21 世纪大学计算机基础系列教材

程序设计基础（C语言）

袁 磊 李 勇 主编

科学出版社
www.sciencep.com

21世纪大学计算机基础系列教材

程序设计基础(C语言)

袁 磊 李 勇 主编

科学出版社

北 京

内 容 简 介

程序设计基础是大学计算机基础核心课程，课程的目标是向学生传授程序的基础知识和基本的程序设计方法，培养学生程序设计的基本能力。本书面向首次学习程序设计的读者，以C语言为基础讲述程序设计的基础知识和方法。全书内容包括程序设计的基本知识、数据类型与表达式、程序控制结构、结构化程序设计、数组、结构体、文件等内容。本书的特点是C语言的使用与程序设计方法紧密结合，并且基本平衡，指针自然地分散到相关章节之中。

本书可作为一般本科院校非计算机专业程序设计基础的教材，特别是课程学时较少（36～42学时）的情况。本书兼顾计算机等级（二级）考试的内容，也可作为计算机等级考试的培训教材。

图书在版编目(CIP)数据

程序设计基础：(C语言)/袁磊，李勇主编. —北京：科学出版社，2010

(21世纪大学计算机基础系列教材)

ISBN 978-7-03-026352-0

Ⅰ.程… Ⅱ.①袁…②李… 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(2010)第003467号

责任编辑：高 嵘/责任校对：王望容

责任印制：彭 超/封面设计：苏 波

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

武汉市首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2010年1月第 一 版 开本：787×1092 1/16

2010年1月第一次印刷 印张：15 1/2

印数：1—3 000 字数：353 000

定价：26.80元

（如有印装质量问题，我社负责调换）

前　言

程序设计是一项综合性的工作。它至少涉及到表达程序的程序设计语言；处理用程序设计语言表达的程序，使得程序能够被计算机执行，并为程序设计提供支持的程序设计平台；被处理数据的组织（数据结构）方法，以及数据处理的过程（算法）三个方面。一个合格的程序设计人员需要经过数年的学习和实践。所以，对于初学者来说，学习程序设计是一件非常艰辛的事情，需要学习者有非常强的耐心和实践精神。

程序设计基础面向初学程序设计者开设的程序设计课程，是大学计算机基础的必修核心课程，其教学目标是向学生传授程序设计的基础知识和基本方法，使学生具有在一种程序设计平台上进行程序设计的初步能力，为今后应用计算机解决实际问题打下基础。由于不同的职业岗位对计算机应用能力的要求不同，不同层次的学校，不同类型的专业在程序设计基础所选择的教学内容、教学侧重面和教学要求上有所不同。教学改革和课程建设也正是围绕着这一核心开展的。本书面向培养应用型人才的普通本科院校的理工科专业，特别是学时较少（讲授：36～42 学时，实验：12～18 学时）的情况。本书以标准 C 语言为平台介绍程序设计的基础知识和基本方法，以 Turbo C 2.0 为实验训练平台，实践课堂讲授的 C 语言程序设计的正确表述和使用方法，训练学生使用 Turbo C 2.0 程序设计平台进行程序设计的能力。为此，本书配有《程序设计基础（C 语言）实验指导与习题解答》，以帮助初学者完成必要的实验和训练。

本书先宏观介绍程序设计基本知识，建立必要概念，再逐步具体细化程序设计方法的基本思路编写。第 1 章介绍程序、程序设计语言、算法、数据、数据结构、程序设计等最基本的概念和相关的基础知识，介绍了 C 语言中程序的基本构成和形式。第 2 章具体化了数据类型、常量、变量、表达式等概念，以及在 C 语言中的表现形式和语法、运算规则。第 3 章介绍程序控制结构、程序的设计和最常用的基本算法思想，并例举了较多的例子来示范程序设计。第 4 章介绍了面向过程的程序设计中设计大型、复杂结构程序的处理方法，以及 C 语言的解决方法和相关的变量独立性问题。第 5 章、第 6 两章介绍复杂数据结构的处理方法，以及定义新的数据类型之后数据处理算法的实现，体现出算法与数据结构的密切关系，同时介绍复杂结构的数据在程序模块（函数）之间传递的方法。第 7 章介绍实现数据与程序分离，保持数据的独立性和可共享的方法——文件及操作方法。指针是 C 语言最有特色的部分，也是 C 语言中最复杂、最困难的地方，本书采用了分散难点、逐步学习和使用的方法，在第二章的最后就引进了地址、指针、指针类型及变量的概念，这样可以帮助学生更深刻地理解数据类型、存储单元、地址、指针、变量的概念，以及它们之间的关系，知道变量不仅可以按名直接访问，也可以通过指针变量间接访问，同时在以后的章节中可以直接介绍和使用指针进行数据传递、复杂结构数据类型的定义和访问等。这是一个从简单应用指针到复杂应用指针的过程。

