



新世纪高等学校计算机系列教材

# C语言 程序设计教程

◎ 陈宝贤 主编

◎ 莫裕清 杨小毛 副主编

新世纪高等学校计算机系列教材

# C 语言程序设计教程

陈宝贤 主编

莫裕清 杨小毛 副主编

人民邮电出版社

## 图书在版编目 (CIP) 数据

C 语言程序设计教程 / 陈宝贤主编. —北京: 人民邮电出版社, 2005.8  
(新世纪高等学校计算机系列教材)

ISBN 7-115-13535-5

I. C... II. 陈... 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 (2005) 第 076199 号

### 内 容 简 介

本书详细地介绍 C 语言基本知识, 以及程序设计的基本原理、结构化程序开发方法。全书共有 14 章, 内容包括: C 语言概述、程序设计基础知识、C 语言数据类型、数据运算、顺序结构程序、分支结构程序、循环结构程序、函数、预处理、数组与字符串处理、指针、结构体与共用体、文件、位运算。

书中大量的例题, 为读者提供了结构化程序设计以及数组、指针、结构体和数据文件的应用实例。

本书可作为大、专院校 C 语言程序设计的教材, 也可作为计算机程序设计初学者的自学参考书。

新世纪高等学校计算机系列教材

### C 语言程序设计教程

- 
- ◆ 主 编 陈宝贤  
副 主 编 莫裕清 杨小毛  
责任编辑 邹文波
  - ◆ 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>  
北京市大中印刷厂印刷  
新华书店总店北京发行所经销
  - ◆ 开本: 787×1092 1/16  
印张: 16  
字数: 381 千字 2005 年 8 月第 1 版  
印数: 4 001—6 000 册 2006 年 7 月北京第 2 次印刷

---

ISBN 7-115-13535-5/TP · 4727

定价: 26.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223



程序设计技术和程序设计语言是大学计算机基础系列课程中的重要组成部分，C 语言程序设计课程是一门重要的计算机基础必修课程。程序设计课程的主要任务是培养学生使用计算机的逻辑思维能力和基本的程序设计能力。通过 C 语言程序设计课程的学习，为学生今后进一步学习计算机有关课程，以及用计算机解决实际问题打下一个良好的基础。

参加本次教材编写的老师，近年来一直在任担“C 语言程序设计”课程的教学工作。大家有一个共识：“C 语言程序设计”课程的教学重点应是 C 语言基本知识，以及程序设计的基本原理、结构化程序设计方法，其内容主要包括程序设计基本概念、C 语言基本数据类型、运算符和表达式，格式化数据输入输出；控制结构语句、结构化程序开发方法；函数及程序模块化开发；数组、字符串、指针及其应用；结构体应用基础；数据文件的应用基础。

由于 C 语言程序设计课程教学课时有限，同时为了强化学生的逻辑思维及自主学习的能力，在教学过程中，特别要把学生在学习过程中经常碰到的难点问题讲解清楚；应避免例题讲解中涉及的比较复杂的数学问题，以致花费较多的教学时间。

在上述共识的基础上，经过多次的商讨，反复修改制订《C 语言程序设计教程》的编写大纲和《C 语言程序设计实验指导》的编写大纲。在此基础上，我们在得到了各参编院校和人民邮电出版社的大力支持下，组织编写了《C 语言程序设计教程》和《C 语言程序设计实验指导》。

《C 语言程序设计教程》由湖南大学计算机与通信学院陈宝贤主编，湖南信息职业技术学院莫裕清、长沙商贸旅游职业技术学院杨小毛任副主编。担任各章编写任务的老师有：陈宝贤（第 1、2、3 章），湖南大学计算机与通信学院陈燕（第 4 章），杨小毛（第 5、6、7 章），湖南信息职业技术学院杨洁（第 8 章），莫裕清（第 9、11、13 章），湖南信息职业技术学院甘清明（第 10 章），湖南商务职业技术学院戴开明（第 12 章），湖南大学教务处黄萍华（第 14 章），株洲市城市规划局胡阿辉（附录）。全书最后由陈宝贤编纂定稿。

