

轻松学  
编程

# 轻松学

# C语言编程

张银鹤 宋礼鹏 郝春雨 等编著



DVD-ROM

## 653分钟多媒体教学视频

- **结构独特:** 每个知识点按照“概念→语法描述→实例描述→实例应用→运行结果→示例分析”的模式进行讲解
- **形式新颖:** 用准确的语言总结概念,用直观的图示演示过程,用详细的注释解释代码,用形象的比喻帮助记忆
- **内容丰富:** 重要知识点覆盖全面,实例丰富多彩

**赠光盘:** 在随书附赠的光盘中配备了完整的视频教学讲座和本书所使用到的全部项目文件与代码

**技术支持:** 读者可直接登录[www.itzcn.com](http://www.itzcn.com)与作者取得联系,会在第一时间内得到帮助



化学工业出版社

轻松学  
编程

轻松学

# C语言编程

张银鹤 宋礼鹏 郝春雨 等编著



化学工业出版社

· 北京 ·

本书从初学者的角度出发,以通俗易懂的语言、丰富多彩的实例,详细介绍了使用C语言进行程序开发应该掌握的各方面知识。全书共14章,首先介绍了C语言的基础知识,包括C语言的历史和特点、C程序的结构和开发流程、C语言的数据类型、变量、常量、运算符以及表达式等;然后重点对C程序的编写进行讲解,包括算法与顺序结构、分支语句、循环语句、数组和函数的应用等;另外,还对C语言中的指针、结构体、预处理指令、字符串和文件操作进行了简单介绍;最后通过一个综合案例介绍C语言在实际开发中的应用。

本书非常适合学习C语言编程的初学者,也可作为大、中院校师生和培训班的教材。

### 图书在版编目(CIP)数据

轻松学C语言编程 / 张银鹤, 宋礼鹏, 郝春雨等  
编著. —北京: 化学工业出版社, 2012. 4

(轻松学编程)

ISBN 978-7-122-13615-2

ISBN 978-7-89472-587-5 (光盘)

I. 轻… II. ①张… ②宋… ③郝… III. C语言—  
程序设计 IV. TP312

中国版本图书馆CIP数据核字(2012)第028420号

责任编辑: 瞿 微

装帧设计: 王 宇

出版发行: 化学工业出版社(北京市东城区青年湖南街13号 邮政编码100011)

印 装: 化学工业出版社印刷厂

787mm×1092mm

1/16

印张 23

字数 589千字

2012年4月北京第1版第1次印刷

购书咨询: 010-64518888(传真: 010-64519686)

售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价: 45.00元(含1DVD-ROM)

版权所有 违者必究

C 语言具有高级语言和汇编语言的特点，受到了广大编程人员的喜爱。C 语言简洁紧凑、灵活方便，运算符和数据类型丰富，语法限制不太严格，程序设计自由度大，适用范围广，适用于多种操作系统。由于 C 语言是结构式语言，并允许直接访问物理地址，可对硬件直接进行操作，因此学习并掌握 C 语言，既可以了解计算机底层工作机制，又为学习其他高级编程语言打下基础。

## ● 本书内容

第 1 章 C 语言概述。介绍了 C 语言的产生背景、语言特点、标准、与 C++ 的关系、程序的基本结构和程序流程，以及两款主流的 C 语言开发工具。

第 2 章 C 语言数据类型。介绍了 C 语言中的变量、常量、标识符、关键字、数据类型、数据类型间的转换以及 C99 中新增的数据类型。

第 3 章 运算符和表达式。介绍了 C 语言中丰富的运算符，包括基本运算符、位运算符和其他运算符，运算符的优先级和结合性，以及表达式的定义和使用。

第 4 章 顺序结构。介绍了 C 语言算法的概念、算法的常见描述方式、顺序结构中的语句以及格式化输入和输出函数。

第 5 章 分支结构。介绍了 C 语言中实现分支结构的语句，分别是 if 分支语句和 switch 分支语句，其中 if 分支语句包括双分支和多分支语句以及嵌套 if 语句。

