

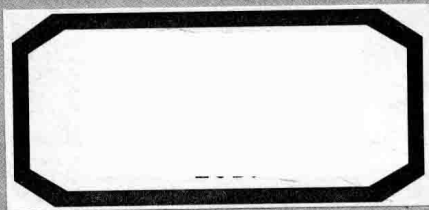
高等学校教材

程序设计方法与技术 ——C语言

主 编 顾春华
副主编 陈章进 叶文珺

高等教育出版社

高等学校教材



程序设计方法与技术 ——C语言

主 编 顾春华
副主编 陈章进 叶文珺

高等教育出版社·北京

本书以程序设计初学者为阅读对象,以程序设计解决问题为主线,以编程思维、编程技能、语法知识和编程规范为内容框架,通过丰富的实例由浅入深地介绍 C 语言程序设计的基本思想与方法。

本书导言部分介绍程序和程序设计及其教学建议,随后包括了程序设计概述、输入输出、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、结构体和指针等内容。为了提高读者的学习兴趣和成就感,各章节都选取了大量贴近生活的有趣案例;书中以思考、常见错误、编程经验等形式总结了程序设计的技术和方法。

本书适合作为高等院校各专业学生的教学用书,也可作为广大编程爱好者的自学读物,对从事软件设计与开发的技术人员也是一本很好的参考书。

图书在版编目(CIP)数据

程序设计方法与技术: C 语言 / 顾春华主编. -- 北京: 高等教育出版社, 2017.9
ISBN 978-7-04-048404-5

I. ①程… II. ①顾… III. ①C 语言-程序设计
IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 202171 号

Chengxu Sheji FangFa yu Jishu——C Yuyan

策划编辑 耿芳
插图绘制 杜晓丹

责任编辑 耿芳
责任校对 高歌

封面设计 李卫青
责任印制 耿轩

版式设计 马云

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	北京市白帆印务有限公司		http://www.hepmall.com
开 本	850mm × 1168mm 1/16		http://www.hepmall.cn
印 张	20.25		
字 数	440 千字	版 次	2017 年 9 月第 1 版
购书热线	010-58581118	印 次	2017 年 9 月第 1 次印刷
咨询电话	400-810-0598	定 价	40.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物料号 48404-00

数字课程资源使用说明

与本书配套的数字课程资源发布在高等教育出版社易课程网站，请登录网站后开始课程学习。

一、注册/登录

访问 <http://abook.hep.com.cn/185284>，点击“注册”，在注册页面输入用户名、密码及常用的邮箱进行注册。已注册的用户直接输入用户名和密码登录即可进入“我的课程”页面。

二、课程绑定

点击“我的课程”页面右上方“绑定课程”，正确输入教材封底防伪标签上的 20 位密码，点击“确定”完成课程绑定。

三、访问课程

在“正在学习”列表中选择已绑定的课程，点击“进入课程”即可浏览或下载与本书配套的课程资源。刚绑定的课程请在“申请学习”列表中选择相应课程并点击“进入课程”。

四、与本书配套的易课程数字课程资源包括电子教案、程序源代码、微视频等，以便读者学习使用。

账号自登录之日起一年内有效，过期作废。

如有账号问题，请发邮件至：abook@hep.com.cn。

前 言

在过去 50 多年中，程序设计技术与程序设计语言从来没有停止过创新和发展，未来，这种持续的改进仍将继续。程序设计课程的教和学也在不断进步，同时还会不断面临新的挑战。一直以来被很多高等学校作为第一门程序设计课程的 C 语言，由于其语言简单和思维清晰，成为程序设计课程中的常青树。尽管语言本身变化不大，但如何更有效地提高 C 语言教学效果的改革实践，一直都在进行中。

问题驱动、案例驱动、重在应用等教学思想，MOOC、SPOC、翻转课堂等教学技术和手段，都给经典的程序设计课程带来了新的活力与机遇。多年来，来自同济大学、华东理工大学、华东师范大学、上海大学、东华大学、上海理工大学和上海电力学院等多所高校的计算机基础教学一线教师结合计算机等级考试的持续改进和“以考促教”为目标，坚持开展程序设计课程的教学改革，与时俱进地进行教学重构，不断积累教学经验和教学资源。本书就是在这个基础上编写的，试图融合现代程序设计的新理念，平衡专业性与普适性，兼顾对学生的知识传授、能力培养与思维训练。

