

# 《C语言程序设计教程》

## 图书基本信息

书名：《C语言程序设计教程》

13位ISBN编号：9787121103841

10位ISBN编号：7121103842

出版时间：2010-3

出版社：电子工业出版社

页数：340

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

# 《C语言程序设计教程》

## 前言

计算机程序设计基础是高等学校各专业开设的一门必修的计算机基础课程，课程的重点在于培养学生的程序设计思想和程序设计能力，以适应当今社会对人才的需求。C语言由于其自身简洁、紧凑和灵活的特点，以及具备其他高级语言所不具备的低级语言的特性，而使得它成为一种在计算机软件设计和计算机程序设计教学中备受欢迎的程序设计语言。本书是作者多年教学经验和应用C语言体会的结晶，根据教育部高教司非计算机专业计算机教学指导分委员会提出的高等学校计算机基础课程教学基本要求，在广泛参考有关资料的基础上编写而成。内容分为基础篇、提高篇和实验篇。

基础篇主要包括程序设计和C语言基础知识，以及顺序、选择和循环三种结构化程序设计基本结构的C语言实现，使读者初步建立起利用C语言进行简单程序设计的思想，学会进行简单程序设计。

提高篇主要包括函数、编译预处理、数组等构造型数据类型、指针类型及对文件的操作，使读者理解并体会C语言模块化的编程思想及对数组、指针类型的应用，学会使用构造型数据类型和指针类型处理问题，学会对文件进行操作。实验篇共设计了10个实验，采用循序渐进的方式引导读者掌握C语言程序设计的特点，详细的上机实践练习，便于读者深入理解语法和培养程序设计能力。在实验篇的最后还设计了若干综合性实验项目以开拓读者的思路，激发读者的学习兴趣。同时本书在编写时兼顾了全国计算机等级考试的要求。本书在结构组织上合乎学习逻辑，内容循序渐进，每个知识点的介绍都以引起读者的学习热情和兴趣为出发点，以提高读者的程序设计思想和能力为目标，既注重理论知识，又突出实用性。书中例题丰富，注重实用，且均在Visual C++6.0环境下调试通过。各章均配有丰富的习题，以帮助读者深入理解教材内容，巩固基本概念，达到培养良好的程序设计能力和习惯的目的。本书可作为高等学校本科、高职高专软件专业及相关专业程序设计的入门教材，也可作为全国计算机等级考试的辅导教材，还可供广大程序设计初学者自学使用。本书具有以下特点：

突出算法理解，重视实际操作。加强对学生程序设计思想和实际编程能力的培养，以适应信息社会对人才的需求。提供多样式的学习环境。本书同时提供丰富的教学资源，包括课程学习网站例题源代码、配套教学用多媒体电子课件及习题指导与参考答案。例题源代码、配套教学用多媒体电子课件及习题指导与参考答案。注重可读性。本书的编写小组由具有丰富的教学经验，多年来一直从事计算机基础教育的一线资深教师组成，教材内容组织合理，语言使用规范，符合教学规律。

# 《C语言程序设计教程》

## 内容概要

《C语言程序设计教程(第2版)》分为基础篇、提高篇和实验篇。主要内容包括：程序设计和C语言基础知识，以及顺序、选择和循环结构程序设计。读者可初步建立起利用C语言进行简单程序设计的思想，学会进行简单的程序设计。提高篇的主要内容包括函数、编译预处理、数组等构造型数据类型、指针类型及对文件的操作，使读者学习并体会C语言模块化的编程思想及对数组、指针类型的应用，学会使用构造型数据类型和指针类型处理问题，学会对文件进行操作；实验篇共设计了10个实验，以加强编写程序的实战能力。

《C语言程序设计教程(第2版)》在编写时兼顾了全国计算机等级考试的要求。书中例题丰富，注重实用，且均在Visual C++6.0环境下调试通过。各章都配有丰富的习题。《C语言程序设计教程(第2版)》程序源代码、配套课后习题指导和教学用多媒体电子课件，请通过华信教育资源网免费索取。

《C语言程序设计教程(第2版)》可作为高等学校本科、高职高专软件专业及相关专业程序设计的入门教材，也可作为全国计算机等级考试的辅导教材，还可供广大程序设计初学者自学使用。