本书在程序设计语言和程序设计算法思维两个方面的内容介绍上基本平衡。C 语言的介绍本着简洁、基本、规范使用的原则，有些使用的细节没有去追求，这一点也希望读者在上机实验和阅读学习他人程序时注意吸取和总结，不断提高对语言使用的技巧和能力。在算法思维的介绍上，以最基本的穷举、递推和迭代三种算法思想为基础和重点，在具体

应用这三种算法思想求解问题时，都给出了问题分析，以帮助读者理解程序设计算法思想的形成过程。本书内容和所配备的习题兼顾了全国计算机等级考试（二级）的要求，在《程序设计基础（C 语言）实验指导与习题解答》中还给出了计算机二级考试的模拟试题。

本书是在作者多年从事 C 语言程序设计的基础上，根据程序设计基础新要求而编写的。具有程序设计基础知识宏观概要，指导性强；C 语言介绍简洁明了，以标准使用方法为主；程序设计方法的介绍既注意一般规律和方法的介绍，更注重在具体问题求解过程中应用基本算法思想分析问题，逐步形成程序的过程，可借鉴性得到充分体现；指针难点分散，逐步加强应用等特点。

全书由袁磊负责编写内容的选取、写作大纲的编制和最后的统稿。袁磊编写了第 1 章、第 2 两章，丁函编写第 3 章，项东升编写第 4 章，李勇编写第 5 章、第 6 两章，王毅编写第 7 章。

本书中难免有不当之处，我们诚恳期待使用本书的老师和读者的批评指正或建议，以供今后在不断完善中参考。

编　者

2009 年 9 月

目　录

第1章 程序设计概述

随着计算机技术的发展，计算机的应用已深入到了社会和人们日常生活的方方面面。虽然，现在已有许多现成的计算机软件系统可以帮助人们解决问题，但要更好地利用计算机这一计算工具去不断地解决生活或工作中的问题，就要学会利用计算机语言去控制计算机按自己的想法工作，即进行程序设计。本章从宏观上介绍程序设计的基本知识，这有助于程序设计技术和方法的学习和把握。

本章学习目标

- 了解计算机程序、程序设计语言的发展、程序设计语言的基本元素、C语言的特点。
- 理解算法的概念、了解算法设计的基本方法、掌握算法描述的方法。
- 了解数据结构的概念、数据结构与算法之间的关系。
- 了解程序设计及程序设计的过程、程序设计技术的发展。
- 初步认识C语言程序的构成、C语言程序的编辑、编译和运行。

1.1 程序与程序设计语言

1.1.1 程序的基本概念

程序是完成问题求解过程的步骤的列表。要使计算机能按照人预定的想法去工作，就必须要把计算机完成工作的具体步骤用计算机能够认识的语言(指令)逐条编写出来，计算机执行这个语句序列以后，就能完成指定的工作。所以，计算机程序是用计算机语言表示问题求解过程的步骤的语句序列。