本书具有以下四个特点。

(1) 强调编程兴趣

选择贴近学生生活和年轻人感兴趣的案例，配上生动活泼的展示形式，注重激发学生学习编程的兴趣；通过提供可复用的公共库等形式，让学生通过简单的编程就能得到完整的程序和实用的结果，解决日常生活中的热点问题，增强学生学习成就感。

(2) 兼顾编程四个维度

本书强调编程的四个维度：编程思维、编程技能、语言知识点和编程规范。通过例题分析、经典算法等，以“思考”等形式描述常用的编程思维和思考问题的方式；分析、设计、编写、调试、运行程序，在此基础上归纳出“常见错误”，用以训练学生的编程技能；总结编程规范和经验，引导学生从一开始学习程序设计就养成良好的编程习惯。

(3) 由浅入深循循善导

内容组织上更突出从简单到复杂，将知识点的结构性和系统性淡化；将“指针”的概念和简单应用提前，将“文件”分散到章节而不独立成章；同一个问题从简单到复杂分解到多个程序例子中，让学生们从简单程序开始，逐步增加功能，在不知不觉中学会编程技能，习惯编程思维。

(4) 线上线下配有立体资源

配合本书同时建设了实验指导、习题库和知识点视频等立体化学习资源，设计了每一章的课堂教案设计、PPT 讲稿和网上教学平台等，便于学生预习、复习和自学，方便师生加强课堂互动，提高课堂教学效果。本书中的二维码都链接到一个网上资源，读者可在阅读时实时学习。

本书由上海市计算机等级考试二级命题组教师共同策划，得到了上海市教委优质在线课程项目和上海市教育考试院的支持。导言部分由顾春华编写，第1章到第9章分别由陈莲君、黄小瑜、陈优广、文欣秀、闫红曼、胡庆春、高枚、王淮亭、叶文珺、陈章进、朱弘飞、夏耘等编写。全书由顾春华、陈章进、叶文珺等修改统稿。刘江、吉顺如、张晨静、高建良等给本书提出了建议和帮助，对此一并表示感谢。

由于编者水平有限，书中难免存在错误与不足，恳请读者批评指正。

编者

2017年6月于上海

目 录

0 导言	1	1.5.4 运行调试	26
0.1 程序无所不在	1	1.6 常见 C 程序的错误	27
0.2 人人都要理解编程	2	1.6.1 语法错误——编译错误	27
0.3 解剖一个程序	4	1.6.2 语法错误——连接错误	28
0.4 编程的主要内容	5	1.6.3 逻辑错误——结果不正确	29
0.5 如何学好程序设计	7	1.6.4 逻辑错误——运行时错误	30
0.6 如何教好程序设计	9	小结	30
小结	10	习题 1	31
1 程序设计概述	11	2 输入输出	33
1.1 程序的概念	11	2.1 计算机与外界的交互	33
1.2 程序设计语言	11	2.1.1 输入输出设备	33
1.2.1 问题描述与程序设计	12	2.1.2 程序的输入输出	34
1.2.2 汇编语言和机器语言	13	2.2 信息的显示与录入	35
1.2.3 高级语言及其翻译	14	2.2.1 显示固定内容的信息	35
1.3 初识 C 程序	15	2.2.2 信息录入	37
1.3.1 C 语言概述	15	2.3 输入输出设计	40
1.3.2 数值计算的 C 程序	16	2.3.1 输出设计	40
1.3.3 简单游戏的 C 程序	17	2.3.2 输出的多样化	41
1.3.4 C 程序的实现过程	18	2.3.3 输入设计	45
1.4 C 程序的基本语法	20	2.3.4 输入的多样化	45
1.4.1 C 程序的基本结构	20	2.4 输入输出格式控制	48
1.4.2 C 程序的基本元素	21	2.4.1 显示内容格式控制	48
1.4.3 C 程序编程风格	22	2.4.2 数据输入格式控制	50
1.5 C 程序设计方法	23	2.5 综合案例	52
1.5.1 问题分析	23	小结	54
1.5.2 算法设计	24	习题 2	55
1.5.3 程序编写	26	3 顺序结构程序设计	57