## 书籍目录

基础篇第1章 程序设计及C语言概况 1.1 程序设计的基本概念 1.1.1 程序和程序设计语言 1.1.2 程序设计 1.2 算法 1.2.1 算法及算法的特性 1.2.2 算法的描述工具 1.3 结构化程序设计方法 1.4 C语言的初步知识 1.4.1 C语言的起源与发展 1.4.2 C语言的特点 1.4.3 C语言程序的构成 1.4.4 C语言程序的上机调试过程 本章小结 习题1第2章 C语言基础 2.1 数据类型 2.2 标识符、常量和变量 2.2.1 字符集 2.2.2 标识符 2.2.3 常量 2.2.4 变量 2.3 基本类型数据 2.3.1 整型数据 2.3.2 实型数据 2.3.3 字符型数据 2.3.4 字符串常量 2.4 运算符及表达式 2.4.1 算术运算 2.4.2 赋值运算 2.4.3 自增、自减运算符 2.4.4 逗号运算 2.4.5 sizeof运算符 本章小结 习题2第3章 顺序结构 3.1 C语言程序的语句 3.1.1 说明语句 3.1.2 执行语句 3.2 数据的输入和输出 3.2.1 字符输入函数getchar() 3.2.2 字符输出函数putchar() 3.2.3 格式输出函数printf() 3.2.4 格式输入函数scanf() 3.3 程序举例 本章小结 习题3第4章 选择结构 4.1 关系运算 4.1.1 关系运算符 4.1.2 关系表达式 4.2 逻辑运算 4.2.1 逻辑运算符 4.2.2 逻辑运算的值 4.2.3 逻辑表达式 4.3 if语句 4.3.1 if语句的3种形式 4.3.2 if语句的嵌套 4.3.3 条件运算符和条件表达式 4.4 switch语句 4.5 程序举例 本章小结 习题4第5章 循环结构 5.1 while语句 5.2 do-while语句 5.3 for语句 5.4 转移语句 5.4.1 break语句 5.4.2 continue语句 5.5 程序举例 本章小结 习题5 提高篇第6章 函数 6.1 标准函数 6.2 函数的定义和返回值 6.2.1 函数的定义 6.2.2 函数的返回值 6.3 函数的调用和声明 6.3.1 函数的调用 6.3.2 函数的声明 6.4 函数的嵌套调用和递归调用 6.4.1 函数的嵌套调用 6.4.2 函数的递归调用 6.5 变量的存储类别 6.5.1 局部变量及其存储类别 6.5.2 全局变量及其存储类别 6.5.3 函数的作用域和存储类别 6.6 程序举例 本章小结 习题6第7章 编译预处理 7.1 宏定义 7.1.1 不带参数的宏定义 7.1.2 带参数的宏定义 7.2 文件包含 7.3 条件编译 本章小结 习题7第8章 数组 8.1 一维数组 8.1.1 一维数组的定义 8.1.2 一维数组的初始化 8.1.3 一维数组元素的引用 8.1.4 一维数组的应用 8.2 二维数组 8.2.1 二维数组的定义 8.2.2 二维数组的初始化 8.2.3 二维数组元素的引用 8.2.4 二维数组应用举例 8.3 字符数组 8.3.1 字符数组的定义 8.3.2 字符串与字符数组 8.3.3 字符数组的初始化 8.3.4 字符数组的输入/输出 8.3.5 字符串(字符数组)处理函数 8.3.6 字符数组的应用 8.4 数组元素作为函数实参 本章小结 习题8第9章 指针 9.1 指针变量 9.1.1 变量的指针和指针变量 9.1.2 指针变量的定义和初始化 9.1.3 指针变量的引用 9.1.4 指针变量作为函数参数 9.2 数组指针变量 9.2.1 数组指针变量的定义和引用 9.2.2 数组名和数组指针变量作为函数参数 9.2.3 二维数组的指针 9.2.4 指针与字符串 9.3 指针与函数 9.3.1 指向函数的指针变量的定义和赋值 9.3.2 指向函数的指针变量作为函数参数 9.3.3 指针函数 9.4 指针数组和二级指针变量 9.4.1 指针数组 9.4.2 main()函数的形参 9.4.3 二级指针变量 本章小结 习题9第10章 结构体与共用体 10.1 自定义类型标识符 10.2 结构体的定义与引用 10.2.1 结构体类型的定义 10.2.2 结构体类型变量、数组和指针变量的定义 10.2.3 结构体类型变量、数组和指针变量的初始化 10.2.4 结构体类型变量、数组和指针变量的引用 10.2.5 函数之间结构体类型数据的传递 10.2.6 用指针处理链表 10.3 共用体的定义与引用 10.3.1 共用体类型的定义 10.3.2 共用体类型变量的定义 10.3.3 共用体变量的引用 10.4 枚举类型 本章小结 习题10第11章 位运算 11.1 位运算符 11.2 位运算符的功能 11.2.1 按位与运算(&) 11.2.2 按位或运算(|) 11.2.3 按位异或运算(^) 11.2.4 按位取反运算(~) 11.2.5 按位左移() 11.2.7 复合位赋值运算符 本章小结 习题11第12章 数据文件 12.1 C语言文件 12.2 定义、打开和关闭文件 12.2.1 文件指针 12.2.2 打开文件(fopen()函数) 12.2.3 关闭文件(fclose()函数) 12.3 文件的输入/输出 12.3.1 fputc()函数和fgetc()函数 12.3.2 fgets()函数和fputs()函数 12.3.3 fprintf()函数和fscanf()函数 12.3.4 fread()函数和fwrite()函数 12.4 文件的定位 12.4.1 rewind()函数 12.4.2 fseek()函数和随机读/写 12.4.3 ftell()函数 本章小结 习题12 实验篇实验1 熟悉Visual C++ 6.0可视化集成开发环境实验2 顺序结构程序设计实验3 选择结构程序设计实验4 循环结构程序设计实验5 函数实验6 数组实验7 指针实验8 结构体和共用体实验9 文件实验10 综合编程附录A 运算符的优先级和结合性附录B 标准函数附录C ASCII字符编码表附录D 程序调试中常见错误信息一览 D.1 编译时的常见错误 D.2 连接时的常见错误 D.3 运行时的常见错误 D.4 错误提示中英文对照附录E 模拟试卷 模拟试卷(一) 模拟试卷(二)参考文献

