

<<C语言程序设计>>

图书基本信息

书名：<<C语言程序设计>>

13位ISBN编号：9787508465456

10位ISBN编号：7508465458

出版时间：2010-2

出版时间：水利水电出版社

作者：胡宏智 编

页数：264

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：<http://www.tushu007.com>

<<C语言程序设计>>

前言

21世纪的信息时代,承担培养信息技术人才任务的高校在众多的程序设计语言中,选用“C语言程序设计”作为计算机类和多数理工类专业的重要基础课程之一,不仅仅是因为C语言从诞生之日起到现在一直流行,未来也会常葆青春,而且也是由于C语言自身固有的、其他程序设计语言难于媲美的优点所决定的。

任何事物都有两面性,C语言也不例外。

C语言介于高级语言与低级语言之间,具有晦涩难懂的指针概念及既严格又灵活的语法规则,对于初学者来说,确有其不易入门的缺点。

为了帮助初学者尽快掌握C语言的基础知识、基本概念,开拓程序设计的分析思路、建立模块化程序设计思想,培养编程技能,我们综合各位老师多年的教学经验,编写此教材。

本书通过188个示例,全面介绍了C语言语句的功能、用法,重点讲解问题求解的算法思想、编程思路。

全书共由8章和4个附录组成。

第1章C语言概述。

通过本章的学习,使读者了解C语言的主要特点、C语言源程序的结构组成、开发工具和程序设计步骤。

第2章主要介绍基本的数据类型、运算符、表达式与标识符、常量、变量的概念,基本的输入输出函数调用。

通过本章的学习,读者可以进行简单的顺序程序设计。

第3章主要介绍算法的概念、程序的选择和循环结构。

通过本章的学习训练,读者可以进行较复杂问题的求解算法描述,并转化为C语言描述的程序设计技能。

第4章全面介绍了函数的概念与编译预处理。

通过本章的学习与练习,逐步学会把复杂问题分解成若干个相对简单、独立的功能模块,实施分而治之,形成模块化程序设计思想。

学会如何把分散的多个程序文件组合成一个完整的C程序。

第5章介绍了数组的概念与用法。

通过本章的学习,读者将学会对数据类型相同的多个变量的简化定义、字符数组和字符串的引用的方法。

第6章着重介绍了指针的概念、指针与数组的关系、指针函数、函数指针、指针数组、数组指针、动态数组、图形处理函数等。

通过本章的学习,读者可以学会高级、高效的程序设计。

第7章介绍了结构体、共用体、枚举等构造数据类型及其变量的定义、引用方法。

第8章介绍了文件的概念、分类和文件的使用。

书中前面加*号的章节,可根据各自的具体情况,作为选学或因课时有限不作为讲授内容。

全书例题和习题中的程序一一经过上机验证通过,确保程序的正确性。

本书特色在于:(1)每一章的开篇都给出教学内容提要、教学目标、教师要求与对学生的要求。

对教与学都具有一定的建议或指导性参考价值。

(2)全书以实例为导引,避免空洞的抽象描述。

结合问题的分析与程序设计讲解概念、语句的应用和注意事项,做到理论联系实际。

<<C语言程序设计>>

内容概要

本书以程序案例为导向，全面介绍C语言的基本概念、数据类型、语句及其特点，深入浅出地讲解C语言程序设计的基本方法，注重读者程序设计思想和程序设计能力的培养。

全书共分8章，第1-4章介绍了C语言基础知识、基本的数据类型、运算符与基本的程序语句、C语言结构化程序的算法设计与程序的控制结构、函数及编译预处理，通过前4章的学习，读者可以掌握C语言简单程序设计的方法、步骤，C语言结构化、模块化程序设计的思想和方法。

第5-8章的内容包括：数组、指针、构造数据类型和文件的概念与应用。

读者通过5-8章的学习，能够利用C语言进行比较复杂的程序设计。

本书每章均设置适量的习题、常见问题的问和答，并附习题解答。

全书语言通俗易懂，内容取舍合理、由浅入深，适合用于普通高等院校本、专科的教材，也可供工程技术人员参考。

<<C语言程序设计>>

书籍目录

前言第1章 C语言概述第2章 基本的数据类型、运算符与程序语句第3章 算法设计与程序的控制结构第4章 函数与编译预处理第5章 数组第6章 指针第7章 构造数据类型第8章 文件附录A C语言的保留字与关键字附录B C语言运算符优先级结合性表附录C ASCII字符表附录D C语言常用库函数参考文献

<<C语言程序设计>>

章节摘录

一台计算机是由硬件系统和软件系统两大部分构成的，硬件是物质基础，而软件可以说是计算机的灵魂，没有软件，计算机就是一台什么也不能干的“白痴”，有了软件，才能灵动起来，成为一台真正的“电脑”。

所有的软件都是用计算机语言编写的。

计算机程序设计语言的发展，经历了从低级到高级的历程。

1. 低级语言 (1) 机器语言。

计算机只懂得由“0”和“1”组成的二进制数，二进制是计算机语言的基础。

计算机发明之初，只是用计算机的语言去指挥计算机干这干那，一句话，就是写出一串串由“0”和“1”组成的指令序列交由计算机执行，这种语言就是机器语言。

使用机器语言是十分痛苦的，特别是在程序有错需要修改时，更是如此。

而且，由于每台计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，造成了重复工作。

但由于使用的是针对特定型号计算机的语言，故而运算效率是所有语言中最高的。

机器语言是第一代计算机语言。

(2) 汇编语言。

为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一条特定指令的二进制串，比如，用“ADD A, 50”代表加法，“SUB B, 20”代表减法运算等，这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。

然而计算机是不认识这些符号的，这就需要有一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件，移植性不好，因而它仍然属于低级语言。

针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精炼而质量高，运行效率十分高，几乎等同于机器语言的效率，所以至今仍是一种常用而强有力的软件开发工具。

2. 高级语言 从最初与计算机交流的痛苦经历中，人们意识到，应该设计一种接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用的这样一种语言。经过努力，1954年，第一个完全脱离机器硬件的高级语言——FORTRAN问世了，50多年来共有几百种高级语言出现，其中影响较大、使用较普遍的有FORTRAN、ALGOL、COBOL、BASIC、LISP、SOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、VC、Delphi、Java、C#等。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的过程。

相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

20世纪60年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是人自为战，缺乏科学规范的系统规划与测试、评估标准，其恶果是大批耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以致几乎没有不出错的软件。这一切极大地震动了计算机界，历史上称之为“软件危机”。

人们认识到：大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。

程序的设计应易于保证正确性，也便于验证正确性。

1969年，提出了结构化程序设计方法，1970年，第一个结构化程序设计语言——Pascal语言出现，标志着结构化程序设计时期的开始。

20世纪80年代初开始，在软件设计思想上又产生了一次革命，其成果就是面向对象的程序设计。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线似的，在一个模块被执行完成前，

<<C语言程序设计>>

人们不能干别的事，也无法动态地改变程序的执行方向。

.....

<<C语言程序设计>>

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:<http://www.tushu007.com>