3.1 顺序结构	57	习题 4	115
3.1.1 设计顺序结构程序	58	5 循环结构程序设计	119
3.1.2 语句的分类	59	5.1 自动售货机问题	119
3.2 表达式语句	60	5.2 三种循环结构	122
3.2.1 表达式	60	5.2.1 while 语句	122
3.2.2 算术运算符	61	5.2.2 do-while 语句	127
3.2.3 赋值语句	63	5.2.3 for 语句	131
3.3 数据与数据类型	64	5.2.4 三种循环语句的比较	134
3.3.1 常量与变量	64	5.3 循环的嵌套	136
3.3.2 整型变量与整型常量	65	5.4 辅助控制语句	139
3.3.3 浮点型变量与浮点型常量	67	5.4.1 break 语句	140
3.3.4 字符变量与字符常量	69	5.4.2 continue 语句	142
3.3.5 变量类型的转换	72	5.5 应用举例	144
3.4 变量的存储	75	5.5.1 穷举法	144
3.4.1 变量与内存的关系	75	5.5.2 迭代法	146
3.4.2 变量在内存中的表示形式	76	5.5.3 累加累乘法	148
3.5 指针变量	77	5.5.4 打印有规律的图形	149
3.6 综合案例	80	5.5.5 其他应用	150
小结	82	5.6 综合案例	151
习题 3	82	小结	153
4 选择结构程序设计	85	习题 5	153
4.1 门票价格问题	85	6 数组	157
4.2 条件的表示	86	6.1 成绩统计问题	157
4.2.1 关系运算	87	6.2 数组的概念	159
4.2.2 逻辑运算	88	6.2.1 数组的定义及访问	159
4.2.3 短路求值	90	6.2.2 数组的初始化	160
4.3 单分支结构	91	6.3 一维数组常见操作	162
4.3.1 if 语句	91	6.3.1 排序问题	162
4.3.2 复合语句	91	6.3.2 插入与删除问题	167
4.4 双分支结构	93	6.3.3 查找问题	170
4.4.1 if-else 语句	93	6.4 二维数组	172
4.4.2 条件运算	95	6.4.1 二维数组的定义及存储	172
4.4.3 if-else 嵌套	95	6.4.2 二维数组应用	174
4.4.4 if-else 配对	96	6.5 字符数组及字符串处理	177
4.5 多分支结构	99	6.5.1 文本数据处理	178
4.5.1 if 语句级联	99	6.5.2 字符数组处理字符串的方法	178
4.5.2 switch 语句	102	6.5.3 字符串的常见处理	180
4.6 综合案例	104	6.5.4 常用字符串处理函数	183
小结	115	6.6 指针与数组关系初步	184

6.6.1 指针的算术运算	185	8.2.2 引用结构体类型变量	245
6.6.2 数组元素的指针表示法	185	8.2.3 结构体变量的初始化	247
6.7 综合案例	188	8.2.4 结构体数组	248
小结	191	8.2.5 应用举例	250
习题 6	192	8.3 结构体指针的应用——单链表	257
7 函数	195	8.3.1 指向结构体的指针	257
7.1 福利彩票问题	195	8.3.2 动态内存分配	259
7.2 函数的概念	196	8.3.3 单链表	261
7.2.1 两类函数	196	8.4 共用体	271
7.2.2 函数的定义	197	8.4.1 共用体的概念	271
7.2.3 函数的声明	199	8.4.2 共用体变量的引用方式	272
7.3 函数的调用和返回语句	200	8.4.3 共用体类型数据的特点	273
7.3.1 函数的调用	200	8.5 枚举类型	274
7.3.2 函数的返回值	201	8.5.1 枚举类型的声明	274
7.4 函数的参数传递	203	8.5.2 枚举类型变量的声明及引用	274
7.4.1 值传递	204	8.6 用 typedef 定义类型	277
7.4.2 地址传递	205	8.7 综合案例	278
7.4.3 数组作为函数参数	206	小结	285
7.5 函数的嵌套与递归	209	习题 8	285
7.5.1 函数的嵌套调用	209	9 指针	289
7.5.2 函数的递归调用	211	9.1 指针解决的问题	289
7.6 变量和函数的作用域	214	9.2 变量的内存地址	290
7.6.1 全局变量和局部变量	214	9.3 指针基础知识汇总	291
7.6.2 变量的存储类别	219	9.4 特殊指针	300
7.6.3 内部函数和外部函数	223	9.4.1 指针数组	300
7.7 模块化程序设计	226	9.4.2 二级指针	302
7.8 综合案例	231	9.4.3 指向一维数组的指针	304
小结	236	9.4.4 函数指针	305
习题 7	236	9.5 综合案例	307
8 结构体	241	小结	308
8.1 平均绩点计算问题	241	习题 9	309
8.2 构建用户自己需要的数据类型	243	参考文献	312
8.2.1 定义结构体及结构体变量	243		

