变量的地址和指针变量	161
7.1.1	变量的地址及变量存取方式	161
7.1.2	指针变量的声明和指针变量的基类型	163
7.1.3	指针变量的定义	164
7.2	指针运算	164
7.2.1	指针运算符	164
7.2.2	指针运算	165
7.3	指针变量作为函数的参数	167
7.4	数组的指针与指向数组的指针变量	169
7.4.1	指向一维数组的指针变量	169
7.4.2	指向多维数组的指针变量	172
7.5	数组名作为函数的参数	174
7.5.1	用指向一维数组的指针作函数的参数	174
7.5.2	用指向多维数组的指针作函数参数	177
7.6	字符串的指针和指向字符串的指针变量	179
7.6.1	字符串的访问形式	179
7.6.2	字符串指针作为函数参数	180
7.6.3	字符数组与字符串指针的区别	181
7.7	指向函数的指针	181
7.7.1	函数的指针	181
7.7.2	用指向函数的指针作函数参数	182
7.8	返回指针值的函数	183
7.9	指针数组与指向指针的指针	184
7.9.1	指针数组的概念	184
7.9.2	指向指针的指针	186
7.9.3	指针数组作 <code>main()</code> 函数的参数	187
7.10	指针的综合应用举例	188
习题 7		191
第 8 章	结构体和共用体	197
8.1	结构体类型	197

8.1.1	结构体类型	197
8.1.2	结构体类型的定义	198
8.2	结构体变量的声明与引用	200
8.2.1	结构体变量的声明	200
8.2.2	结构体变量的引用	201
8.3	结构体数组	204
8.3.1	结构体数组的声明	204
8.3.2	结构体数组的引用	204
8.4	结构体与指针	207
8.4.1	结构体指针变量	207
8.4.2	指向结构体数组的指针	208
8.4.3	结构体指针数组	208
8.5	共用体类型	209
8.5.1	共用体类型	209
8.5.2	共用体类型的定义	210
8.6	共用体变量的声明与引用	211
8.6.1	共用体变量的声明	211
8.6.2	共用体变量的引用	211
8.7	结构体和共用体作为函数的参数	213
8.7.1	结构体作为函数参数	213
8.7.2	共用体作为函数参数	215
8.8	链表的基本操作	216
8.8.1	链表概述	216
8.8.2	动态链表的建立	218
8.8.3	链表节点的查找	220
8.8.4	链表节点的插入	224
8.8.5	链表节点的删除	225
8.9	特殊的基本数据类型——枚举类型	227
8.9.1	枚举型的定义	227
8.9.2	枚举型变量的声明与引用	228
8.10	位段及其运算	229
8.10.1	位段的定义	230
8.10.2	位段变量的声明	230
8.10.3	位段变量的使用	231
8.11	结构体共用体枚举的综合应用	231
习题 8		236
第 9 章	文件	239
9.1	文件概述	239

9.1.1	ASCII 文件	239
9.1.2	二进制文件	240
9.1.3	文本文件与二进制文件的存取	240
9.2	文件的打开与关闭	241
9.2.1	文件打开函数 <code>fopen()</code>	242
9.2.2	文件关闭函数 <code>fclose()</code>	243
9.3	文件的顺序读写	244
9.3.1	字符读写函数	244
9.3.2	字符串读写函数	246
9.3.3	文件格式化读写函数	247
9.3.4	数据块读写函数	250
9.4	文件的定位与随机读写	252
9.5	文件操作应用举例	256
习题 9		261
参考文献		264
附录 A	常用字符与 ASCII 代码对照表	265
附录 B	C 语言中的关键字	266
附录 C	C 语言中的运算符和结合性	268
附录 D	C 语言常用语法提要	270
附录 E	Turbo C 语言常用库函数表	273

第 1 章

C 语言程序开发过程简述

【本章内容简介】

计算机文化基础知识告诉我们，现代电子计算机是冯·诺依曼结构的计算机。此类计算机有两个特点，一是采用二进制；二是存储程序和程序控制。计算机一定是在程序方式下运行的，也只有在程序方式下，其“快”的优点才会表现得淋漓尽致。因此程序设计始终是计算机应用永恒不变的主题之一。