第 6 章 循环结构。介绍了 C 语言中实现循环结构的语句，分别是 for 循环语句、while 循环语句、do while 循环语句，以及循环控制语句和循环嵌套。

第 7 章 数组。介绍了 C 语言中的一维数组、二维数组、多维数组、字符数组和字符串，以及数组的基本应用。

第 8 章 函数。介绍了如何定义一个函数、为函数指定值参数和引用参数、函数的返回值、嵌套和递归调用函数，以及函数中的变量作用域等。

第 9 章 指针。介绍了 C 语言指针的概念、指针变量的声明和使用、将指针变量作为函数参数、指针与数组、字符串、函数的关系以及二级指针。

第 10 章 结构体与枚举体。介绍了如何定义结构体、如何声明结构体变量、结构体数组、指针与结构体的关系，以及枚举类型的声明和使用。

第 11 章 预处理指令。介绍了根据作用划分的三大类预处理指令，包括宏定义指令、文件包含指令和条件编译指令。

第 12 章 操作字符及字符串。介绍了 C 语言处理字符及字符串的函数，关于字符串的操作大致可分为字符串的输入、输出、连接、修改、转换、比较、复制、查找几大类。

第 13 章 文件操作。介绍了 C 语言中文件操作的各种函数，如打开和关闭文件、读取和写入文以及文件的定位等。

第 14 章 图书管理系统。介绍了使用 C 语言实现图书管理系统的过程，包括图书模块的录入、浏览、添加、删除、修改、查找功能的实现方法，以及学生借书模块的注册账号、借书操作、查看和清除账号的实现方法。

## ● 本书特色

本书从初学者的角度出发，通过通俗易懂的语言、丰富多彩的实例，详细介绍了使用 C 语言编写程序所需要的各方面技术。

### ● 结构独特

每个知识点都按照“概念→语法描述→实例描述→实例应用→运行结果→示例分析”的模式进行讲解，非常符合科学的学习思维，很容易掌握。

### ● 形式新颖

用准确的语言总结概念，用直观的图示演示过程，用详细的注释解释代码，用形象的比喻帮助记忆。

### ● 内容丰富

涵盖了使用 C 语言开发系统中所涉及的数据类型、运算符和表达式、顺序结构、分支结构、循环结构、数组、函数、指针、结构体与枚举体、操作字符串、预处理指令、文件操作等方面的各种技术。

### ● 随书光盘

本书为实例配备了视频教学文件，读者可以通过视频文件更加直观地进行学习。

### ● 网站技术支持

读者在学习或者工作的过程中，如果遇到实际问题，可以直接登录 [www.itzcn.com](http://www.itzcn.com) 与编者取得联系，我们会在第一时间给予帮助。

### ● 贴心的提示

为了便于读者阅读，全书还穿插着一些技巧、提示等小贴士，体例约定如下。

**提示：**通常是一些贴心的提醒，让读者加深印象、提供建议或解决问题的方法。

**注意：**提出学习过程中需要特别注意的一些知识点、内容或相关信息。

**技巧：**通过简短的文字，指出知识点在应用时的一些小窍门。

## ● 读者对象

本书具有知识全面、实例精彩、指导性强的特点，力求以全面的知识及丰富的实例来指导读者透彻地学习 C 语言各方面的知识。本书可以作为 C 语言的入门书籍，也可以帮助中级读者提高技能。