0 导言

当今社会，大数据、云计算、物联网等概念随处可见，人工智能、虚拟现实等应用也逐步普及，信息技术的应用已经并正在继续改变着人们的工作和生活，而这些新技术和应用的最重要核心是程序。毫不夸张地说，今天，每一个人都应该懂编程。那么，什么是程序？程序是怎样工作的？怎样编写程序呢？这些问题接下来将逐一得到解答，同时，你还会发现，学习编写程序的过程并不困难，而且还充满乐趣，学会编程会让你富有成就感，对自己更充满信心。

0.1 程序无所不在

神话故事西游记中，各路神仙可以随意上天入海。而今，这一切都已经成为现实，我国研制的神舟系列飞船多次载人往返于太空和地球之间，蛟龙号载人潜水器顺利潜入 7 000 多米的深海。而神话故事里的这一切，又是怎样实现的呢？神舟十号与天宫一号在 30 多万米以外的太空精准对接，是如何控制的呢？程序，使这一切成为可能！

人们生活在一个程序控制的时代。

绝大岁数的高科技，都是在程序的帮助下实现的，机器人、火箭升天、航空母舰、高科技武器、智慧农业、工业 4.0、智能电网、智能汽车和高铁等，都是在程序的控制下才能工作的。

如今最基本的工作、生活、学习、娱乐活动，也离不开程序。日常生活中使用的网络电视、冰箱、微波炉、洗衣机等智能家用电器，网上点餐、约车、购物、水电煤缴费、转账、理财等互联网生活方式，微信、微博、QQ 等社交软件，没有程序，都不可能实现。工作中，人们需要运行程序来完成网上办公、视频会议、E-mail 收发、财务等各类管理功能，方便地完成预订机票、预订酒店等。学习方面，现在也要依靠程序来选课、提交作业、查找和下载学习资料、与老师同学在线讨论问题、观看视频课等。生活中的视频点播、回看电视、网络小说、网络游戏等娱乐也离不开程序的帮助。

互联网、智能移动设备、云计算、大数据的共同基础、共同指挥官就是程序。



程序改变了人们的生活方式，推动了社会的发展。程序像空气和水一样，无处不在。无法想象，离开了程序，世界会变成什么样。

0.2 人人都要理解编程

人类社会过去需要几百年才能取得的进步，在这个信息时代，几年甚至几个月就能达到。为什么以程序为核心的信息技术能有如此大的威力呢？为了回答这个问题，首先必须理解程序，理解编程。

过去，不认识字的人被称为“文盲”，将来，不懂编程的人将是新的“文盲”。学会编程可以教会人们以一个全新的方式看世界，编程可以改变人们的思维方式，教会人们在这个时代里如何思考。

为了号召全体民众学习编程，2016年2月，美国政府专门投资40多亿美元推出了“全民计算机科学行动计划（Computer Science for All）”，要求全体民众，特别是从幼儿园到大学的学生，都要学习编程。苹果公司创始人乔布斯说：“我觉得每个人都应该学习一门编程语言。学习编程教你如何思考，就像学法律一样。学法律并不一定要为了做律师，但法律教你一种思考方式。学习编程也是一样，我把计算机科学看成是基础教育，每个人都应该花至少1年时间学习编程。”

人人都应该了解程序、懂程序、会编程序。所谓了解程序，就是要知道程序是什么，程序能改变世界依靠的是什么，程序是哪里来的，程序是如何工作的……

就像知道电能（交流、电流、电压）一样，人们也必须懂程序。所谓懂程序，就是要知道程序并不是高深的原理，要理解程序带给人们的独特逻辑思维和计算思维，就是要学会编程语言和工具的选择和使用。