《C 语言程序设计实验指导》由湖南大学计算机与通信学院陈宝贤主编，湖南信息职业技术学院莫裕清、长沙商贸旅游职业技术学院杨小毛任副主编。担任各部分编写任务的老师有：陈宝贤（第一部分

第1章、第2章实验一), 杨小毛(第一部分第2章实验二、三、附录), 湖南信息职业技术学院张瑛(第一部分第2章实验四), 湖南科技职业技术学院李禹(第一部分第2章实验五), 湖南信息职业技术学院甘清明(第一部分第2章实验六), 莫裕清(第一部分第2章实验七), 湖南商务职业技术学院戴开明(第一部分第2章实验八), 株洲生物工程学校易卫星(第一部分第2章实验九)。湖南大学计算机与通信学院杨小林(第一部分第3章)。第二部分中各章的习题答案, 由陈宝贤汇总、整理。

C 语言程序设计课程教学总学时数, 一般可以安排 80~90 学时, 其中课堂讲授学时 40~50 学时, 上机实践学时 40。(附课堂教学具体的课时分配表)

编 者

2005年6月

### 课堂教学具体的课时分配表

周	章、节	内 容	实 验
1	第 1 章	C 语言的发展简史和特点、认识 C 语言、Turbo C V2.0 的运行环境及基本操作	实验 1
2	第 2 章	算法与程序、结构化程序设计、结构化程序算法描述、面向对象设计简介	
3	第 3 章	数据类型、整型数据、实型数据、字符数据	
4	第 4 章	数据运算的基本概念、算术运算、赋值运算 关系运算、逻辑运算、自增自减运算、其他运算	实验 2
5	5.1~5.3	数据输入输出实现方法、字符输入输出函数 格式输入输出函数	实验 3
6	5.4、6.1	顺序结构程序、if 语句	
7	6.2、6.3	switch 语句、选择结构程序举例	
8	7.1~7.4	goto 语句、while 语句 do-while 语句、for 语句	实验 4
9	7.5~7.7	break 和 continue 语句、循环的嵌套 循环结构程序举例	
10	8.1 ~8.4	函数的分类、函数的定义 函数返回值、函数的调用	实验 5
11	8.5~8.6	变量的作用域与存储类别 函数的存储类别	
12	第 9 章	宏处理、文件包含、条件编译	
13	10.1~10.2	数组的概念、一维数组、程序举例	实验 6
14	10.3~10.5	多维数组、字符数组与字符串 字符串处理函数	
15	11.1~11.3	指针的概念、指针与变量、数组与指针	实验 7
16	11.4~11.5	字符串与指针、函数与指针	
17	11.6	指针数组和指向指针的指针	
18	12.1~12.5	结构体的定义、结构体变量的初始化和引用结构体数组、结构 指针、链表	实验 8
19	12.6~12.9	共用体、枚举类型、typedef 类型定义 结构体共用体综合实例分析	
20	13.1~13.5	文件概述、文件类型指针、文件的打开与关闭、文件的读写、 非缓冲文件	实验 9

# 目 录

<b>第 1 章 C 语言概述</b> .....	1
1.1 C 语言的发展简史和特点.....	1
1.1.1 计算机语言的发展过程.....	1
1.1.2 C 语言的发展简史.....	2
1.1.3 C 语言的特点.....	3
1.2 认识 C 语言.....	3
1.2.1 C 语言程序的结构.....	3
1.2.2 C 语言的标识符.....	5
1.2.3 C 语言语句.....	6
1.2.4 系统库函数.....	7
1.2.5 C 语言源程序的书写规则.....	8
1.3 Turbo C V2.0 的运行环境及基本操作.....	9
1.3.1 Turbo C V2.0 系统简介.....	9
1.3.2 TC 的启动与退出.....	10
1.3.3 TC 主屏幕.....	11
1.3.4 编辑一个 C 语言源程序.....	12
1.3.5 编译、连接单个源程序文件.....	13
1.3.6 运行一个 C 语言程序并查看运行结果.....	13
习题 1.....	14
<b>第 2 章 程序设计基础知识</b> .....	16
2.1 算法与程序.....	16
2.1.1 计算机解题过程.....	16
2.1.2 算法与程序.....	17
2.2 结构化程序设计.....	18
2.2.1 程序设计基本概念.....	19
2.2.2 结构化程序的 3 种基本结构.....	19
2.2.3 结构化程序设计方法.....	19
2.3 结构化程序算法描述.....	20
2.3.1 程序流程图.....	20
2.3.2 N-S 图.....	21
2.3.3 PAD 图.....	22
2.3.4 结构化程序算法描述实例.....	23