计算机程序主要包含数据结构和算法两个方面的内容。数据结构描述数据元素及数据元素之间的关系；算法描述处理这些数据的步骤。计算机程序有以下性质：

(1) 目的性——程序有明确的目的，程序运行完成时将实现预定的功能。

(2) 分步性——程序由计算机可执行(或自动处理)的一系列语句组成。

(3) 有序性——程序的执行步骤是有序的，不可随意改变程序步骤的执行顺序。

(4) 有限性——程序所包含的语句序列被执行时在有限的时间内可以结束。

(5) 操作性——通过程序中各步骤(语句)的操作,实现程序的最终目的。

程序设计语言是编写实现计算机程序的基础工具,是软件开发技术的重要组成部分和研究对象,其中渗透了大量的软件实现的技术和方法,是今后学习的重要内容之一。

1.1.2 程序设计语言的发展与分类

伴随着软件开发技术的发展,程序设计语言的发展大致经历了四代,可以分成面向机器的语言、面向过程的高级语言和非过程化的高级语言三类。

1. 机器语言

在计算机诞生初期,程序设计是直接使用机器的指令系统来实现的。机器语言即机器的指令系统,用二进制码表示。它与机器硬件系统直接相关。机器语言程序可以直接被计算机执行,所以执行速度快。但存在着难识别、不易记、难阅读、难理解、易错且不易查找等问题,要求程序员熟习计算机的硬件系统各部件及其工作方式。

2. 汇编语言

汇编语言是将计算机指令采用具有一定意义的助记符号表示的程序设计语言。这些助记符号通常与指令一一对应。所以它与计算机硬件系统密切相关,是面向机器的。汇编语言程序需要经过汇编程序的翻译,将符号指令译成机器指令后才能被机器执行,这一翻译过程称为汇编。汇编语言改善了机器语言难识别、不易记、难阅读、难理解、易错且不易查找等不足。

3. 面向过程的高级语言

20 世纪 50 年代人们就开始研究开发更易进行程序设计的语言。1957 年,由 IBM 公司的巴克斯领导的研究开发组开发出了第一个高级语言 FORTRAN 语言。FORTRAN 语言是针对科学计算而开发的高级语言,以后经历了多代的更新发展,至今仍在科学计算领域广泛使用。从 20 世纪 60 年代开始,人们研究开发了许多高级程序设计语言,从早期的面向过程的高级语言,如 COBOL、C、PASCAL、BASIC 等,到现在广泛使用的面向对象的高级语言,如 C++、VisualBasic、JAVA、Delphi 等。至今高级程序设计语言的研究开发仍是一个非常活跃的领域。

高级语言是用接近于自然语言,经过专门设计的表达方式表达程序的程序设计语言。其特点是直观,好理解,便于记忆,大大改善了错误难于查找、不易维护的状况。它屏蔽了程序设计中与硬件相关的细节,实现了程序设计对机器硬件的独立性,使程序设计转向求解问题过程本身。高级语言把硬件系统的差别交给“翻译”系统去自动处理,所设计的程序相对低级语言具有更好的可移植性。

面向过程的高级语言其功能是围绕着问题求解的过程描述而设计的。语言主要通过变量的定义、语句和流程控制结构来表达问题求解的过程(程序)。使用这类程序设计语言进行程序设计时既要回答做什么,更要细致地告诉计算机怎么做的过程,其程序的执行是按照程序设计中所规定的流程来执行的。如 FORTRAN、COBOL、C、PASCAL、BASIC 等都属于这一类。

高级语言设计的程序必须经过“翻译”以后才能被机器执行。“翻译”的方法有两种,

一种是解释，一种是编译。解释是把源程序翻译一句，执行一句的过程，而编译是把源程序翻译成机器指令形式的目标程序的过程，再用链接程序把目标程序链接成可执行程序后才能执行。

4. 非过程化高级语言

非过程化高级语言称为第四代语言。在面向过程的语言中，问题求解不但要考虑做什么，同时还要考虑怎样做。这使得程序员要把精力放在计算机具体执行操作的过程上，程序设计复杂细致，效率不高。非过程化高级语言把求解问题的重点放在做什么上，只需向计算机说明做什么，而如何去做是由计算机自己生成和安排执行的。这类语言有以下种类的典型代表。