可能有人会说，我学的不是计算机专业，将来也不可能从事程序员的职业，那我为什么要学会编程序。殊不知，今天的编程不是一个狭义的概念，它不仅包括懂不懂编程知识，也包括会不会编写程序，还包括能不能以编程思维考虑问题，等等。学会编程，可以更从容地应对现代社会的各种问题。

一般而言，编程有以下5个步骤。

- ① 理解问题需求：明确需要解决的问题是什么。
- ② 设计解题方案：明确怎么解决问题，解题步骤是什么。
- ③ 选择编程资源：明确哪些资源和技术可以使用。
- ④ 编程实现：用某种程序设计语言按确定的方案和技术编写所需的程序。
- ⑤ 调试运行：执行程序，观察结果。

下面介绍一个编程的例子。2014年12月8日，时任美国总统的奥巴马（本科学习政治学与国际关系、研究生学习法学）在参加一场由code.org组织的编程大会时，用了一个小时的时间，学会了编写一个小程序，画了一个正方形。他的学习步骤如下。

1. 理解问题需求

通过编程画一个规定边长的正方形，即输入一个正整数 n 代表边长，输出边长为 n 的正方形。

2. 设计解题方案

思考：一个正方形由 4 条边组成，其中两条是水平边，两条是垂直边。边的长短决定了正方形的大小，只要画了 4 条边，正方形就画好了。一条边由一些点组成的，水平边的点从左到右排列，垂直边的点自上而下排列。最后需要知道如何画一个点。

如下是一个长度为 5 的正方形（为了简化，这里的点用 * 表示）。

```
* * * * *
*           *
*           *
*           *
*           *
* * * * *
```

根据上面的分析，可以确定以下解题步骤（算法）。

- ① 首先输入边的长短 n ，确定正方形的大小。
- ② 重复画 n 个点“*”，完成画最上面的一条边。
- ③ 换行，在两条垂直的边的位置画点“*”，重复 $n-2$ 次。
- ④ 重复第②步，画出最底下的一条边。

3. 选择编程资源

不是所有解题方案都需要从头开始编程实现的，不是所有的算法都要自己设计的，很多共性的问题都有了成熟的解决方法，只要不存在知识产权问题，都可以直接使用，特别是针对功能比较复杂的程序，首先要考虑有没有现成的框架或代码可以直接使用。所以，在确定方案后的重要步骤，就是查找并确定可用的资源。这种可用资源可以是程序设计语言自带的，也可以是其他程序员贡献出来的在网上公开的开源代码（Open Source）。有些情况下，编程工作只要将各种可用资源集成起来就完成任务了，这时编程也称构建（Construction）。

本例比较简单，不需要使用框架，但需要使用程序设计语言提供的标准函数（库函数）实现以下功能。

- ① 输入正整数 n ，如 C 语言中的 `scanf()` 函数。
- ② 输出一个点“*”，如 C 语言中的 `printf("*")`。

在调用上述标准函数后，上述解题方案中的步骤需要如下思想来实现。

- ① 直接调用库函数输入点数 n ；
- ② 重复执行画点的操作，画出水平边；
- ③ 重复执行画点的操作，画出垂直边。

4. 编程实现

选择一种程序设计语言，例如 C 语言。一般情况下，选择程序设计语言主要考虑以下几个方面：一是编程者熟悉程度，总是选择大家最熟悉的语言；二是语言的特点，不同的语言有不同的特点，如有的语言适合科学计算，有的适合字符处理，

有的擅长界面设计等，可根据解题方案的需求选择最合适的语言；三是语言的可用资源，不同的语言自带或开源的资源不一样，一般选择可用资源最多的语言。有时，程序的使用者也会对于程序设计语言提出要求。

用选定的程序设计语言，按上述解题方案，利用选定的编程资源，编写程序完成上述解题步骤。

5. 调试运行

在编程环境中编辑、调试程序，定位错误、修改错误直至程序正确，生成可执行程序，然后运行程序观察结果，如果结果不正确，则继续修改，直至完全正确。

编程环境是辅助编程的集成环境，几乎所有的现代程序设计语言都有相应的编程环境，在这个集成环境中，可以方便地使用各种资源，智能化地完成编辑、调试、运行程序的功能。因而，熟悉编程环境、充分使用编程环境，已经是学习编程的重要步骤之一。

