

C语言与程序设计

The C Programming Language

华中科技大学计算机学院
毛伏兵



群名称: C语言0910课堂2023秋
群 号: 494790738

■ 请加群:



群名称: C语言0910课堂2023秋
群 号: 494790738



2023/9/17

The C Programming Language

C程序设计语言

- 简称**C语言**

- **BCPL**语言进化成**B**语言，**B**语言又进化为**C**语言

- 是一种高级语言

- **C**语言被分类为高级语言，但实际上它是一种介于高级语言和低级语言之间的语言

- 上天入地，无所不能

- 几乎没有不能用**C**写出来的软件，没有不支持**C**的系统

- 很多流行语言、新生语言都借鉴了它的思想、语法

- 从**C++**，到**Java**，再到**C#**

一、学习C语言的意义

- C语言具有功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好等特点,因而被广泛用于开发应用软件和系统软件,成为程序设计的首选语言。
- 学好C语言,能为学习C++、Java、C#等现代程序设计语言奠定良好的基础。

二、程序设计的内涵

- 程序设计是能力培养的起点：计算机的本质是“程序的机器”，程序和指令的思想是计算机系统中最基本概念。
- 算法是程序设计的灵魂：算法+数据结构=程序
- 语言是程序设计的工具：用语言设计程序形如用文字写文章，这也是语言与程序的关系。

三、课程性质和任务

- 《C语言程序设计》是计算机专业的一门基础课，也是从事计算机软件工作的一门专业技能。本课程蕴含了丰富的程序设计理念，通过本课程的学习，使学生熟练掌握C语言的有关基础概念、语法规则、结构化程序设计方法和基本算法，能够用C语言表达基本算法、编写小型应用程序。
- C语言是一门实践性很强的课程，必须通过大量的编程实践和上机调试，方能掌握。

四、课程目标

- 通过本课程学习，训练逻辑思维能力；
- 使学生熟练掌握C语言的基本语法、语义与程序结构，掌握正确和规范的编程方法；
- 掌握程序设计的基本思想、基本概念和基本方法；
- 掌握一些常用、经典的算法及其C语言的编程实现；
- 使学生具备用C语言编写小型软件的能力；

五、学习建议（1）

- **采用整体学习方法：**在学习前面的内容时，如果遇到不懂的内容，先暂时接受，不要考虑为什么，待整个内容学习结束后再回头解决前面的问题。
- **培养基本思维模式：**首先掌握基本的思维过程，从固定思维模式入手；其次，正确理解和应用“自顶向下，逐步求精”的基本原则和思想方法。

学习建议（2）

- **掌握基本设计方法：**掌握常用算法的程序设计方法以及形如求交换两个变量的值，多项数据求和，多项数据最值等编程模式。
- **把握数据组织方式：**程序的核心是算法，算法的核心是数据处理，只有经过有效组织的数据（数组、字符串、链表、结构体和数据文件等）才便于算法处理。

学习建议（3）

- **通过大量编程实践：**这是一门实践性强的课程，只有经过大量的编写和调试程序，才能掌握程序设计方法。语言只是手段，设计才是目的。
- **合理选择开发平台：**作为C语言的初学者，建议选择那些操作简单、易学的开发平台来作为C语言程序设计的实践平台（**建议CodeBlocks或 Dev C++**）。
- **充分利用学习资源：**充分利用网络获取学习资源非常重要。互联网上有很多C语言程序设计的网站，是学习者很好的学习与交流平台。

Chapter 0 Course Arrangement

1.教材

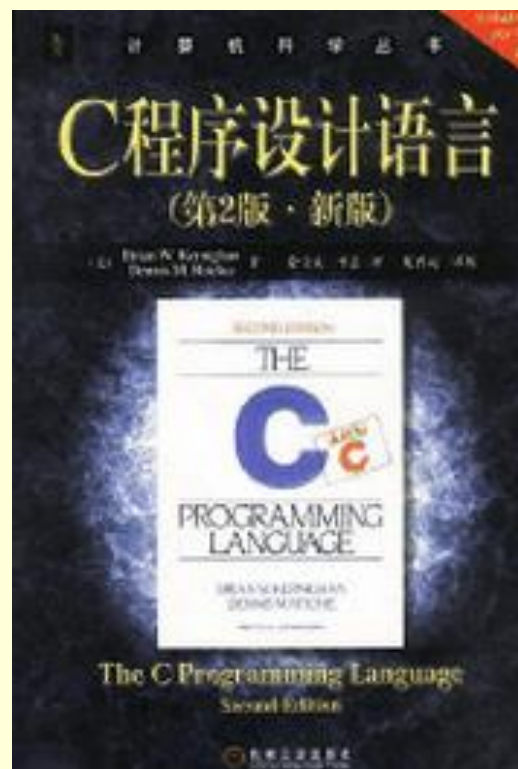
[1]卢萍,李开, 王多强, 甘早斌. **C语言程序设计**.
清华大学出版社, 2021.8.