(1) SQL(结构化查询语言)。用于数据库查寻的程序设计，只要告诉计算机到数据库查找满足条件的信息即可，不需要说明怎样去查找的过程。

(2) 描述型语言。描述问题是什么，给出问题的描述，执行的步骤按语句对问题描述的逻辑来执行，如用于逻辑推理问题求解的 Prolog 语言等。

(3) 面向对象的程序设计语言。以对象为基础，把问题的求解视为对象之间相互作用的结果。对象是一个封装了对象特征(属性，即数据)和行为(方法，即操作过程)的抽象体，如"学生"这个对象的属性有学号、姓名、性别、年龄、电话等，行为有注册、查寻课程考试成绩等。对象可以通过继承来拥有其父类对象的属性和行为，提高了程序代码的重用性。对象在继承的过程中可以具有不同的数据类型或不同的行为，这称为对象的多态性。多态性提供了程序设计的灵活性、可扩展性。当对象被实例化(即各属性被明确为具体的数据)后，通过对象之间相互发送消息的方式来使程序得到执行，产生需要的结果。C++、VisualBasic、JAVA、Delphi 等语言都是面向对象的程序设计语言。这一类程序设计语言是目前程序设计的主流语言。

5. 高级语言的分类

高级语言可以有多种分类方式。常用的有按照设计要求、应用范围、描述问题等方式。其中按描述问题的方式分类是最常用的分类方法。其分类及特征如表 1.1 所示。

表 1.1 按描述问题的方式分类

语言类型	特征	典型语言
命令型语言	求解步骤用命令方式给出，按描述操作步骤执行	Fortran、Pascal、Basic、C
函数型语言	求解过程由函数块构成，通过调用函数块执行	Lisp、ML
描述型语言	描述问题是什么，按问题描述的逻辑来执行	Prolog、Gpss
面向对象语言	以对象为基础，消息驱动方式执行	C++、Java、Delphi、C#

1.1.3 高级语言的基本元素

1. 符号系统

每一种高级语言都有自己的符号系统，分为基本符号和标识符两类。基本符号规定了语言所使用的基本字母、数字和特殊符号。高级语言的基本符号一般有以下几种。

• 字母：26 个英文字母。有些语言区分大、小写，如 C 语言；有些语言不区分大、小

写，如 BASIC 语言。

• 数字：0～9，十个数字符号。

• 特殊字符：+、-、*、/、%、=、.、,、;、>、<、(、)、“、”等，各种语言所使用的特殊符号多少并不相同，各符号所表达的意思通常是遵循自然语言的语意的，如“+”表示加，“-”表示减，“>”表示大于。

需要注意的是，单字符的特殊符号往往不够用，一般语言都有自己的多字符基本符号，如在 C 语言中有++、--、+=、-=等多字符基本符号。

标识符是用来标识程序中的实体的符号，此符号代表实体的名。如程序中的常量、变量、过程、函数、语句等都是实体。一般语言规定标识符由字母和数字构成的字符串表示，必须以字母开头，根据语言的不同，标识符所允许的字符串的长度有所不同，长的允许达到 128 个字符，短的只允许 8 个字符。如 X，Y 表示变量，Fun1 表示一个函数的名等。使用标识符时最好见名知意，如 StudentName 表示学生名这个变量。需要注意的是有些标识符已被语言系统本身使用，在程序中不能再作他用。这些被系统使用了的标识符称为系统关键字。

2. 数据类型

数据类型是具有同种性质的数据的集合。它是计算机科学中的核心概念之一。在计算机程序设计语言中数据类型表明了数据的三个方面，即数据的取值范围、数据存储的存储单元的大小和这类数据上可以有什么运算操作。一般在语言中，数据类型分为以下两类。

(1) 基本数据类型。由语言系统直接提供使用，一般有字符型、整型、实数型、逻辑型等。例如，在 C 语言中，一般整型数的取值范围为 -2^{16}～$2^{16}-1$，用两个字节存储，整型数可以进行加法、减法、乘法、除法、求余数等运算。实数类型也有加法、减法、乘法、除法运算，但所用的运算命令与整型数的运算命令不同。