0.3 解剖一个程序

在实际应用中，程序往往和某些硬件结合在一起，例如，虚拟现实的程序一般都和一些可穿戴的设备一起工作。随着网络的广泛应用，往往是多个程序协同合作来实现一个功能，例如在网上点播视频时，视频服务器上有专门的视频管理程序来提供所需的视频，再通过视频传输程序将视频从服务器传输到客户端计算机上，再由客户端计算机上的视频播放程序来放映视频。视频管理程序、视频传输程序和视频播放程序是3个独立的程序，它们同时协同工作，就可以完成视频点播功能。

为了简单起见，只考虑完成独立功能的单个程序。每个程序都是为实现某些功能的一组对数据进行操作的序列。下面，通过一个简单的“运动计步器”例子来理解程序。

“运动计步器”记录人们每天行走的步数，显示人们最近一段时间内每天的运动步数。“运动计步器”中有一个称为震动传感器的硬件和一个计步器程序，震动传感器会感应到人们是否在行走。

每行走一步震动传感器会发一个信号给计步器程序，计步器程序需要实现以下功能。

- ① 要记录当天任何时候的行走步数。
- ② 在每天的零点，存储前一天的步数为历史步数，并将今天的步数置为0。
- ③ 每接到一个震动传感器发来的信号，步数就加1。
- ④ 可以根据功能选择显示当天的步数。
- ⑤ 可以根据功能选择以数据或曲线的形式显示最近几天的步数。

为了实现上面这些功能，计步器程序中需要存储的数据和对这些数据进行的操作如下。

① 数据：当前步数、历史步数。

② 操作：当前步数置 0、当前步数加 1、存储前一天步数、显示历史步数。可以看到，整个计步器程序实际上就是对数据执行操作的序列。

再看一个具有健康提示的“运动计步器”，除了上述记录和显示每天行走步数的功能外，它还存储了体重等数据，并且可以通过一定的方法计算出有益于健康的每天理想行走步数，如果连续 3 天行走的步数与理想步数差距达到一定阈值，就会显示“你每日行走太少”或者“你每日行走过多”的健康提示信息。

与前面的计步器程序相比，这个具有健康提示的“运动计步器”增加了以下数据和操作。

① 数据：体重等数据。

② 操作：计算理想步数、判断实际行走步数与理想步数的差距后显示健康提示信息。

同样，如果把这个“运动计步器”扩展为网络版，可以将行走步数传输到网络服务器，并且可以与朋友的步数比较，进行排序。这就成了现在的“微信运动”，读者可以自行分析一下，“微信运动”这个程序又由哪些数据和操作组成。

再进一步分析，程序中的数据是有区别的：当前步数是单个数据；历史步数是包含很多天行走步数的一组数据。

同样，操作也是有区别的：当前步数置 0 是单个独立的操作；判断实际行走步数与理想步数的差距后，显示不同的健康提示信息是选择性操作；显示历史步数是需要重复显示多个数据的操作。

但是，不论复杂与简单，一个程序的主要内容就是两个部分：数据和操作。程序设计的过程就是用一种程序设计语言正确地表示出对数据进行操作的过程。

0.4 编程的主要内容

活到老，学到老。从小到大，人们需要不断学习。小时候学走路、学游泳，长大了学骑自行车、学开车。在小学、中学阶段，学语文、学数学、学物理，到了大学要学编程。不同的学习，目的不同，达到的效果也不一样，学习的内容自然也不同。

学游泳和学开车是为了学一种技能，需要掌握在一定理论指导下的实践。学习的关键在于实践技能，如果仅仅停留在理论层面，只学习书本上的游泳要领和技巧，那是没有用的。当然，理论也是需要的，有了理论的指导，才有姿势优美的蛙泳、蝶泳和仰泳等。同样，学会开车了，没有交通法规的指导，照样寸步难行。

学语文，学的是一种文学修养，背诵“三字经”“论语”等，学的是处世之道、修身齐家治国平天下的方法；学数学，学的是理论思维，它是其他学科的基础，关键在于学会定义、定理、证明的精髓和公理化的理论思维方法；学物理，目的是掌