2.4 面向对象程序设计简介 .....	24
2.4.1 对象 .....	24
2.4.2 类 .....	24
习题 2 .....	25
<b>第 3 章 C 语言数据类型</b> .....	<b>26</b>
3.1 数据类型 .....	26
3.1.1 数据类型的种类 .....	26
3.1.2 常量与变量 .....	27
3.2 整型数据 .....	28
3.2.1 整型变量 .....	28
3.2.2 整型常量 .....	29
3.3 实型数据 .....	30
3.3.1 实型变量 .....	30
3.3.2 实型常量 .....	31
3.4 字符型数据 .....	32
3.4.1 字符常量 .....	32
3.4.2 字符变量 .....	33
3.4.3 字符串常量 .....	34
习题 3 .....	34
<b>第 4 章 数据运算</b> .....	<b>37</b>
4.1 数据运算的基本概念 .....	37
4.1.1 运算符 .....	37
4.1.2 表达式 .....	38
4.2 算术运算 .....	39
4.2.1 算术运算符 .....	39
4.2.2 算术表达式与运算规则 .....	39
4.3 赋值运算 .....	41
4.3.1 赋值运算符 .....	41
4.3.2 赋值表达式 .....	41
4.3.3 复合赋值运算 .....	43
4.4 关系运算 .....	44
4.4.1 关系运算符 .....	44
4.4.2 关系表达式 .....	44
4.5 逻辑运算 .....	45
4.5.1 逻辑运算符 .....	45
4.5.2 逻辑运算表达式 .....	45
4.6 自增自减运算 .....	47



4.6.1 自增、自减运算符 .....	47
4.6.2 自增、自减运算符的用法与运算规则 .....	47
4.7 其他运算 .....	48
4.7.1 条件运算 .....	48
4.7.2 逗号运算 .....	49
4.7.3 求字节数运算 .....	50
习题 4 .....	51
<b>第 5 章 顺序结构程序设计 .....</b>	<b>53</b>
5.1 顺序结构程序 .....	53
5.2 字符输入输出函数 .....	54
5.2.1 字符输入函数 <code>getchar</code> .....	54
5.2.2 字符输出函数 <code>putchar</code> .....	55
5.3 格式输入输出函数 .....	56
5.3.1 格式输出函数 <code>printf</code> .....	56
5.3.2 格式输入函数 <code>scanf</code> .....	58
5.4 顺序结构程序设计应用实例 .....	61
习题 5 .....	62
<b>第 6 章 选择结构程序设计 .....</b>	<b>65</b>
6.1 <code>if</code> 语句 .....	65
6.1.1 <code>if</code> 语句的 3 种形式 .....	65
6.1.2 <code>if</code> 语句与条件运算符 .....	69
6.1.3 <code>if</code> 语句的嵌套 .....	69
6.2 <code>switch</code> 语句 .....	72
6.2.1 <code>switch</code> 语句 .....	72
6.2.2 <code>switch</code> 语句中 <code>break</code> 语句的使用 .....	74
6.3 选择结构程序举例 .....	75
习题 6 .....	78
<b>第 7 章 循环结构程序设计 .....</b>	<b>82</b>
7.1 <code>goto</code> 语句 .....	82
7.2 <code>while</code> 语句 .....	83
7.3 <code>do-while</code> 语句 .....	84
7.4 <code>for</code> 语句 .....	86
7.5 <code>break</code> 和 <code>continue</code> 语句 .....	89
7.6 循环的嵌套 .....	91
7.7 循环结构程序举例 .....	92
习题 7 .....	94