(2) 非基本数据类型。由语言系统提供构造方法，用户根据实际情况在基本数据类型的基础上构造出的数据类型。一般语言都提供构造数组、结构等类型方法。例如，在 C 语言中，数组的定义方法为

```
int a[10];
```

它表示定义了一个名为 a，有 10 个整型数构成的数组。它用一块 20 个字节的连续存储空间来存储。

3. 常量与变量

程序中的数据以常量或变量的形式出现。常量是在程序中不能发生变化的量，如3.14这个具体的数在程序中是不会变的。常量也可以用标识符表示，该标识符称为符号常量。

变量是在程序执行过程中可以发生变化的量。变量是程序中的基本实体，用一个标识符表示。一个变量与一个存储单元相对应，在程序中代表了该存储单元中存放的数据，此数据称为变量的值。变量在程序的执行过程中之所以会发生变化是此存储单元中存放的数据发生了变化。例如，X 是一个变量，开始时它的值是 1，在以后的处理中它可能被修改为 2 或其他数值，即 X 变量的存储单元中的数据变成了 2 或其他数据。

变量分为局部变量和全局变量两类。局部变量是只允许在某个局部(或模块)使用的

变量，全局变量是允许在整个程序的多个模块中使用的变量。

4. 表达式

由运算符连接起来的一个字符串，表达对有关参加运算的实体，如变量、常量、函数等实施的运算。每个表达式都有一个确定的值。一般一种语言至少会提供以下三种表达式形式。

(1) 算术表达式。实施加、减、乘、除、求余数等算术运算的表达式。例如，

$$2*(X+Y)-3.14159*\sin(2.0*Z)/4$$

是一个 C 语言算术表达式。

(2) 关系表达式。由关系运算符(>、≥、<、≤、=、≠)连接表达的表达式。其值是一个逻辑值。例如，X>=5 是一个 C 语言关系表达式，它表示自然语言中的 5≤X。

(3) 逻辑表达式。由逻辑运算符(与、或、非)连接表达的表达式。其值是一个逻辑值。例如，

$$(X>=5)\&\&(X<=10)$$

是一个 C 语言逻辑表达式。它表示自然语言中的 5≤X≤10。

从以上表达式的形式可见，它们与大家熟习的自然语言的表达方式没有多大的区别。需要注意的是，表达式的书写格式为横向在一行上书写，没有竖式表达式。表达式中各运算符有运算的优先性，这一点与以往所熟习的优先性基本没有区别。

5. 语句

语句一般可以由语句定义符、基本元素(如变量、常量、函数等)、表达式和分隔符号来构成。例如，

```
x=5+y;
```

C 语言的赋值语句，语意为将 5+y 的值赋给 x，";"是语句分隔符。

```
printf("%d",x);
```

C 语言的输出语句，语意为将变量 x 的值输出到屏幕上，printf( )是函数。

```
10 LET X=5+Y
```

BASIC 语言的赋值语句，10 是行号，LET 是语句定义符，语意为将5+Y的值赋给 X。该语句之后没有分隔符，因为 BASIC 语言规定一行只写一条语句。

```
20 PRINT X
```

BASIC 语言的输出语句，20 是行号，PRINT 是语句定义符，语意与 C 语言的输出语句相同。

6. 控制结构

控制结构规定了程序中语句的执行顺序。在程序设计语言中至少提供顺序结构、选择结构和循环结构三种基本控制结构。这些控制结构一般通过语句来提供。

7. 程序

由若干个语句按语法规定以列表的形式构成，不同的语言其程序的外在表现不同。以下是两个具有同样功能的 C 语言和 BASIC 语言书写的程序。

C 语言程序

```
main( )
 {
   int x,y,sum;
   scanf("%d,%d",&x,&y);
   sum=x+y;
```

BASIC 语言程序

```
10 INPUT "x,y",x,y
20 LET SUM=X+Y
30 PRINT SUM
40 END
```

```
    printf("% d",sum);
}
```