1.1.1 程序和程序设计语言 什么是程序？怎样设计程序？这往往是计算机语言初学者首先遇到的问题。有人以为计算机是“万能”的，只要把任务告诉计算机，计算机就会自动完成一切，并给出正确结果。这其实是一种误解，要让计算机按照人的意志来完成某项任务，首先要制订该项任务的解决方案，再将其分解成计算机能够识别并可以执行的基本操作指令，把这些指令按一定的规则组织排列起来存放于计算机内存存储器中，当给出执行命令后，计算机按照规定的流程依次执行存放在内存存储器中的指令，最终完成所要实现的目标。人们把这种计算机能够识别并可以执行的指令序列称为程序。也就是说，程序是人与计算机进行“交流”的工具，它用我们常说的程序设计语言来描述。程序设计语言是计算机能够理解和识别的语言。它通过一定的方式向计算机传送操作指令，从而使计算机能够按照人们的意愿进行各种操作处理。任何一种程序设计语言都有一定的使用规则，通常称为语法规则。要学习程序设计语言，必须注意学习它的语法规则，就像学习汉语要学习汉语语法一样。而学习程序设计语言的目的就是为了设计计算机程序。 程序设计语言的种类很多，大体上经过了由低级语言到高级语言的发展过程，目前广泛使用的有C、C++、Java、Visual BASIC、Visual FoxPro等高级语言，这些高级语言采用的都是接近于人们熟悉的数学语言和自然语言的表达形式，使人们的学习和使用更加容易和方便。我们把由高级语言编写的程序称为源程序。显而易见，用高级语言编写的源程序，计算机不能直接识别并执行，因为计算机只能识别和执行二进制形式的指令或数据，因此，必须有一个工具先将源程序转换成计算机能够识别的二进制形式的程序，我们把这种二进制形式表示的程序称为目标程序，而承担转换的工具称为语言处理程序。每一种程序设计语言都有与它对应的语言处理程序，语言处理程序对源程序的处理方式有编译方式和解释方式两种，相应的转换工具分别称为编译程序和解释程序。编译方式是指将源程序输入到计算机中后，用相应的编译程序将整个源程序转换成目标程序，然后再通过装配连接程序形成可执行程序，最后运行可执行程序得到结果。目标程序和可执行程序都是以文件的方式存放在磁盘上的，再次运行该程序，只需直接运行可执行程序，不必重新编译和连接。 .....

# 《C语言程序设计教程》

## 编辑推荐

以引起读者的学习热情和兴趣为出发点。以提高读者的程序设计思想和能力为目标。合乎学习逻辑，内容循序渐进——基础篇、提高篇、实验篇。提供丰富的教学资源——程序源代码、课后习题指导、多媒体电子课件、课程学习网站。注重可读性，是作者多年的教学经验和应用c语言体会的结晶。

# 《C语言程序设计教程》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu111.com](http://www.tushu111.com)