1.1 程序设计概述

人类进入 21 世纪后，以数学为代表的逻辑思维、以物理学为代表的实证思维、以计算机科学为代表的计算思维，在人类科学实践活动中越来越表现出无穷的生命力。为了实现四个现代化，我国从幼儿开始，大力普及了计算机基础教育。计算机基础教育分为计算机文化基础、计算机应用基础、计算机技术基础三个不同层面的教育。

计算机文化基础是计算机操作，它对应于教育部推行的计算机等级考试（简称等级考试）的一级考试以及工信部推行的计算机软件水平考试（简称软考）的初级考试。它是现代社会衡量一个人文化素质高低的标准，也是这个人能否立足于现代社会的“门面”。目前我国，这一层面的工作，在中小学阶段已基本完成。

计算机应用基础，其主题主要为程序设计，因为只有程序才能使计算机的“本事”不断得到开发和扩充，它对应于等级考试的二级考试以及软考的中级考试。这需要有一定数学、物理、化学、外语等自然科学知识、社会科学基础的人才能学懂用好。能否熟练进行计算机编程则是检验一个人实践能力强弱的试金石。

计算机技术基础涉及的知识面很广，既有硬件方面的要求，又有软件方面的要求，对人的要求则更专业。它对应于等级考试的三级和四级以及软考的高级考试。这个层面的工作，大多在大学理工科有关专业的高年级实现。

1.1.1 程序设计与程序设计语言

众所周知，计算机命令就是指示计算机完成某项操作的符号序列。程序则是由人们编写的要求计算机实现某项任务的一系列计算机命令的有序集合。人们在编写计算机程序时，要使用程序设计语言来完成。

程序设计语言是人们事先约定的在全世界范围内通用的编程规定。从 1946 年世界上第

一台电子计算机面世到现在 60 多年的时间里, 程序设计语言先后经历了由低级语言到高级语言的 2 个阶段, 目前程序设计的第 3 个更高级的阶段, 即非过程化语言, 已崭露头角。

1. 低级语言阶段

计算机文化基础的知识告诉人们, 指令是用二进制所编写的、能被计算机硬件直接识别并执行的命令。指令集则指某类计算机拥有的全部指令的集合, 不同类型的计算机指令集不同。

所谓低级语言, 指编程所采用的语言, 与计算机的指令集密切相关, 即依赖于计算机。根据低级程序设计语言是否使用二进制进行编写, 又把它分为了机器语言和汇编语言两大类。

(1) 机器语言 (Machine Language)

机器语言指计算机诞生之初, 使用计算机指令所编写的计算机程序。这种语言的优点是代码质量高, 执行速度快。然而缺点非常明显, 表现在二进制代码使得编程、修改、阅读、交流都非常困难, 编程效率低下, 只有少数熟悉机器指令集的人才能够胜任编程工作。更要命的是这种语言所编写的程序不独立于机器, 因而不可移植。

(2) 汇编语言 (Assemble Language)

鉴于机器语言使用二进制编写计算机程序, 因而在编程、修改、阅读、交流等方面都非常困难, 20 世纪 50 年代初, 在机器语言的基础上发展了汇编语言。这是一种符号语言, 用助记符来表达机器指令。用它所编写的程序称为汇编源程序 (Source Program), 也称为汇编源代码 (Source Code)。汇编语言的出现, 使得程序的编写、修改、阅读、交流相对于机器语言简单便捷得多了, 但仍没有摆脱机器的束缚。汇编源程序必须经过专门的翻译程序 (称之为汇编程序, Assembler) 进行翻译, 才能成为机器所能识别的机器语言。

2. 高级语言阶段

随着计算机技术的发展, 计算机应用的普及, 产生了一种全新的程序设计语言。这种语言采用近似于人类所习惯的描述形式, 例如用类似于自然语言、数学表达式的形式来编写程序。此类全新的程序设计语言, 克服了低级语言的所有缺点, 它具有易书写、好维护、便交流, 开发效率高, 独立于机器便于移植的优点, 备受人们的青睐, 被称为高级语言。