8. 扩展结构

为了使得程序在结构上更加清晰，避免程序段的重复书写，提高程序的重复使用，可以把具有某些特定功能的程序段独立出来，这样的程序段称为子程序（过程或函数）。这些程序段自身不能单独运行，而是必须通过在某个程序中被调用的方式来执行，调用程序称为主程序，被调用程序称为子程序。

这样的子程序段（过程或函数）可以作为程序的一个单元在程序中使用，这使得这些程序段可以被反复重用，提高了编程的效率，简化、清晰程序结构。

9. 注释

注释是程序的非有效部分，仅供在阅读理解程序时使用。不同的语言，注释的表示方法不同。例如，

```
x=5+y;      /*将 5+y 的值赋给 x*/
```

这里用"/*"和"*/"是 C 语言规定的注释起、止符，括起来的部分是注释的内容。

```
x=5+y;      //将 5+y 的值赋给 x
```

这是 C++语言的注释表示形式，"//"是注释的起始符。

不同的程序设计语言所提供的程序设计语法元素及表达方式是不同的，这里所介绍的程序设计语言基本元素，在绝大多数程序设计语言中都有，其基本语义是相同的，只是表现形式有所不同。掌握这一点，对于快速掌握一种程序设计语言是非常有好处的。

1.1.4 C 语言的发展及特点

1. C 语言发展概况

C 语言是一种应用广泛并受到用户欢迎的计算机程序设计语言。它既有高级语言的特点，又具有汇编语言的特点。它可以作为系统软件的设计语言，编写计算机系统软件程序，也可以作为应用软件设计语言，编写不依赖计算机硬件的应用软件程序。因此，它的应用范围广泛。

C 语言的开发与 UNIX 操作系统的开发密切相关。20 世纪 60 年代末，美国贝尔实验室为了开发 UNIX 操作系统，希望能研制开发一种具有汇编语言特点的高级语言。1970 年，在英国剑桥大学的 Matin Richards 开发的 BCPL（B combined programming language）语言的基础上，Ken Thompson 将 BCPL 进行了修改，并为它起了一个有趣的名字"B 语言"。意思是将 CPL 语言煮干，提炼出它的精华。并且他用 B 语言写了第一个 UNIX 操作系统。1973 年，贝尔实验室的 D. M. Ritchit 在 B 语言的基础上进一步简化，最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，即 C 语言。他用 C 语言写成了第一个在 PDP-11 计算机上实现的 UNIX 操作系统。为了使 UNIX 操作系统推广，1977 年 Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》，即著名的 ANSI C。1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchit 合著了著名的《The C Programming Language》一书。随着 UNIX 的广泛使用，C 语言也迅速

得到推广。

1983年，美国国家标准协会(American National Standards Institute，ANSI)根据C语言问世以来的各种版本，对C语言进行了扩充和规范，公布了ANSI C。随着微型计算机的日益普及，为了适应C语言在微型机上的使用，1987年，美国ANSI又公布了新修订的87 ANSI C。1999年，国际标准化组织(ISO)在87 ANSI C语言的基础上，修订公布了新的ISO C 99。

目前在微型计算机上常使用的C语言集成开发环境有微软公司开发的Microsoft Visual C++和Microsoft C，Borland公司开发的Borland C++，Turbo C等。

2. C语言的特点

事实证明，C语言是一种实用且极具生命力的程序设计语言，它的特点可归结为以下几点。

(1) 简洁紧凑、灵活方便。C语言一共只有32个关键字，9种控制语句，输入/输出通过函数来实现。这使得其语言成分和编译系统简洁紧凑。函数是C语言的主要结构成分，函数可以在程序中被定义，完成独立的任务，可独立地被编译成可执行目标程序，供共享调用。这使得程序编写灵活方便，可重用性高，编写效率高。

(2) 运算符丰富，包含的范围广泛。C语言的运算符有34个。它把赋值、括号、强制类型转换等都作为运算符处理。从而使C的运算类型极其丰富，表达式类型多样化，灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(3) 数据类型丰富。C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型、枚举类型等，能用来实现各种复杂的数据结构(如栈、队、链表、树等)及其操作运算。尤其是引入了指针概念，使程序更是灵活、多样、执行效率高。