握实验思维方法，借助于特定的设备，从基本实验出发归纳出最一般的结论，再推理出符合逻辑的规律，由实验检验。

那么，学习计算机，学习程序设计，其目的又是什么呢？是学习运用计算机科学的知识进行问题求解的方法，是训练如何设计出能解决问题的计算机系统的实践能力，是学习与数学思维相类似、抽象形式更为丰富的一种新的思维方式。

学习一种程序设计语言，学会用某种程序设计语言来编程，必须掌握 4 个维度的内容，即语法知识、编程技能、编程思维、编程规范。

1. 语法知识

和其他任何语言一样，程序设计语言也有自己的语法和语义。简单地说，语法就是一些书写规则，语义就是符合语法规则的表述的含义。由于程序最终交给计算机去执行，完成程序的功能，因此，编程时必须严格按照语法规则书写程序中的各个元素，任何细微的语法错误都会导致计算机不能正常执行程序。

虽然程序设计语言种类繁多，但各种程序设计语言的语法大同小异。从本质上讲，一个程序就是对数据的一系列操作，因而，程序设计语言的语法知识一般都包括程序的结构组成、数据如何表示、操作怎么表示等。

例如，C 语言中，一个程序是由一个 `main()` 函数和一些（0 个或多个）自定义函数组成。每个函数都有规定的定义和调用格式，数据可以有不同的类型，如 `int`、`float` 分别表示整数和实数，不同数据类型的数据所占空间、表示范围和可执行运算都可能不同。数据还可以是简单数据或构造数据，构造数据是指含多个分量的复杂数据，需要说明分量个数、分量类型、分量次序等。不同的操作表示方法也不同，包括基本的数据输入、输出和赋值等操作，还包括选择和循环等操作。

语法知识必须理解，不用死记硬背，编程时可以查阅相关程序设计语言的书籍。使用一种程序设计语言编写程序一段时间后，语法知识自然也就能信手拈来了。现在一些集成开发环境，都会提供一些智能化的语法检查工具，通过文字、颜色、声音等实时提示语法知识或语法错误情况。

2. 编程技能

如果学习编程只学会了语法知识，面对一个问题，不会设计程序去解决它，那就像只学了丰富的游泳知识却不会游泳一样，是没有实用意义的。因此，学习程序设计语言，比学习语法知识更重要的是，语法知识的应用，动手编程解决问题能力的训练。

学习程序设计的重要环节是编程实践。一般而言，每一次理论课后都需要编程来练习所学的理论知识，编程实验的时间应该大于理论学习的时间。在学会如何使用每个知识点的情况下，从解决简单问题开始，练习运用前面所学的知识编写程序、调试程序、运行程序得到预期结果。

编程技能除了对语法知识点的灵活应用、对集成开发环境的熟练使用外，还包括一些常用技巧的掌握，如分而治之（把一个复杂问题分解为多个简单问题），充分复用已有功能模块，先设计解题思路再动手编程，自顶向下逐步求精思想，以及顺序、选择、循环 3 种控制结构的应用等。

编程技能的掌握没有捷径，唯有多练习、多实践。学一门程序设计语言，少则需要编写 30~40 个程序，多则 100 多个程序也不算多。功夫到了，编程技能自然就有了。

3. 编程思维

前面讲过，在这个互联网+时代，信息技术已经改变了人们的工作和生活方式，同时也需要人们有与现代信息技术相适应的思维方式，称为计算思维。编程思维，就是这种新的思维方式的重要内容。除了专业程序员外，大多数人未来都不是以编程作为职业的，但是编程思维是每个人必备的思考方法，大家都要学会“怎样像计算机科学家一样思维”。学习程序设计的更重要任务，就是训练并学会使用编程思维。

有了计算机的帮助，就能用新的思维方式去解决那些之前不敢尝试的问题，实现“只有想不到，没有做不到”的境界。其实，这种新的思维方式也出现在人们的日常生活中，例如，早晨去教室时，把当天需要的东西放进背包，这就是预置和缓存；当发现钥匙不见了，沿着走过的路返回去寻找，这就是回推；在超市付账时，应当去排哪个队呢？这就是多服务器系统的性能模型。

那么，编程思维包含哪些内容呢？周以真教授在定义计算思维时，总结了包括约简、嵌入、转化、仿真、抽象、分解、并行、递归和推理等 20 多种方法和能力，其中大部分在学习编程时都会涉及，即使在编写简单程序时也会应用到，如将具体问题一般化的抽象，遍历所有数据的穷举，自动重复并按条件结束的迭代等，以及数据的比较、交换、查找、排序等，还有各种经典的算法思想等，这些都属于编程思维。