本书适合以下人员阅读学习。

- C 语言初学者。
- C 语言程序开发人员。
- 各大中专院校的相关授课老师。

## ● 本书编者

本书主要由张银鹤、宋礼鹏、郝春雨编写，其他参与编写、资料整理、程序开发的人员还有郭郑州、郝相林、陈军红、张水波、王俊伟、于利敏等。

由于编者水平有限，书中难免存在不足和疏漏之处，恳请读者批评指正。

编者

2012 年 2 月

<b>第 1 章 C 语言概述</b> .....	1
1.1 C 语言简介 .....	2
1.1.1 计算机语言简介 .....	2
1.1.2 C 语言的历史 .....	3
1.1.3 C 语言的特点 .....	3
1.1.4 C 语言的标准 .....	4
1.1.5 C 和 C++ .....	4
1.2 简单 C 程序的介绍 .....	5
1.2.1 C 语言程序的结构 .....	5
1.2.2 C 语言程序的开发流程 .....	7
1.2.3 程序注释 .....	9
1.2.4 输入和输出函数简介 .....	10
1.3 集成开发环境 .....	11
1.3.1 Turbo C++ 3.0 .....	11
1.3.2 Visual C++ 6.0 .....	14
1.4 实践案例：编写学生成绩管理程序 .....	17
1.5 动手练一练 .....	19
1.5.1 练习题 .....	19
1.5.2 上机实践 .....	20
<b>第 2 章 数据类型</b> .....	21
2.1 数据类型的定义与分类 .....	22
2.2 变量与常量 .....	23
2.2.1 变量 .....	23
2.2.2 常量 .....	24
2.3 标识符和关键字 .....	25
2.3.1 标识符 .....	26
2.3.2 关键字 .....	26
2.4 整型数据 .....	27
2.4.1 整型常量 .....	27
2.4.2 整型变量 .....	28
2.5 浮点型数据 .....	29
2.5.1 浮点型常量 .....	29
2.5.2 浮点型变量 .....	30

2.6	字符型数据 .....	32
2.6.1	字符型常量 .....	32
2.6.2	字符型变量 .....	32
2.7	类型转换 .....	34
2.8	C 99 中新增的数据类型 .....	36
2.9	实践案例：不同类型之间的运算 .....	36
2.10	动手练一练 .....	37
2.10.1	练习题 .....	37
2.10.2	上机实践 .....	38
<b>第 3 章</b>	<b>运算符和表达式 .....</b>	<b>39</b>
3.1	运算符概述 .....	40
3.2	基本运算符 .....	40
3.2.1	算术运算符 .....	40
3.2.2	赋值运算符 .....	43
3.2.3	自增和自减运算符 .....	44
3.2.4	关系运算符 .....	46
3.2.5	逻辑运算符 .....	48
3.3	位运算符 .....	50
3.3.1	位逻辑运算符 .....	50
3.3.2	位移运算符 .....	52
3.3.3	复合位赋值运算符 .....	53
3.4	其他运算符 .....	53
3.4.1	条件运算符 .....	53
3.4.2	逗号运算符 .....	55
3.4.3	sizeof 运算符 .....	55
3.5	运算符的优先级和结合性 .....	56
3.6	表达式 .....	57
3.7	实践案例：打印购物清单 .....	58
3.8	动手练一练 .....	60
3.8.1	练习题 .....	60
3.8.2	上机实践 .....	61
<b>第 4 章</b>	<b>顺序结构 .....</b>	<b>62</b>
4.1	算法简介 .....	63
4.1.1	什么是算法 .....	63
4.1.2	算法示例 .....	63
4.2	算法描述方式 .....	65
4.2.1	自然语言描述方式 .....	65