[2]李开,卢萍,曹计昌.**C语言实验与课程设计**.科
学出版社,2011

2. 参考书目

(1) The C programming language,

K&R的书 如**C**语言的设计理念：
简单而高效
建议：里面的习题都认真做一遍，





C语言的设计者

Dennis M. Ritchie

在哈佛大学学习物理学和应用数学毕业，**1967**年进入贝尔实验室，目前是朗讯技术公司系统软件研究部门的领导人。

C语言之父，UNIX之父，对C语言和其他编程语言、Multics和Unix等操作系统的发展做出了巨大贡献。

1978年他和Brian W. Kernighan出版了名著《The C Programming Language》，成为C语言最权威的教材之一。

1983年他与Ken Thompson（Unix的设计者）一起获得了**图灵奖**。理由是他们“研究发展了通用的操作系统理论，尤其是实现了**UNIX操作系统**”。

1999年两人为发展C语言和Unix操作系统一起**获得了美国国家技术奖章**。



“C语言圣经”一书的作者 **Brian W. Kernighan**

贝尔实验室计算科学研究中心高级研究人员，著名的计算机科学家。他参加了**UNIX**系统、**C**语言、**AWK**语言和许多其他系统的开发，同时出版了许多在计算机领域具有影响的著作，包括

《**The Elements of Programming Style**》、
《**The Practice of Programming**》、
《**The UNIX Programming Environment**》、
《**The AWK Language**》、
《**Software Tools**》等。

(2) C专家编程

把c讲解到了一个很高的层次，深入剖析了其他书上没有提到过的好东西。

(3) C和指针

对指针的基础和深入的探讨，有助于初学者更好的理解C语言，并明白C的存储机制

(4) C标准库

C标准库“圣经”，提供完整源代码，全面深入阐述库函数的实现与运用。C程序员必备参考书。本书是由世界级C语言专家编写的C标准库经典著作。

(5) ISO/IEC 9899:2011(最新标准)

(6) 刘汝佳编的《算法竞赛入门经典》，清华大学出版社

3、参考网站

1. C语言中文网: c.biancheng.net
2. ANSI C standard library - Wikipedia
3. 北京大学程序在线评测系统: poj.org
4. 浙大程序在线评测系统: acm.zju.edu.cn
5. 华中科技大学的在线评测系统, 以其Virtual Judge功能最为著名: acm.hust.edu.cn/vjudge
6. 国外的UVa: Uva.onlinejudge.org
7. 与程序相关的IT技术问答网站, 程序员必上的开发者服务平台: <http://stackoverflow.com/>



4.考核方式

闭卷笔试：考试成绩 **70%** + 平时成绩**30%**
(到课率**10%**+书面作业**10%**+编程作业**10%**)

5.学习方法 (Study methods)

- (1) 课前要**预习**，课堂要专心，课后要复习
通过预习、听课、复习、以及作业和实验，确保消化课堂和教材的相关内容。
- (2) 要学会自己看书、理解，逐步减少对老师的依赖，培养良好的**自学能力**。同时，由于学时紧张，假期的冲击，讲课速度会逐步加快到一个合适的水平，以保证教学计划的完成，因此课前课后一定**研读教材**。
- (3) 认真、**独立的完成作业**。
- (4) **要重视实验**。上机实验前要求预先编好有关程序，把上机的重点放到程序的编辑、修改、编译、链接、跟踪、调试程序方面来。放到观察程序运行过程中的中间结果和运行完毕后的运行结果方面来。最终使自己的大脑变成一台“计算机”，能够在脑袋中运行自己的程序，判断各种情况下程序的走向等等。

6. 定位问题(Orientation)

- 培养的学生应该是将来能够与MicroSoft的程序员，软件大师们一争高下的人才，应该是有朝一日能够根据实际应用需求，发明创造新型计算机程序设计语言的人才。因此要**高标准、严要求**。不仅要掌握C的语法、语义，学好用好C；而且要了解各种语言成分产生、流传、延续、发展的深层次原因和需求背景。为学习后续课程，为将来能够熟练的用C写系统软件、底层软件、共享软件奠定坚实的语言基础。

7. 关于作业

- 编程题:

- 上机编辑、编译和运行程序
- 提交正确的源码

8.建议和意见

欢迎就教学方法，讲课速度，教学内容的深度和广度提出好的建议和意见，以便把本门课程的教学组织实施好。