在后续章节的 C 语言程序设计内容中，会逐步介绍到这些方法，读者要学习并习惯使用它们来思考问题。

4. 编程规范

编程工作，有人认为是极需创意的设计工作，属于艺术，程序员也是一类艺术家；也有人认为只是用现成技术完成既定解题方案，充其量也就是技术，程序员是工程师，并形象化地称为“码农”。但不论怎样，编程应该遵循一定的规范，养成很好的习惯，把自己编的程序打造成高质量的作品！

学习编程时，从一开始就要重视程序代码的质量，要学习好的编码规则，养成良好的编码习惯，使用并积累编程经验。如要先设计再编码，要有明确的命名规则，书写程序时遵循统一的缩进规则，程序中要加入适当的注释等。

0.5 如何学好程序设计

经常听到有人这样说，“编程很难”“听得懂，不会做”“看得懂，不会做”“考的出，不会编”。那么，程序设计真的这么难吗？其实不然。除非是解决很复杂的问题，设计一般的程序，中学生都不会有任何问题。很多人觉得难的主要原因是不适

应思维方式上的变化，这就像隔了一层窗户纸，感觉对面是一个完全陌生的世界，其实捅破了，就会发现也就那么回事。要做的就是捅破那层不用花太大力气就能捅破的窗户纸。

“兴趣是最好的老师”。学会编程，会看到一个不一样的世界；学会编程，在探究、改造这个世界时将如虎添翼；学会编程，创造力会倍增；学会编程，人生将更加丰富多彩。学习编程，首先要培养对程序的兴趣，有了兴趣，学习起来就会觉得轻松快乐。随着设计的一个个程序运行出正确结果，就会越来越喜欢编程，越来越有成就感，越来越自信。

那么，应该如何来学习程序设计呢？为了帮助读者顺利学习编程，这里总结了学习程序设计的“一个要领”“三个关键”“十大诀窍”。

1. 一个要领：循序渐进

不要想一口吃个胖子。学习程序设计，从依样画葫芦开始，逐步过渡到自己编程；从简单有趣的程序开始，逐步增加功能。刚开始学习编程时，重要的是程序运行的结果，然后在弄明白程序的基础上逐步增加功能。不要一开始就编写复杂程序，简单程序编熟练了，复杂程序自然也就编了。

2. 三个关键：紧跟、坚持、理解

“紧跟”是指教到哪、学到哪、编到哪。与其他知识的学习不同，程序设计的学习涉及新的思维方式和习惯的适应。因而，学习第一门程序设计语言课程时，教师的作用不可或缺；学习时要尽量跟上教师的讲课节奏，及时学会教师讲授的内容，并及时在编程实践中应用和巩固所学的理论知识。

“坚持”是指遇到问题不退缩，没有解决不了的问题。刚开始学习程序设计时，会接触很多新的概念和知识，思维方式也有差别，遇到一些小的困难也是正常的，这时候决不能轻言放弃，咬定青山不放松，问题也就迎刃而解了。在解决一个又一个的困难时，不知不觉中就会成为一个编程高手。

“理解”是指得到正确结果不算结束，了解所以然才达到目的。学习程序设计的目标不是记住知识点，即使将知识点背得滚瓜烂熟也没用，会用来编程解决问题才有意义。编出程序，运行出结果，并不是完成任务，必须理解“为什么要这样设计”，思考有没有其他更优的解决方案，还有哪些问题也能用类似的方法解决。

3. 十大诀窍

一是先看后听。就是在听课学习新的知识之前，要有所准备，先看书预习、看相关的视频熟悉内容，知道重点和难点，有备而来的听课会达到事半功倍的效果。

二是先听后问。就是听课中不能完全理解的内容要及时问老师、问同学，不要让问题积累起来，尽量做到学习时就理解。

三是边学边练。学习了新的思维方法、新的语法知识等，要及时应用它们来练习编写程序，通过编写程序做到真正掌握所学的内容。

四是边练边调。在练习本上编写程序是不够的，因为只有将程序放到计算机上来调试运行，才知道编写的程序是否正确，能不能运行出正确的结果。