4.2.2	流程图描述方式 .....	65
4.2.3	N-S 流程图描述方式 .....	67
4.2.4	伪代码描述方式 .....	69
4.2.5	计算机语言描述方式 .....	70
4.3	顺序结构中的语句 .....	71
4.3.1	语句编写方式 .....	71
4.3.2	表达式语句 .....	72
4.3.3	空语句 .....	72
4.3.4	复合语句 .....	73
4.4	格式化输出和输入函数 .....	73
4.4.1	printf()函数 .....	74
4.4.2	scanf()函数 .....	78
4.5	实践案例：判断是否为素数 .....	80
4.6	动手练一练 .....	83
4.6.1	练习题 .....	83
4.6.2	上机实践 .....	85
<b>第 5 章</b>	<b>分支结构 .....</b>	<b>86</b>
5.1	if 分支语句 .....	87
5.1.1	单分支 if 语句 .....	87
5.1.2	双分支 if 语句 .....	89
5.1.3	多分支 if 语句 .....	90
5.1.4	if 嵌套语句 .....	92
5.2	switch 分支语句 .....	94
5.2.1	switch 语句的基本形式 .....	94
5.2.2	switch 语句中的 break .....	96
5.3	实践案例：了解两种分支语句的优缺点 .....	97
5.4	动手练一练 .....	100
5.4.1	练习题 .....	100
5.4.2	上机实践 .....	101
<b>第 6 章</b>	<b>循环结构 .....</b>	<b>102</b>
6.1	for 循环语句 .....	103
6.1.1	for 循环语句格式 .....	103
6.1.2	for 循环实例 .....	106
6.2	while 循环语句 .....	107
6.2.1	while 循环语句格式 .....	107
6.2.2	while 循环语句实例 .....	108
6.3	do while 循环语句 .....	109



6.3.1	do while 循环语句格式 .....	109
6.3.2	do while 循环语句实例 .....	110
6.4	循环控制语句 .....	111
6.4.1	break 语句和 continue 语句 .....	111
6.4.2	goto 语句 .....	113
6.5	循环嵌套 .....	115
6.5.1	循环嵌套格式 .....	115
6.5.2	利用循环嵌套打印菱形 .....	116
6.6	实践案例：求素数 .....	117
6.7	动手练一练 .....	119
6.7.1	练习题 .....	119
6.7.2	上机实践 .....	121
<b>第 7 章</b>	<b>数组</b> .....	<b>122</b>
7.1	了解数组 .....	123
7.1.1	使用数组的好处 .....	123
7.1.2	数组的概念 .....	124
7.1.3	数组的维数 .....	125
7.2	一维数组 .....	126
7.2.1	一维数组的声明 .....	126
7.2.2	一维数组的存储 .....	126
7.2.3	一维数组的引用 .....	127
7.2.4	一维数组的初始化 .....	128
7.3	二维数组 .....	129
7.3.1	二维数组的声明 .....	129
7.3.2	二维数组的存储 .....	130
7.3.3	二维数组的引用 .....	131
7.3.4	二维数组的初始化 .....	132
7.4	字符数组和字符串 .....	133
7.4.1	一维字符数组 .....	133
7.4.2	了解字符串 .....	134
7.4.3	字符串的输入输出 .....	135
7.4.4	字符串的输入输出函数 .....	136
7.4.5	二维字符串数组 .....	137
7.5	多维数组 .....	138
7.6	数组的基本应用 .....	139
7.6.1	反转字符串 .....	139
7.6.2	查找最大值 .....	140

7.6.3	数据排序 .....	141
7.7	实践案例：二维数组行和列的元素互换 .....	142
7.8	动手练一练 .....	144
7.8.1	练习题 .....	144
7.8.2	上机实践 .....	145
<b>第 8 章</b>	<b>函数</b> .....	<b>146</b>
8.1	函数简介 .....	147
8.1.1	C 语言中的函数 .....	147
8.1.2	函数的分类 .....	147
8.2	定义函数 .....	148
8.2.1	定义无参函数 .....	148
8.2.2	定义有参函数 .....	149
8.2.3	定义函数的注意事项 .....	150
8.3	声明函数 .....	151
8.3.1	声明函数的语法格式 .....	151
8.3.2	声明函数的位置 .....	152
8.4	函数的参数 .....	153
8.4.1	形式参数与实际参数 .....	153
8.4.2	值参数 .....	155
8.4.3	引用参数 .....	156
8.5	函数的调用和返回值 .....	158
8.5.1	函数调用的一般形式 .....	158
8.5.2	函数返回值 .....	159
8.5.3	函数的嵌套调用 .....	161
8.5.4	函数的递归调用 .....	163
8.6	数组作为函数的参数 .....	166
8.7	函数中的变量作用域 .....	169
8.7.1	局部变量 .....	169
8.7.2	全局变量 .....	171
8.8	实践案例：学生成绩管理 .....	172
8.9	动手练一练 .....	176
8.9.1	练习题 .....	176
8.9.2	上机实践 .....	178
<b>第 9 章</b>	<b>指针</b> .....	<b>179</b>
9.1	地址和指针的概念 .....	180
9.1.1	什么是地址 .....	180
9.1.2	什么是指针 .....	181