<b>第8章 函数</b> .....	99
8.1 概述 .....	99
8.1.1 函数的分类 .....	99
8.1.2 函数的作用 .....	100
8.2 函数定义 .....	100
8.2.1 函数的定义 .....	100
8.2.2 函数的定义说明 .....	101
8.3 函数的返回值 .....	103
8.4 函数的调用 .....	104
8.4.1 函数的调用方式 .....	104
8.4.2 函数的形参与实参 .....	105
8.4.3 函数的嵌套调用 .....	107
8.4.4 函数的递归调用 .....	109
8.5 变量的作用域与存储类别 .....	111
8.5.1 变量的作用域 .....	111
8.5.2 变量存储类别 .....	114
8.6 函数的存储类别 .....	116
习题 8 .....	118
<b>第9章 预处理</b> .....	121
9.1 宏定义 .....	121
9.1.1 不带参数的宏定义 .....	121
9.1.2 带参数的宏定义 .....	123
9.2 文件包含 .....	125
9.3 条件编译 .....	126
习题 9 .....	128
<b>第10章 数组与字符串处理</b> .....	131
10.1 数组的概念 .....	131
10.2 一维数组 .....	132
10.2.1 一维数组的定义 .....	132
10.2.2 一维数组元素的引用 .....	133
10.2.3 一维数组的初始化 .....	133
10.2.4 一维数组程序举例 .....	134
10.3 多维数组 .....	136
10.3.1 二维数组的定义 .....	136
10.3.2 二维数组元素的引用 .....	136
10.3.3 二维数组的初始化 .....	137

10.3.4 二维数组程序举例 .....	137
10.4 字符数组与字符串 .....	138
10.4.1 字符数组 .....	138
10.4.2 字符串和字符串结束标志 .....	138
10.5 字符串处理函数 .....	139
10.5.1 几个最常用的字符串函数 .....	140
10.5.2 应用举例 .....	142
10.6 数组作为函数参数 .....	143
习题 10 .....	145
<b>第 11 章 指针</b> .....	<b>150</b>
11.1 指针的概念 .....	150
11.1.1 地址 .....	150
11.1.2 指针的类型和指针的值 .....	150
11.2 变量和指针 .....	151
11.2.1 变量的指针 .....	151
11.2.2 指向变量的指针变量 .....	151
11.2.3 指向变量的指针变量应用举例 .....	152
11.3 指针和数组 .....	154
11.3.1 数组中地址的概念 .....	154
11.3.2 指向一维数组的指针变量 .....	154
11.3.3 指向二维数组的指针变量 .....	157
11.4 字符串和指针变量 .....	160
11.5 函数和指针 .....	163
11.5.1 函数的指针和指向函数的指针变量 .....	163
11.5.2 指向函数的指针变量作为函数的参数 .....	164
11.5.3 返回指针值的函数 .....	165
11.6 指针数组和指向指针变量的指针 .....	166
11.6.1 指针数组 .....	166
11.6.2 指向指针的指针变量 .....	167
11.6.3 main 函数的形参 .....	168
习题 11 .....	169
<b>第 12 章 结构体与共用体</b> .....	<b>175</b>
12.1 概述 .....	175
12.2 结构类型与结构变量的定义 .....	175
12.2.1 结构类型的定义 .....	175
12.2.2 声明结构体类型变量 .....	176
12.3 结构体变量的初始化和引用 .....	178