当然像汇编语言编写的源程序一样, 高级语言所编写的源程序机器不可能直接识别并执行, 也必须经过翻译程序 (称之为编译程序, Compiler) 进行翻译, 才能成为二进制的目标码 (Object Code)。而对于目标码还要再利用连接程序 (Liker) 进行连接方可形成能被计算机执行的执行码 (Execution Code)。

自 1954 年人类诞生了第一个程序设计高级语言 FORTRAN (Formula Translation) 至今, 各种高级语言如雨后春笋般地层出不穷。从编程时人们的侧重点出发, 可以把高级语言分为面向过程的程序设计语言、面向对象的程序设计语言两大类, 而后者是以前者为基础发展起来的。

(1) 面向过程的程序设计语言 (Process-Oriented Programming Language)

面向过程的程序设计语言, 指人们在进行程序设计时, 将主要精力放在如何方便、快捷、正确地解决问题上, 即解决问题的过程上。这主要是由于当时的计算机速度慢、内存

小，人们不得不考虑如何能最大限度地提高程序运行效率，节约宝贵的内存资源。面向过程的程序设计语言有：最适合于进行科学计算的 FORTRAN 语言，最适合于初学者学习的 BASIC 语言，最适应于教学的 Pasical 语言，最适于写操作系统的 C 语言等。

(2) 面向对象的程序设计语言 (Object-Oriented Programming Language)

随着计算机技术的发展，面向对象的程序设计语言应运而生。人们在进行程序设计时，将主要精力放在研究客观事物（即对象）的属性、行为及事物间的关系上，而非集中在问题的具体求解过程上，它是以面向过程的程序设计为基础而发展起来的。这种程序设计语言更符合人们对客观世界的认知规律，因而效率更高，运行速度更快，解决的问题更为复杂。伴随着视窗界面的出现，这种程序设计语言又引进了可视化界面，使程序设计更加方便。目前比较流行的面向对象的程序设计语言有 Visual Basic、Visual C++、Java 等。

3. 非过程化语言 4GL (Forth Generation Language)

此类语言是目前程序设计的最高阶段。它不像过程语言一定要告诉计算机每一步应该“怎么做”，而只告诉计算机这个问题“要做什么”，至于如何做的问题则留给计算机自己去完成。因此它的一条语句相当于高级语言的多条语句，可能被编译成几十条机器指令。其代表是用于数据库系统的结构化查询语言 SQL (Structured Query Language)、数学软件 MATLAB 等。

1.1.2 用户应用程序开发过程

不管是用哪种高级语言进行程序设计，都要按下面的步骤进行。

1. 分析问题确定算法

算法是解决问题的方法和步骤，不同的问题有不同的算法。当从现实生活中获得了要解决的问题时，应对其进行认真细致的分析研究，从而确定出科学、合理、可行的算法。

例如，有一个非常简单的问题，输入任意三角形的三条边的边长，求它的面积。对于这个问题，从平面几何的知识可以知道，尽管计算三角形面积的方法很多，但对于已知三个边长的任意三角形，计算面积的最好算法是使用海伦公式求解。

2. 根据算法绘流程图

算法的描述，通常有自然语言、伪代码、流程图 3 种方式，其中流程图是最流行的方式。程序设计流程图，是人们根据事先约定的代表一定意义的几何图案、符号，将算法表示出来的图形，它是编写程序代码的依据。根据算法绘制程序流程图，是程序设计中非常重要的一个关节，特别是对于复杂的问题，绘制流程图非常重要。

结构化程序设计认为，复杂程序是由若干个比较简单的功能程序模块像搭积木一样搭起来的。功能模块是具有一定功能的独立的程序段，每一个模块都可以最终被细分为顺序、选择、循环 3 种基本结构。它们都可以用流程图表示出来。

最常用的流程图有传统流程图和 N-S 图两种。

(1) 传统流程图

传统流程图由如图 1-1 所示的一些主要的几何图案组成。