9.2	指针变量 .....	182
9.2.1	定义指针变量 .....	183
9.2.2	使用指针变量 .....	183
9.2.3	指针变量作为函数的参数 .....	186
9.3	指针与数组 .....	188
9.3.1	指向数组元素的指针与指向数组的指针 .....	188
9.3.2	指针变量在数组中的运算 .....	190
9.3.3	数组元素的操作 .....	193
9.3.4	指向多维数组的指针变量 .....	196
9.3.5	数组名（指针）作为函数参数 .....	201
9.4	指针与字符串 .....	204
9.4.1	字符串指针 .....	204
9.4.2	字符串指针作为函数参数 .....	206
9.4.3	字符指针和字符数组的区别 .....	207
9.4.4	字符指针数组 .....	209
9.5	指针与函数 .....	211
9.5.1	指向函数的指针 .....	211
9.5.2	返回指针值的函数 .....	212
9.6	二级指针 .....	214
9.7	实践案例：编写简单的成绩统计程序 .....	216
9.8	动手练一练 .....	221
9.8.1	练习题 .....	221
9.8.2	上机实践 .....	222
<b>第 10 章</b>	<b>结构体与枚举体 .....</b>	<b>224</b>
10.1	结构体 .....	225
10.1.1	定义结构类型 .....	225
10.1.2	定义结构变量 .....	226
10.1.3	引用结构变量 .....	228
10.1.4	初始化结构变量 .....	229
10.1.5	结构体应用举例 .....	230
10.2	结构体数组 .....	231
10.2.1	定义结构体数组 .....	231
10.2.2	初始化结构体数组 .....	233
10.2.3	结构体数组应用举例 .....	234
10.3	指针与结构体 .....	235
10.3.1	指向结构体变量的指针 .....	235
10.3.2	指向结构体数组的指针 .....	237

10.3.3	结构体变量作为函数的参数	239
10.3.4	指向结构体变量的指针作为函数的参数	240
10.4	用 typedef 定义类型	241
10.5	枚举类型	243
10.5.1	定义枚举类型及变量	243
10.5.2	使用枚举类型的一些说明	244
10.5.3	枚举类型应用举例	244
10.6	实践案例：员工信息录入和输出	247
10.7	动手练一练	249
10.7.1	练习题	249
10.7.2	上机实践	251
<b>第 11 章</b>	<b>预处理指令</b>	<b>252</b>
11.1	宏定义指令	253
11.1.1	不带参数的宏定义	253
11.1.2	带参数的宏定义	255
11.2	文件包含指令	259
11.2.1	文件包含指令的语法格式	259
11.2.2	文件包含指令应用实例	260
11.2.3	文件包含指令注意事项	262
11.3	条件编译指令	263
11.3.1	#ifdef 指令	264
11.3.2	#ifndef 指令	265
11.3.3	#if 指令	266
11.4	实践案例：根据不同时间输出不同问候	267
11.5	动手练一练	270
11.5.1	练习题	270
11.5.2	上机实践	272
<b>第 12 章</b>	<b>操作字符及字符串</b>	<b>273</b>
12.1	字符的操作	274
12.1.1	检查英文字母函数	274
12.1.2	检查字母数字函数	275
12.1.3	检查控制字符函数	276
12.1.4	十进制数字检查函数	276
12.1.5	可打印字符检查函数	277
12.1.6	包含空格的可打印字符检查函数	278
12.1.7	格式字符检查函数	278
12.1.8	小写英文字母检查函数	279