12.4	结构体数组 .....	181
12.4.1	结构体数组的定义 .....	181
12.4.2	结构体数组的初始化 .....	181
12.4.3	结构体数组应用 .....	182
12.5	结构体指针 .....	185
12.5.1	结构体指针定义 .....	185
12.5.2	结构体指针引用 .....	185
12.5.3	用结构体变量的指针作函数参数 .....	187
12.6	结构体指针与链表 .....	189
12.6.1	链表概述 .....	189
12.6.2	单链表的建立 .....	190
12.6.3	用于动态存储分配的函数 .....	191
12.6.4	动态申请结点, 建立单链表 .....	192
12.6.5	链表的删除 .....	193
12.6.6	链表的插入 .....	194
12.6.7	单链表的应用实例 .....	195
12.7	共用体 .....	198
12.7.1	共用体的定义 .....	198
12.7.2	共用体变量的引用 .....	199
12.7.3	共用体类型数据的特点 .....	200
12.8	枚举类型 .....	201
12.8.1	枚举型变量的定义和引用 .....	202
12.8.2	枚举类型变量的应用实例 .....	203
12.9	typedef 类型定义 .....	204
习题 12	.....	205
<b>第 13 章</b>	<b>文件 .....</b>	<b>208</b>
13.1	文件及其分类 .....	208
13.2	文件指针 .....	209
13.3	文件的打开和关闭 .....	210
13.3.1	文件的打开 .....	210
13.3.2	文件的关闭 .....	211
13.4	文件的读写 .....	212
13.4.1	字符读写函数 fgetc 和 fputc .....	212
13.4.2	字符串读写函数 fgets 和 fputs .....	214
13.4.3	数据块读写函数 fread 和 fwrite .....	215
13.4.4	格式化读写函数 fscanf 和 fprintf .....	217
13.4.5	putw 和 getw 函数 .....	219
13.5	文件的随机读写和出错检测 .....	219

13.5.1 文件的定位 .....	219
13.5.2 文件检测函数 .....	221
习题 13 .....	221
<b>第 14 章 位运算</b> .....	<b>224</b>
14.1 位逻辑运算 .....	224
14.1.1 按位与运算 .....	224
14.1.2 按位或运算 .....	225
14.1.3 按位异或运算 .....	226
14.1.4 按位取反运算 .....	227
14.2 位移位运算 .....	228
14.2.1 按位左移运算 .....	228
14.2.2 按位右移运算 .....	228
14.3 复合位赋值运算 .....	230
14.3.1 复合位赋值运算符 .....	230
14.3.2 复合位赋值运算规则 .....	231
14.4 位段 .....	231
14.4.1 位段的概念与定义 .....	231
14.4.2 位段的引用 .....	232
习题 14 .....	233
<b>附录 1 常用字符与 ASCII 码对照表</b> .....	<b>236</b>
<b>附录 2 编译错误信息</b> .....	<b>238</b>
<b>参考文献</b> .....	<b>242</b>

# 第1章 C语言概述

程序设计语言是计算机软件中最重要的软工具，是为描述计算过程而设计的一种具有语法规义描述的特定符号集合。

C语言是一种结构化程序设计语言。由于具有许多类似汇编语言的特征，有先进的程序控制结构，有丰富的数据类型和运算（包括位运算），使得C语言在系统软件开发，在事务处理、自动控制、科学研究等各个领域的应用软件开发中，取得了巨大的成功。

## 1.1 C语言的发展简史和特点

### 1.1.1 计算机语言的发展过程

计算机语言是人与计算机之间进行信息交换的工具，是人机对话的媒介。计算机语言经历了从低级到高级，从面向机器到面向用户、面向对象的逐步发展的过程。

#### 1. 第一代：机器语言

计算机硬件系统只能执行由0、1二进制代码构成的操作指令。每一台计算机都有一套指令系统，指令系统中的每一条指令称为机器指令。

每一种类的计算机都有它特有的机器指令系统，这样的机器指令集合称为机器语言。

机器语言是面向特定机器的低级语言，不易记忆，通用性差，但它却是计算机机器惟一能直接执行的语言。

#### 2. 第二代：汇编语言

汇编语言又称符号语言。汇编语言用简单、易记忆的助记符来取代机器语言中的操作码，用十进制或十六进制数取代机器语言中的操作数，对机器指令进行简单的符号化。

用汇编语言编写的程序，称为汇编语言源程序。计算机不能直接执行这种程序，必须经过汇编程序将其汇编形成目标程序，并连接生成可执行文件，才能执行。不同种类的计算机都随机带有不同的汇编程序。

汇编语言仍属于面向机器的低级语言。由于汇编程序执行效率高、执行速度快，所以在实时控制、实时数据处理领域得到广泛应用。

#### 3. 第三代：高级语言

20世纪50年代后期至60年代中后期，出现了高级语言和操作系统。高级语言使用接近人类习惯使用的自然语言来编写计算机程序，如BASIC, FORTRAN, C, PASCAL等。