(4) C语言是结构式语言。它具有结构化的流程控制语句(如if-else语句、switch语句、while语句、do-while语句、for语句等)，支持模块内的结构化编程；它通过函数、预编译支持模块划分，实现程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于调试、使用以及维护。

(5) C语言语法限制不太严格，程序编写自由度大。C语言编写中区分大、小写字母，主要用小写字母编写。虽然C语言也是强类型语言，但它的语法比较灵活，书写没有严格的规定，允许程序编写者有较大的自由度。

(6) C语言允许直接访问物理地址，可以直接对硬件进行操作。C语言可以像汇编语言一样对位、字节、地址、寄存器等进行操作，而这三者是计算机最基本的工作单元。因此它既具有高级语言的功能，又具有低级语言的许多功能。这使得它很适合用来编写系统软件(UNIX操作系统的90%代码是用C语言写成的)。

(7) C语言程序生成代码质量高，程序执行效率高。一般它只比汇编程序生成的目标代码效率低10%～20%。

(8) C语言适用范围大，可移植性好。C语言配有图形库函数，具有强大的图形功能，支持多种显示器和驱动器。另外，C语言适合于多种操作系统，如Windows、DOS、UNIX、Linux等；也适用于多种机型，从微机、小型机到大型机都可以使用。

C语言优点很多，但其缺点也很明显。如运算优先级过多，比较复杂，并且不同的编译系统也有所不同；数值运算能力方面不如其他高级语言，在精度把握上需要程序员作专

门的处理；语法限制不太严格，书写自由度大，这导致有时书写与语义不一致；对变量的类型约束不严格，影响程序的安全性，对数组下标越界不作检查等。从应用的角度，C 语言比其他高级语言较难掌握。

C++语言是在 C 语言的基础上发展起来的。它改进了 C 语言的许多缺点和不足，特别是在数据的封装性、指针的使用上得到了显著的改进，使得数据的安全性得到增强。另外增加了面向对象的程序设计功能。

1.1.5 C 语言字符集、标识符与关键字

1. C 语言字符集

C 语言所使用的字符集包括有英文字母、阿拉伯数字和一些特殊字符。具体归纳如下。

(1) 英文字母：大、小写各 26 个字母，共计 52 个。

(2) 阿拉伯数字：0～9，十个数字符号。

(3) 下划线：_。

(4) 特殊字符：

```
+   -   *   /   %   ++   --
=   ==   !=   >   >=   <   <=
&&   ||   !   &   |   ^   ~   <<   >>
(   )   [   ]   {   }   \
#   ?   ;   .   ,   :   '   "
```

(5) 空白字符：空格符、换行符、制表符、回车符等。

2. C 语言标识符构成

标识符是用来标识程序中的实体的符号，此符号称为所代表实体的名。C 语言规定，标识符由英文字母、下划线和数字符号排列构成，英文字母大、小写含义不同，必须以字母或下划线开头。

符合 C 语言标识符构成规定的合法标识符，如 Abc，a23，x，X，Student_1，_123，ABC32xyt_w2 等；不符合 C 语言标识符构成规定的非法标识符，如 M. N，12xy，York. M. C，$ std，axy/12b，8_stu 等。

需要注意的是不同的 C 语言编译系统，对标识符字符串的长度规定有所不同，长的允许达到 128 个字符，短的只允许 8 个字符。例如，早期微机上使用的 DOS 版 MS C 规定程序中标识长度不超过 8 个字符，即只有前 8 个字符被认为是有意义的，超过 8 个字符以外的字符不作识别。如 Teacher_1 与 Teacher_2 这两个标识符在该系统看来是相同的标识符。现在常用的 Windows 版 Microsoft C、Turbo C 2.0 系统，对于标识符的长度规定是不超过 32 个字符。另外，使用标识符时最好见名知意，如 StudentName 表示学生名这个变量。

3. C 语言的关键字

关键字是被语言系统预定义了具有特定含义的标识符。关键字不能作为程序中的实体名使用，用户只能按语言规定的使用方式使用。ANSI C 使用的关键字有 32 个，它们是：