12.1.9	大写英文字母检查函数 .....	280
12.1.10	十六进制数字检查函数 .....	280
12.2	字符串的存储 .....	281
12.2.1	字符串的静态存储 .....	281
12.2.2	字符串的动态存储 .....	282
12.3	检测字符串长度函数 .....	283
12.4	查找字符串 .....	284
12.4.1	查找字符位置函数 .....	284
12.4.2	查找相同字符函数 .....	285
12.4.3	查找包含字符位置函数 .....	285
12.4.4	查找包含字符函数 .....	286
12.4.5	查找子串函数 .....	287
12.4.6	字符查找函数 .....	288
12.4.7	字符包含函数 .....	289
12.5	比较字符串 .....	289
12.5.1	比较字符串函数 .....	290
12.5.2	比较部分字符串函数 .....	291
12.5.3	比较时忽略大小写函数 .....	292
12.5.4	包含结束符的比较函数 .....	293
12.6	连接字符串 .....	294
12.6.1	连接字符串函数 .....	294
12.6.2	连接部分字符串函数 .....	295
12.7	复制字符串 .....	295
12.7.1	复制字符串函数 .....	295
12.7.2	复制部分字符串函数 .....	296
12.7.3	更省事的复制函数 .....	297
12.7.4	字符串复制函数 .....	298
12.7.5	带终止字符的复制函数 .....	298
12.7.6	字符串移动函数 .....	299
12.8	字符和字符串的转换 .....	300
12.8.1	大小写字母的转换 .....	300
12.8.2	转换大小写字符串 .....	301
12.8.3	字符串与数值的转换 .....	302
12.9	其他字符串函数 .....	303
12.9.1	字符串反转函数 .....	303
12.9.2	替换字符串中的字符函数 .....	303
12.9.3	分解字符串函数 .....	304

12.10 实践案例：字符串函数操作系统 .....	305
12.11 动手练一练 .....	307
12.11.1 练习题 .....	307
12.11.2 上机实践 .....	308
<b>第 13 章 文件操作</b> .....	<b>309</b>
13.1 文件的概念 .....	310
13.1.1 文件的分类 .....	310
13.1.2 文件指针 .....	311
13.2 打开文件与关闭文件 .....	311
13.2.1 打开文件 .....	312
13.2.2 关闭文件 .....	313
13.3 文件的读写 .....	314
13.3.1 fgetc()函数与 fputc()函数 .....	314
13.3.2 fgets()函数与 fputs()函数——字符串读写函数 .....	316
13.3.3 fread()函数与 fwrite()函数 .....	317
13.3.4 fscanf()函数与 fprintf()函数——格式化读写函数 .....	319
13.4 文件的定位 .....	320
13.4.1 重置文件指针 .....	321
13.4.2 定位文件指针 .....	322
13.4.3 获取文件指针位置 .....	323
13.5 出错检测 .....	324
13.5.1 feof()函数 .....	324
13.5.2 ferror()函数 .....	325
13.5.3 clearerr()函数 .....	325
13.6 实践案例：通讯录程序 .....	326
13.7 动手练一练 .....	332
13.7.1 练习题 .....	332
13.7.2 上机实践 .....	335
<b>第 14 章 图书管理系统</b> .....	<b>336</b>
14.1 需求分析 .....	337
14.2 系统设计 .....	338
14.3 系统实现 .....	339
14.3.1 创建工程 .....	339
14.3.2 功能选择界面 .....	340
14.4 图书管理模块 .....	342
14.4.1 录入图书信息 .....	342
14.4.2 浏览图书信息 .....	344

# Contents

14.4.3	添加图书信息 .....	345
14.4.4	修改图书信息 .....	346
14.4.5	删除图书信息 .....	347
14.4.6	查找图书信息 .....	347
14.5	学生借书模块 .....	348
14.5.1	注册一个账号 .....	348
14.5.2	借书操作 .....	348
14.5.3	查看账号信息 .....	349
14.5.4	清除账号信息 .....	350
14.6	总结 .....	350
<b>练习题参考答案 .....</b>		<b>351</b>