由于高级语言与具体的计算机机器指令系统无关，因而高级语言是一种面向操作者（用户）的语言。用高级语言编制的程序能在不同类型的计算机上运行，通用性好，这大大地促

进了计算机应用的普及。

高级语言的种类很多，无论用哪一种高级语言编写的程序，必须经过解释或编译才能被执行。

编译程序读进整个源程序并将其转换为目标代码，然后由计算机直接执行，运行速度快。Turbo C 编译程序对大型的源程序，允许分块编译，可节省大量时间。解释程序一次只读一行源程序，并执行该行语句指定的操作；每次运行用户程序时必须用解释程序。

用传统的结构化程序设计方法编写的程序，是一系列预先定义好的操作序列的组合，往往由若干个功能模块组成。程序直接控制程序中的顺序、选择、循环和调用的执行流程，它是一种面向用户的程序设计方法。

#### 4. 第四代：甚高级语言

甚高级语言是一种面向对象或面向问题的高级语言。面向对象的程序设计方法从所要处理的数据入手，以数据为中心，把编程问题看作一个数据集合，而结构化程序设计方法考虑的是程序功能。

面向对象的程序设计中的“对象”，可以用这样一个描述来表达，即“对象” = “数据” + “作用于数据上操作”。

面向对象的程序设计方法具有抽象、继承、封装和多态性 4 个特征，较好地解决了编程中的代码重用问题。Visual Basic（简称 VB）语言、C++、Visual C++（简称 VC）语言都具有面向对象的编程优势。

### 1.1.2 C 语言的发展简史

在 C 语言诞生以前，人们希望能开发一种兼有汇编语言和高级语言特性的新语言，既可以用来开发应用软件，又能用来开发系统软件。

#### 1. C 语言的起源

C 语言的起源可以追溯到 ALGOL 60 语言。ALGOL 60 是一种用于科学计算的语言，它采用了精确而形式化的语法描述体系，孕育了许多很有用的程序设计思想。但 ALGOL 60 与计算机的硬件距离较远，不宜于用来编写系统程序。

1963 年英国剑桥大学推出 CPL (Combined Programming Language) 语言，企图在 ALGOL 60 的基础上和计算机硬件更接近一些，但其语言规模大，难以实现。因而，在 1969 年英国剑桥大学又推出简化的 CPL 语言，语言名称是 BCPL (Basic CPL)。

1970 年贝尔实验室的 UNIX 操作系统的创始人 Ken Thompson 在实施 UNIX 系统时，对 BCPL 作了进一步简化，设计了非常简单又接近计算机硬件的 B 语言。但 B 语言过于简单，它的适用范围小。

#### 2. C 语言的诞生

1970 年贝尔实验室的 D.M.Ritchie 在 B 语言的基础上，设计出 C 语言。C 语言保持了 BCPL 语言和 B 语言的精练、接近硬件的优点，克服了 B 语言“无类型”的数据结构等缺点。系统地引进多种基本数据类型，并导入了其他组合数据类型和函数。

1973 年 Ken Thompson 和 D.M.Ritchie 用 C 语言重新改写 UNIX 内核，推出 UNIX 的第 5 版本。1975 年 UNIX 的第 6 版本公布后，C 语言的突出优点才引起计算机专家们的普遍注意。与此同时，C 语言的编译器也被移植到 IBM 360/370、VAX-11/780 等计算机。

### 3. C 语言的推广应用

1977 年出现了不依赖于具体机器的 C 语言编译文本“可移植 C 语言编译程序”，推动了 UNIX 迅速在各种机器上的实现。随着 UNIX 系统在美国各大学和公司的普及，C 语言也迅速得到推广和应用。

### 4. C 语言的 ANSI C 标准以及普及

1978 年 Brian W·Kernighan 和 Ritchie 两人出版了 C 语言白皮书，书名为《The C Programming Language》。给出了 C 语言的详细的定义。