# 第 1 章

## C 语言概述

在众多的计算机语言中，C 语言是优秀的结构化程序设计语言。因为它既具有高级语言（如 Java）的特点，又具有汇编语言的特点。

C 语言的应用范围非常广泛，它不仅可以在软件开发方面编写各种应用程序，还可以作为系统语言编写不依赖计算机硬件的应用程序，甚至在科研方面也会用到 C 语言，如单片机以及嵌入式系统。

本章首先对 C 语言的背景知识进行介绍，然后讲解 C 程序的结构和开发流程，接下来介绍两款主流的 C 语言开发工具，最后介绍了一个完整的 C 语言程序的实现过程。通过本章的学习，读者将对 C 语言有一个全面、深入的了解，并掌握如何开发一个简单的程序。

### 本章学习要点：

- 了解产生 C 语言的背景
- 了解 C 语言的特点
- 了解 C 语言的标准及与 C++ 的关系
- 熟悉一个 C 语言程序的基本结构
- 掌握开发 C 语言程序的流程
- 了解程序注释、输入和输出函数的简单用法
- 熟悉 Turbo C++ 3.0 开发 C 语言程序的步骤
- 掌握 Visual C++ 6.0 开发 C 语言程序的步骤



## 1.1 C 语言简介

C 语言是一种面向过程的计算机程序设计语言，具有高效、灵活、功能丰富和移植性高等特点，受到广大编程人员的喜爱。

虽然 C 语言从产生到现在经历了几十年的发展，但它仍然是最受欢迎的语言之一。而且目前产生的很多新编程语言都是借鉴 C 语言而发展起来的，如 Java、C++ 和 PHP 等。这也是为什么大多数初学者学习编写程序时都选择 C 语言的原因。

### 1.1.1 计算机语言简介

所谓语言就是一种按照默认的约定，双方进行交流的方式。例如，汉语和英语是人与人之间交流最常用的语言。另外，不同种类的动物之间也有它们的语言。

同样，人和计算机交流信息，也要解决语言问题，需要创建一种计算机和人都能识别的语言，这就是计算机语言。按照计算机语言的发展过程可以为机器语言、汇编语言和高级语言三类。

#### 1. 机器语言

机器语言是低级语言，也称为二进制代码语言。计算机使用的是由“0”和“1”构成的二进制数组成的一串指令来表达计算机的语言。机器语言的特点是，计算机可以直接识别，不需要进行任何翻译。

#### 2. 汇编语言

汇编语言是面向机器的程序设计语言。为了减轻使用机器语言编程的痛苦，用英文字母或符号串来替代机器语言的二进制码，这样就把不易理解和使用的机器语言变成了汇编语言。这样一来，使用汇编语言就比使用机器语言便于阅读和理解程序。

#### 3. 高级语言

由于汇编语言依赖于硬件体系，并且该语言中的助记符号数量比较多，所以应用起来仍然不够方便。为了使程序语言能更贴近人类的自然语言，同时又不依赖于计算机硬件，于是产生了高级语言。这种语言，其语法形式类似于英文，并且因为远离对硬件的直接操作，而易于被普通人所理解与使用。有以下 3 个不同的发展阶段。

(1) 非结构化语言。这是高级语言发展的第一阶段，编程风格比较随意，只要符合语法规则即可，而且也没有严格的规范要求，程序中的流程可以随意跳转。此时，很多开发人员为了追求程序执行的效率而采用了很多“小技巧”，使程序变得难以阅读和维护。早期的 BASIC 和 FORTRAN 等都是属于这个阶段的语言。

(2) 结构化语言。为了解决非结构化语言带来的问题，提出了结构化程序设计的方法。它规定程序必须由具有良好特性的基本结构（顺序、分支和循环）构成，且程序中的流程不允许随意跳转，总是由上而下顺序执行各个基本结构。这个阶段编写的程序结构清晰，易于