1983 年美国国家标准化协会 (ANSI)，根据 C 语言问世以来各种版本对 C 语言的发展和扩充，制定了 ANSI C 标准。此后的许多 C 语言新版本都参照 ANSI C 标准，这样就给 C 语言程序的移植创造了更有利的环境。

1987 年 ANSI 公布 C 新标准 87 ANSI C。

在 20 世纪 80 年代，C 语言更加普及，几乎在每一种操作系统上都有 C 语言的编译器，并成为微型计算机上的通用的编译工具。在微型计算机上常见使用的 C 语言编译系统有：Microsoft C、Turbo C、Borland C 等。本书使用 Turbo C V2.0 编译系统。

1990 年 ISO (国际标准化组织) 接受上述 87 ANSI C 标准，作为 ISO 的 C 标准。

### 1.1.3 C 语言的特点

C 语言是一种既适合系统软件开发，又适合应用程序设计的程序设计语言。C 语言兼有汇编语言和高级语言的特性，它主要有以下几个特点。

(1) C 语言具有多达 44 种的运算符，表达式简练、使用灵活、实用。C 语言具有丰富的数据类型，具有在基本数据类型的基础上，按层次构造多种复杂的数据类型，如数组、指针、结构体和联合体等。尤其是指针类型的使用，能方便地对复杂的数据结构进行数据处理。

(2) C 语言语句简洁，规整，C 语言程序由若干个函数定义的集合构成。C 语言中的函数提供了编制结构化程序的手段，程序结构清晰，易于阅读和进行程序维护。

(3) C 语言生成的目标代码质量高，程序执行效率高。C 语言有些运算符直接反映了机器指令，可生成较短的机器代码，其生成的目标代码效率仅比用汇编程序生成的目标代码效率低 10%~20%。

(4) C 语言具有较高的可移植性，没有依赖于硬件的输入/输出语句。I/O 功能通过调用 I/O 函数库来实现，这些函数是系统提供的独立于 C 语言的程序模块库，因而使得 C 语言本身可以不依附于机器的硬件系统。

(5) 对于一种机型的 C 语言核心编译器可以很小，系统的实用部分和预处理部分均与机器无关，这样使得 C 语言编译器从一个系统到另一个系统的移植改写变得很容易实现。

(6) C 语言是一种结构化程序设计语言，适合于大型程序的模块化设计。

## 1.2 认识 C 语言

### 1.2.1 C 语言程序的结构

#### 1. 几个 C 语言的程序

**例 1.1** 输出一个字符串：This is a C program.的 C 语言程序。



程序如下:

```
main()
{
    printf("This is a C program.\n");
}
```

程序运行结果:

This is a C program.

该程序仅由一个 main() 主函数构成, 一对单括号 “( )” 前是主函数的名称。一对花括号 “{ }” 内是函数体, 一般由若干条 C 语言语句组成, 本函数中只有一条语句, 语句的功能是调用标准输出库函数输出一个字符串。

**例 1.2** 编写输出两个整数 26、16 和的 C 语言程序。

程序如下:

```
main()
{
    int a,b,c;
    a=26; b=16;
    c=a+b;
    printf("%d \n",c);
}
```

程序运行结果: 42

该程序也是仅由一个 main() 主函数构成, 函数体中包含有 1 条变量说明语句, 4 条执行语句。语句的功能为: 定义 3 个变量, 变量的名称分别为: a, b, c; 给变量 a, 赋给整数 26; 给变量 b, 赋给整数 16; 求出两个变量 a, b 的数值和, 赋给第三个变量 c; 最后输出第三个变量的值。

**例 1.3** 从键盘输入两个整型数, 输出两数和的 C 语言程序。

程序如下:

```
main()
{
    int a,b,c;
    scanf("%d", &a);
    scanf("%d", &b);
    c=a+b;
    printf("%d \n",c);
}
```

该程序是将例 1.2 程序中的两条赋值语句, 换成两条调用标准输入库函数语句, 分别从键盘输入整型数赋值给变量 a, b。

**例 1.4** 从键盘输入两个整型数, 输出其中的一个大数。

程序如下:

```
int max(int x, int y)
{
```